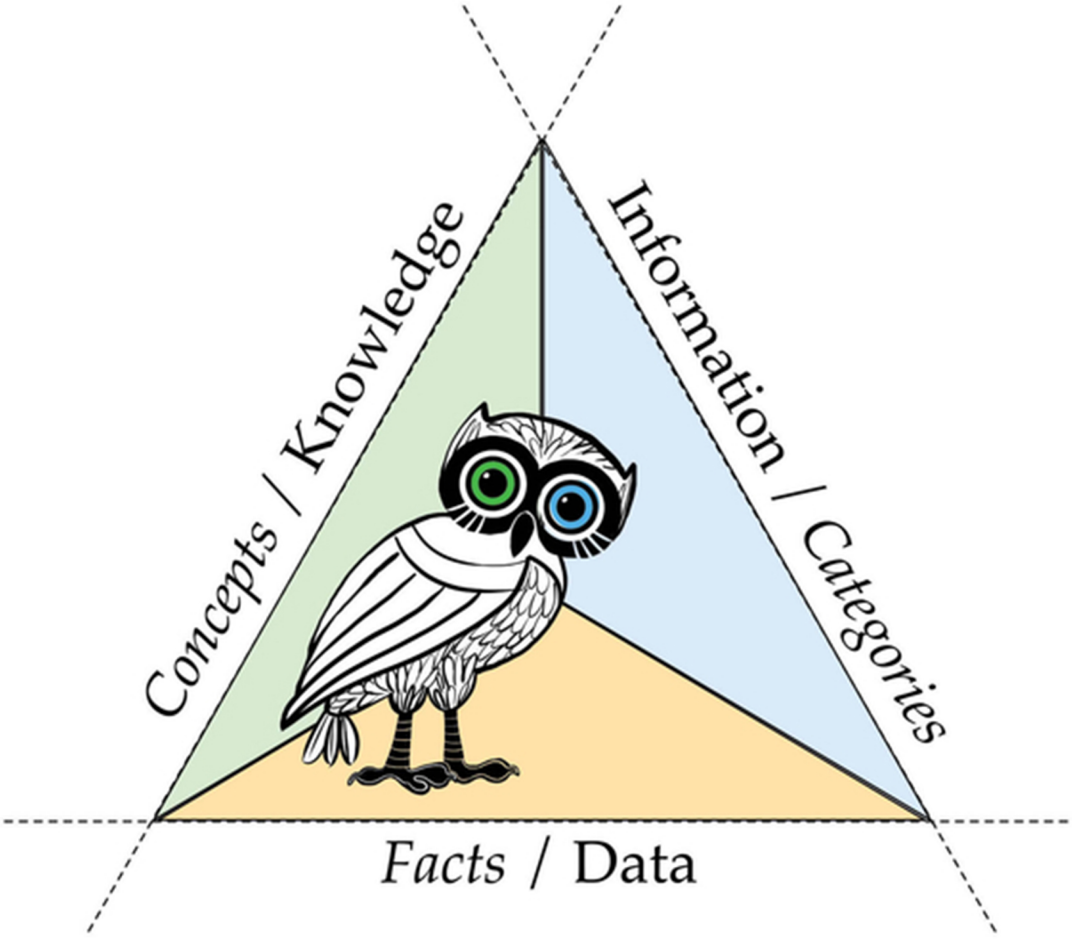
Ontological Prisms

Knowledge Engineering for Enterprise & Agentic Systems



Rémy Fannader

WILEY

Ontological Prisms

Ontological Prisms

Knowledge Engineering for Enterprise & Agentic Systems

Rémy Fannader
Founder, Caminao

WILEY

Copyright © 2026 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial intelligence technologies or similar technologies

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permission.

The manufacturer's authorized representative according to the EU General Product Safety Regulation is Wiley-VCH GmbH, Boschstr. 12, 69469 Weinheim, Germany, e-mail: Product_Safety@wiley.com.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and the authors have used their best efforts in preparing this work, including a review of the content of the work, neither the publisher nor the authors make any representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data has been applied for:

Hardback ISBN: 9781394352623

ePDF ISBN: 9781394352647

ePUB ISBN: 9781394352630

oBook ISBN: 9781394352654

Cover Design: Wiley

Cover Image: Courtesy of Rémy Fannader

Perplexity is the beginning of knowledge.

Khalil Gibran

Contents

Preamble *xxi*
About the Author *xxiii*

Part I 1

1 Current Approaches and Paradigm Shift *3*
1.1 Ontologies and Artificial Intelligence *3*
1.1.1 A Brief and Personal Perspective on Artificial Intelligence Past and Present *3*
1.1.1.1 Mechanical Brains *3*
1.1.1.2 Symbolic vs Connectionist Tracks *4*
1.1.1.3 Language and Common Sense *4*
1.1.1.4 Mimicry and Rote Learning *5*
1.1.1.5 Agents in No Mind's Land *6*
1.1.2 An Ontological Manifesto *7*
1.1.2.1 Facts and Data *7*
1.1.2.1.1 Facts Are Not Information *7*
1.1.2.1.2 Neural Connectors Represent Facts, Not Meanings *7*
1.1.2.2 Language and Reason *7*
1.1.2.2.1 Information Is Not Knowledge *7*
1.1.2.2.2 Language Is Not Knowledge *8*
1.1.2.2.3 There Is No Truth in Languages *8*
1.1.2.2.4 There Is No Reason in Data *8*
1.1.2.3 Judgment and Computation *8*
1.1.2.3.1 Reason Is Not Judgment *8*
1.1.2.3.2 Judgments Begin Where Algorithms End *8*
1.1.2.3.3 Knowledge Graphs Are Not Knowledge *8*

1.1.2.3.4 Intelligence Cannot Emerge from Canned Words *9*
1.2 Ontologies and Knowledge *9*
1.2.1 Foundational Ontologies *9*
1.2.1.1 Unified Foundational Ontology *9*
1.2.1.2 Descriptive Ontology for Linguistic and Cognitive Engineering *10*
1.2.1.3 Typical Extensions *11*
1.2.1.3.1 Unified Foundational Ontology Extensions *11*
1.2.1.3.2 Basic Formal Ontology *11*
1.2.1.3.3 Tropos *11*
1.2.1.3.4 Common Core Ontology *12*
1.2.1.4 Limits of Foundational Ontologies *12*
1.2.1.4.1 Foundational Blind Spot *12*
1.2.1.4.2 From Dimensions to Concepts *13*
1.2.1.4.3 Ontologies Are Not Models *14*
1.2.2 Empirical Approaches and Meta-models *14*
1.2.2.1 Domain-specific Ontologies *14*
1.2.2.2 Abstractions, Reuse, and the Limits of Meta-models *16*
1.2.3 What to Expect *17*
1.3 The Ontological Prisms Paradigm *18*
1.3.1 Ontologies and Semiotics *18*
1.3.2 Ontologies and Philosophy *19*
1.3.3 Ontologies at Work *20*
1.3.4 Process and Case Study *22*
1.3.4.1 Physical *22*
1.3.4.2 Functional *22*
1.3.4.3 Organizational *22*
1.3.5 M*R/K and Ontological Prisms *22*
Key Points *24*

2 Ontological Realms *25*
Overview *25*
2.1 Extensional Realm: Facts *26*
2.1.1 From Things to Facts *26*
2.1.1.1 Collecting Facts *26*
2.1.1.2 The Naming Game *27*
2.1.1.3 The Semantics of Connectors *28*
2.1.2 Facts, Data, Modalities *30*
2.1.2.1 Facts and Data *30*
2.1.2.2 Facts and Modalities *31*
2.1.3 Facts and Models *32*
2.1.3.1 Data Models *33*

2.1.3.2 Generative Language Models *33*
2.2 Intensional Realm: Concepts *34*
2.2.1 Relinquishing Aristotelian Ur-elements *34*
2.2.2 Concepts Modalities *35*
2.2.3 Concepts and Models *36*
2.2.3.1 Entity–Relationship Models *37*
2.2.3.2 Conceptual Graphs *38*
2.2.3.3 Knowledge Graphs *40*
2.3 Logical Realm: Categories *41*
2.3.1 Anchors and Connectors *41*
2.3.1.1 Anchors and References *41*
2.3.1.2 Logical Connectors *43*
2.3.1.3 Powertypes *44*
2.3.2 Aspects *45*
2.3.2.1 Objectives *45*
2.3.2.2 Taxonomy and Syntax *45*
2.3.2.3 Ontological Constraints *46*
2.3.2.4 Aspects and Relational Schemas *47*
2.4 Ontological Modalities *48*
2.4.1 Meta-models vs Modalities *48*
2.4.1.1 Objective *48*
2.4.1.2 The Underlying Flaw of Meta-models *49*
2.4.2 A Taxonomy of Ontological Modalities *49*
2.4.2.1 Intrinsic Modalities *50*
2.4.2.1.1 Temporality *50*
2.4.2.1.2 Instantiation *50*
2.4.2.1.3 Realm *50*
2.4.2.1.4 Identity *51*
2.4.2.1.5 Veracity *51*
2.4.2.2 Functional Modalities *51*
2.4.2.2.1 Structures *51*
2.4.2.2.2 Agency and Communication *52*
2.4.2.3 Modalities and Ontological Commitments *53*
Key Points *53*

3 Ontological Prisms and Interoperability *55*
Overview *55*
3.1 Alignment and Integration *55*
3.1.1 Homogeneous Alignment *56*
3.1.1.1 Alignment Principles *56*
3.1.1.2 Limits of Alignment and Integration *57*

3.1.2 Integration of Heterogeneous Ontologies *58*
3.1.2.1 Ontological Gears *58*
3.1.2.1.1 Thesauri *58*
3.1.2.1.2 Taxonomies *59*
3.1.2.1.3 Domains *60*
3.1.2.2 Semantic Layers and Interoperability *60*
3.2 Interoperability of Heterogeneous Ontologies *62*
3.2.1 Interoperability and Reuse *63*
3.2.2 Facts, Concepts, Meanings *63*
3.2.2.1 Abstraction: Entity Resolution *63*
3.2.2.2 Realization: The Matter of Names *65*
3.2.3 Facts, Categories, Digital Twins *65*
3.2.3.1 Interoperability with Engineering Solutions *66*
3.2.3.1.1 From Categories to Facts: Unified Modeling Language *66*
3.2.3.1.2 From Facts to Categories: Object-role Modeling *66*
3.2.3.2 Abstraction and Artifacts *66*
3.2.3.2.1 The Matter of Time *66*
3.2.3.2.2 Schemas and Reverse Engineering *67*
3.2.3.3 Realization of Artifacts *68*
3.2.3.4 Viable Systems and Digital Twins *69*
3.2.4 Concepts, Categories, Organization *70*
3.2.4.1 Abstraction: Alignment of Emerging Changes *71*
3.2.4.2 Realization: Alignment of Designed Changes *71*
3.2.4.3 Architectures: Alignment of Time Frames *72*
3.2.4.4 Transparency, Traceability, Accountability *73*
3.3 Interoperability and Collective Intelligence *74*
3.3.1 Pushing the Edges, Moving the Lines *74*
3.3.1.1 Perceptions *74*
3.3.1.2 Projections *75*
3.3.1.3 Experience *75*
3.3.2 Collective Intelligence and Holistic Knowledge *76*
3.3.2.1 Interoperability and Holistic Knowledge *76*
3.3.2.1.1 Nonseparability: Symbolic vs Connectionist Knowledge *76*
3.3.2.1.2 Causality: Semantic Layers and Reasoning Tiers *76*
3.3.2.1.3 Maturity: Knowledge Modalities *77*
3.3.2.2 Mechanical Brains vs Collective Minds *77*
Key Points *77*

4 Ontological Prisms and Governance *79*
4.1 Environments and Organizations *79*
4.1.1 Environments: Data and Documents *79*

4.1.2 Enterprise Architecture: Information *80*
4.1.2.1 Technical Integration *80*
4.1.2.2 Functional Integration *81*
4.1.3 Organization: Knowledge *82*
4.1.3.1 Enterprise Level *82*
4.1.3.2 System Level *83*
4.1.3.3 Learning Organization *84*
4.2 Data and Value Chains *85*
4.2.1 Data and Metadata *85*
4.2.1.1 Datasets *85*
4.2.1.2 Documents *85*
4.2.1.3 Metadata *86*
4.2.1.3.1 Objectives *86*
4.2.1.3.2 Representation *86*
4.2.1.3.3 Data Products *87*
4.2.2 Knowledge and Value Chains *88*
4.2.2.1 Added Value *88*
4.2.2.1.1 Perception and Added Value *88*
4.2.2.1.2 Projection and Added Value *89*
4.2.2.1.3 Experience and Added Value *89*
4.2.2.2 Transaction Costs and Value Chains *89*
4.2.2.2.1 Transaction Costs *89*
4.2.2.2.2 Outsourcing Knowledge-intensive Activities *90*
4.2.2.3 Documents and Blockchains *91*
4.2.2.3.1 Documents as Currency *91*
4.2.2.3.2 Blockchains *91*
4.3 Data Protection and Intellectual Property *92*
4.3.1 Data Protection *92*
4.3.1.1 Principles *93*
4.3.1.2 Institutional Approach: General Data Protection Regulation *93*
4.3.1.3 Ontological Approach *94*
4.3.2 Intellectual Property *95*
4.3.2.1 Creative Works and Copyrights *95*
4.3.2.2 The Quagmire of Patents *96*
4.3.2.3 Intellectual Property and Generative Artificial Intelligence *97*
4.4 Complexity Management *97*
4.4.1 The Fabric of Complexity *98*
4.4.1.1 2D Complexity *98*
4.4.1.2 3D Complexity *99*
4.4.2 Organization and Complexity *100*
4.4.2.1 Organization and Complexity Management *100*

4.4.2.2 Ontological Complexity and Systems Entropy *101*
4.4.3 Complexity and Evolution *103*
4.4.3.1 Natural Evolution *103*
4.4.3.2 Evolution of Organizations *104*
4.4.3.3 Sustainability and Resilience Through Ontological Prisms *105*
Key Points *106*

Part II 109

5 Ontologies and Language *111*
Overview *111*
5.1 Anatomy of Languages *111*
5.1.1 Communication vs Representation *112*
5.1.2 Communication: Signs and Symbols *113*
5.1.3 Representation: Grammars and Semantics *114*
5.2 Generative Language Models *115*
5.2.1 A Functional Perspective *115*
5.2.2 Limits of Generative Language Models *117*
5.2.2.1 Functional Limits of Generative Language Models *117*
5.2.2.2 Inherent Limits of Pure Generative Language Models *117*
5.2.3 Agents and Collaborative Architectures *118*
5.2.3.1 Graph-enhanced Retrieval-augmented Generations *119*
5.2.3.2 Language Models and Ontological Prisms *120*
5.2.3.3 Knowledge-driven Architectures *120*
5.3 Languages and Knowledge *121*
5.3.1 Direct and Mediated Communication *121*
5.3.2 Language of Thought and the Computation Paradigm *122*
5.3.3 Conversational vs Contextual Semantics *123*
5.3.3.1 Extensional Semantics *123*
5.3.3.2 Intensional Semantics *124*
5.3.3.3 Semantic Layers Revisited *125*
5.3.4 Language at the Edges of Knowledge *126*
5.3.4.1 Meanings and Semantic Entropy *126*
5.3.4.2 Languages and Emerging Concepts *127*
Key Points *128*

6 Reasoning with Ontological Prisms *131*
6.1 Ontologies and Reason *131*
6.1.1 The Scope of Reason *131*
6.1.2 Reasoning Paths *132*

6.2 Formal Reasoning *133*
6.2.1 Reason and Logic *133*
6.2.1.1 Nominal Logic *133*
6.2.1.2 Propositions and Predicates *134*
6.2.1.3 Predicates and Categories *135*
6.2.2 Explanation and Comprehension *136*
6.2.2.1 Correlations *137*
6.2.2.2 Causations *137*
6.2.2.3 Interpretations *137*
6.2.2.4 Explanation vs Comprehension *138*
6.3 Language Models and Logic *138*
6.3.1 Semantic Fault Lines and Fuzzy Reasoning *138*
6.3.1.1 Lightweight Ontologies *139*
6.3.1.2 Language Concept Models and Fuzzy Reasoning *139*
6.3.2 Hallucinations and Nominal Reasoning *140*
6.3.2.1 Hallucinations *140*
6.3.2.2 Nominal Embeddings and Vector Semantics *141*
6.4 Empirical Reasoning *143*
6.4.1 Data Analytics *143*
6.4.1.1 Reality and Causal Chains *143*
6.4.1.2 Shared Vocabularies and Aspects *144*
6.4.2 Arguing with Facts *146*
6.4.2.1 Empirical Challenges *146*
6.4.2.1.1 Unreliable Data *146*
6.4.2.1.2 Misleading Information *147*
6.4.2.1.3 Subjective Beliefs and Intents *147*
6.4.2.2 Argumentation Profiles *147*
6.4.2.2.1 Evidence Reasoning *147*
6.4.2.2.2 Interpretative Reasoning *147*
6.4.2.2.3 Controversial Reasoning *147*
6.5 Reason and Knowledge *148*
6.5.1 Actionable Knowledge *148*
6.5.1.1 Declarative Knowledge Representation *148*
6.5.1.2 Heterogeneous Reasoning *149*
6.5.1.3 Rules Representation *150*
6.5.2 Ontological Inference *150*
6.5.2.1 Inference Patterns *150*
6.5.2.2 Empirical Inference *151*
6.5.2.3 Semantic Inference *152*
6.5.2.3.1 Metonymies *152*
6.5.2.3.2 Metaphors *152*

6.5.2.4 Deontic Inference *153*
6.5.3 Reason and Collective Learning *153*
6.5.3.1 Communication and Organization *153*
6.5.3.2 Learning Organizations *154*
Key Points *154*

7 Decision-making with Ontological Prisms *155*
7.1 Decision-making Dimensions *155*
7.1.1 Intrinsic Assessment of Decision-making Processes *155*
7.1.1.1 What: The Matter of Decisions *155*
7.1.1.2 Who: Decision-makers *156*
7.1.1.3 How: Decision-making Processes *156*
7.1.1.4 Where: Problem and Solution Spaces *156*
7.1.1.5 When: External Time Frames and Events *156*
7.1.2 Decision-making Environments *156*
7.1.2.1 Document-based Decision Support Systems *157*
7.1.2.2 Model-based Decision Support Systems *157*
7.1.2.3 Knowledge-based Decision Support Systems *158*
7.1.2.4 Decision-making and Ontologies *158*
7.2 Problem and Solution Spaces *158*
7.2.1 Decision-making Spaces *158*
7.2.1.1 Microstates and Macrostates *158*
7.2.1.2 Decision-making Spaces *159*
7.2.2 Decision-making Templates *160*
7.2.2.1 Decisions Intrinsic Footprint *160*
7.2.2.2 Decision-making Templates *161*
7.3 Visibility and Time Frames *162*
7.3.1 Transparency and Traceability *162*
7.3.1.1 What Is to Be Decided, and Why *163*
7.3.1.2 Who Is to Contribute and Be Accountable *163*
7.3.1.2.1 Entitlement and Contributions *163*
7.3.1.2.2 Explainability and Accountability *163*
7.3.1.3 Where Options Should Be Considered *164*
7.3.1.4 When Decisions Should Be Made *164*
7.3.2 Decision-making Horizons *165*
7.4 Parties and Structure *166*
7.4.1 Sided Decision-making *166*
7.4.1.1 One-sided Processes *167*
7.4.1.2 Collaborative Processes *167*
7.4.1.3 Adversarial Processes *167*
7.4.2 Phased vs Iterative Processes *168*

7.4.2.1 Phased Decision-making Processes *168*
7.4.2.2 Iterative Decision-making Processes *169*
7.4.2.3 Decisions, Time, Knowledge *170*
7.5 Agentic Decision-making *171*
7.5.1 Knowledge-based Decision-making *172*
7.5.2 Observation *172*
7.5.2.1 Purpose *172*
7.5.2.2 Transitions *173*
7.5.3 Orientation *173*
7.5.3.1 Purpose *173*
7.5.3.1.1 Horizons *174*
7.5.3.1.2 Modalities *174*
7.5.3.2 Transitions *174*
7.5.4 Decision *175*
7.5.4.1 Purpose *175*
7.5.4.2 Transitions *177*
7.5.5 Action *177*
7.5.5.1 Purpose *177*
7.5.5.2 Transitions *178*
7.5.6 Assessment: Learning Curves and Complexity Management *179*
7.5.6.1 The Brain of the Firm *179*
7.5.6.2 Learning Curves *180*
7.5.6.3 Spaces and Complexity *180*
7.5.6.4 Complexity and Entropy *180*
Key Points *181*

Part III 183

8 Knowledge-based System Engineering *185*
8.1 Systems and Organizations *185*
8.1.1 ABC of Enterprise Architecture *185*
8.1.1.1 Actionable *186*
8.1.1.2 Brainy *186*
8.1.1.3 Collaborative and Competitive *186*
8.1.2 Engineering and Digital Immersion *186*
8.1.2.1 Change Is the New Normal *186*
8.1.2.2 Isolation and Interoperability *187*
8.1.2.3 Sustainability and Time Frames *187*
8.2 Engineering Processes and Ontological Prisms *188*
8.2.1 Scope: The Pagoda Blueprint *188*

8.2.2 Objectives: User Stories and Use Cases *189*
8.2.2.1 Stories *189*
8.2.2.2 Cases *190*
8.2.2.2.1 Use Cases *190*
8.2.2.2.2 Business Cases *190*
8.2.2.2.3 Function and System Cases *191*
8.2.2.3 Models *191*
8.2.3 Processes: Workflows and Remits *192*
8.2.3.1 Workflows *192*
8.2.3.2 Remits and Workshops *193*
8.3 Anchoring Engineering to Ontologies *194*
8.3.1 Anchoring Business, Function, and Use Cases *194*
8.3.1.1 Use Cases *194*
8.3.1.2 Business and Function Cases *195*
8.3.2 Anchoring Models *196*
8.3.2.1 Conceptual Models *196*
8.3.2.2 Logical Models *196*
8.3.2.3 Technical Models *197*
8.3.2.4 Anchors and Semantic Layers *197*
8.3.3 Engineering Loop *198*
8.4 Requirements with Ontological Prisms *199*
8.4.1 Requirements Capture and Elicitation *199*
8.4.1.1 Capture *199*
8.4.1.2 Elicitation and Earmarking *200*
8.4.2 Requirements Analysis: Lines and Anchors *201*
8.4.2.1 Drawing Lines *201*
8.4.2.2 Anchors *201*
8.4.3 Requirements Analysis: Granularity and Variants *203*
8.4.3.1 Requirements Granularity and Aristotle's Classical Three Unities of Drama *203*
8.4.3.2 Object Variants, Powertypes, Delegation *205*
8.4.3.2.1 Powertypes *206*
8.4.3.2.2 Delegation *206*
8.4.3.3 Processes Variants and Rules *207*
8.4.3.3.1 Rules Taxonomy *207*
8.4.3.3.2 Rules Mappings *208*
8.5 Engineering with Ontological Prisms (and Generative Artificial Intelligence) *209*
8.5.1 Work Units *209*
8.5.1.1 Development Models *209*
8.5.1.2 Flow-based Work Units *210*

8.5.2 Model-based Engineering *211*
8.5.2.1 Domain-driven Design *211*
8.5.2.1.1 Roots, Aggregates, and Bounded Contexts *212*
8.5.2.1.2 Ubiquitous or Domain-specific Languages *213*
8.5.2.2 Model-driven Architecture *213*
8.5.2.2.1 Models, Domains, Languages *213*
8.5.2.2.2 Models and Architecture *213*
8.5.2.3 From Models to Code *214*
8.5.2.3.1 From Business Processes to Executable Models *214*
8.5.2.3.2 Interpretation of Conceptual Models *214*
8.5.2.3.3 Actionable and Executable Models *215*
8.5.2.4 Generative Artificial Intelligence Agents and Coding *215*
8.5.3 Ontological Validation *215*
8.5.3.1 Domain Consistency *215*
8.5.3.2 Ontological Consistency *216*
8.5.3.2.1 Abstraction Semantics: Parthood vs Inheritance *216*
8.5.3.2.2 From Concepts to Categories *217*
8.5.3.2.3 From Categories to Facts *218*
8.5.4 Risks Management with Ontological Prisms *219*
8.5.4.1 Threats and Black Swans *219*
8.5.4.2 Tests and Ontological Prisms *219*
8.5.4.3 Tests and Systems Entropy *220*
Key Points *221*

9 The Engineering of Knowledge *223*
Outline *223*
Incremental Development *223*
Iterative Development *224*
9.1 Mapping Extensions *224*
9.1.1 Ontological Footprint *225*
9.1.1.1 Contexts and Concerns *225*
9.1.1.1.1 Anchors *225*
9.1.1.2 Boundaries *226*
9.1.2 Ontological Blueprints *227*
9.1.2.1 Ontological Modalities *227*
9.1.2.1.1 Temporality *227*
9.1.2.1.2 Realm, Instances, and Tally *228*
9.1.2.1.3 Identity Principle *228*
9.1.2.1.4 Agency *229*
9.1.2.1.5 Communication *229*
9.1.2.1.6 Structure *229*

9.1.2.1.7 Veracity *230*
9.1.2.2 Modalities and Processes *230*
9.1.2.3 Blueprints, Profiles, and Templates as Ontological Commitments *232*
9.1.2.3.1 Blueprints *232*
9.1.2.3.2 Profiles *234*
9.1.2.3.3 Templates *234*
9.1.2.4 Modalities and Patterns Representation *234*
9.1.2.4.1 Hypergraphs and Vectors *234*
9.1.2.4.2 Web Ontology Language and Prolog *235*
9.1.3 Perceptions: Semantics and Concepts *236*
9.1.3.1 Semantic Alignments *236*
9.1.3.1.1 Symbolic Approach *237*
9.1.3.1.2 Generative Approach *238*
9.1.3.2 Conceptual Alignments *238*
9.1.3.2.1 Modalities Alignment *239*
9.1.3.2.2 Profiles and Blueprints Alignment *240*
9.1.3.2.3 Applying Profiles and Blueprints *241*
9.1.4 Experience: Aspects, Definitions, Categories *242*
9.1.4.1 Definitions and Taxonomies *242*
9.1.4.1.1 The Matter of Definitions *242*
9.1.4.1.2 Thesauri and Taxonomies *243*
9.1.4.2 Aspects and Partitions *243*
9.1.4.2.1 Relevance, Powertypes, Aspects *244*
9.1.4.2.2 Modalities and Aspects *244*
9.1.4.3 Aspects and Schemas *245*
9.1.4.3.1 Thesauri and Taxonomies *245*
9.1.4.3.2 Taxonomies and Schemas *246*
9.2 Mapping Intensions *247*
9.2.1 Ontological Commitments *247*
9.2.1.1 Conceptual vs Ontological Commitments *247*
9.2.1.2 Heterogeneous Commitments *249*
9.2.1.3 Instanciation, Identification, Structures *249*
9.2.1.4 Agency and Communication *250*
9.2.1.5 Veracity *251*
9.2.2 Projections and The Mapping of Time *252*
9.2.2.1 Ontological Timescales *252*
9.2.2.2 Architecture Projections *253*
9.2.2.3 Domains Projections *255*
9.2.2.4 Project Projections *255*
9.3 Actionable Knowledge *255*

9.3.1 Ontologies Use Cases *256*
9.3.1.1 Principle *256*
9.3.1.2 Taxonomy *257*
9.3.1.2.1 Bounded Use Cases *257*
9.3.1.2.2 Open Use Cases *257*
9.3.2 Use Cases and Ontological Commitments *258*
9.3.2.1 Committed Use Cases *258*
9.3.2.1.1 Homogeneous Commitments *258*
9.3.2.1.2 Realization Commitments *259*
9.3.2.1.3 Abstractions Commitments *259*
9.3.2.2 Uncommitted Use Cases *259*
9.3.2.2.1 Explanatory Use Cases *259*
9.3.2.2.2 Functional Use Cases *260*
9.3.2.3 Roles and Environment *260*
Key Points *261*

10 KEOPS OWL Kernel *263*
10.1 OWL/Protégé Kernel *263*
10.1.1 Overview *263*
10.1.1.1 Protégé Class Hierarchy *263*
10.1.1.2 Protégé Properties *264*
10.1.1.2.1 Object Properties *264*
10.1.1.2.2 Data Properties and Datatypes *264*
10.1.1.2.3 Annotation Properties *264*
10.1.1.2.4 Individuals *265*
10.1.2 KEOPS Kernel Extensions *265*
10.1.2.1 Kernel Nodes *265*
10.1.2.2 Kernel Ontological Connectors *265*
10.1.2.3 Aspects, Features, Properties *266*
10.1.2.4 Semantics of Ontological Connectors *267*
10.2 Representing Territories *269*
10.2.1 Symbolic and Physical Environments *269*
10.2.1.1 Datasets and Documents *270*
10.2.1.2 Partitions, Taxonomies, Powertypes *270*
10.2.2 Representation of Variants *272*
10.2.2.1 Structured Connectors *273*
10.2.2.1.1 Inbound Connectors *273*
10.2.2.1.2 Outbound Connectors *273*
10.2.2.2 Combining Structures *274*
10.2.3 Constraints, Expressions, Rules *275*
10.2.3.1 Constraints Through Connectors *276*

10.2.3.2 Nominal Footprints and Semantic Layers *277*
10.2.3.3 Constraints Through Expressions *278*
10.2.3.4 Inference Rules *279*
10.3 Representing Intents *279*
10.3.1 Agents and Goals *280*
10.3.1.1 Agents, Actors, Roles *280*
10.3.1.2 Goals *281*
10.3.2 Projects *281*
10.3.2.1 Projects *281*
10.3.2.2 Process and Work Units *282*
10.3.3 A Sample of Variants *283*
10.4 Representing Abstractions *284*
10.4.1 Modalities *284*
10.4.1.1 Ontological Modalities *285*
10.4.1.2 Grammatical Modalities *285*
10.4.1.3 Representation Modalities *286*
10.4.2 Patterns *287*
10.4.2.1 Blueprints (Concepts) *287*
10.4.2.2 Profiles (Domains) *287*
10.4.2.3 Templates (Models) *288*
Key Points *288*

Conclusion *289*
Annexes *291*
Annex 1 Summary of Notations *293*
Annex 2 Acronyms and Glossary *295*
Annex 3 Bibliography *309*
Index *315*

Preamble

Ontologies are knowledge commitments. As commitments, they must rely on some agreed-upon symbolic representation. When addressing knowledge, ontologies can focus on specific domains or aim at fundamental entities. Both specific and fundamental approaches are based on the assumption that inherent truths can be universally agreed upon. At first, this undertaking may seem both legitimate and realistic, until allegedly fundamental entities and actual things must be reconciled. Then facts turn rebellious and make their point: truth is a matter of faith, and whatever ontologists and believers may think, faiths are, by nature, singular.

On that account, this book adopts a pragmatic Cartesian approach governed by doubt. Instead of pursuing ontological rainbows, it introduces an ontological framework that allows for cognizant mappings of territories and provides a reliable compass to navigate the galaxies of facts, concepts, and categories. More critically, ontological prisms allow for the ecumenical engineering of knowledge and can serve as blueprints for agentic cognition.

Ontological prisms are framed by philosophical, cognitive, and functional perspectives.

From a philosophical perspective, along with the Cartesian agnosticism mentioned above, ontological prisms borrow a few tenets from Ludwig Wittgenstein. First, they are not so much about ontologies as about an ecumenical framework that could support the development of possible ontologies. Then, distancing itself from existential-phenomenological and analytical traditions, the focus is shifted from things and beings to facts, with an explicit distinction between knowledge and the formal and semantic capabilities of languages. Finally, categories are crossed with facts and concepts in order to address reasoning and engineering concerns.

From a cognitive perspective, ontological prisms are defined by their opposition to the tenets of disembodied minds and the correspondence theory of truth. That's because assuming that minds produce concepts independently of their

biological reality would a priori deny any relationship between connectionist and symbolic cognition. Likewise, ontological prisms allow no room for an inherent correspondence between the names of concepts and their essence. Concepts, whether as mental images or symbolic artifacts, are human cultural products, and there is no reason to assume any kind of inherent attachment between symbols and concepts. Hence, the collective perspective is adopted by this book regarding cognition and knowledge.

From a functional perspective, the objective is to describe an ecumenical approach to the engineering of knowledge as defined by Davis, Shrobe, and Szolovits in their seminal paper (see bibliography):

1. Surrogates: manage the symbolic counterparts of objects, events and relationships identified in context and pertaining to concerns.
2. Ontological commitments: maintain a set of consistent statements about the categories of things that may exist in the domains under consideration.
3. Fragmentary theory of intelligent reasoning: supports actionable representations of what the things can do or can be done with.
4. Medium for efficient computation: make knowledge understandable by computers and support smooth learning curves.
5. Medium for human expression: improves the communication between specific domain experts and generic knowledge managers.

The book is organized into three parts. The first part presents the theoretical foundations of ontological prisms (Chapter 1), their three-pronged architecture (Chapter 2), interoperability (Chapter 3), and their integration within organizations (Chapter 4). The second part takes a functional perspective and focuses on the major cognitive capabilities of ontological prisms: language (Chapter 5), reasoning (Chapter 6), and decision-making (Chapter 7). The third part considers the benefits of ontological prisms in relation to systems (Chapter 8) and knowledge (Chapter 9) engineering. Chapter 10 introduces an OWL kernel that implements ontological prisms using the Protégé online platform.

Finally, two cautionary remarks. First, in line with its agnostic and ecumenical objectives, this book will not aim for universal definitions but will instead opt for clear and unambiguous ones that can be used effectively, consistently, and comprehensively. Then, readers must be aware that the engineering of knowledge is not a kindergarten activity, as its raison d'être is to address the complexity of uncharted territories.

About the Author

Rémy Fannader is an IT professional with a background in social sciences, economics, operations research, and artificial intelligence. He has over 35 years of experience, primarily in consulting, across various industries, including banking, insurance, and software engineering. He has contributed to several European Union research projects and has published four books, along with more than two hundred in-depth articles on systems modeling, enterprise architecture, artificial intelligence, and knowledge management. These works are available on his Caminao website, which has garnered over 700,000 views, and have been discussed through his LinkedIn posts by more than 7000 followers.

Part I

1

Current Approaches and Paradigm Shift

The world is the totality of facts, not of things

Ludwig Wittgenstein

Ontologies can be traced back to the moment humans were able to write down their thoughts, with Aristotle often regarded as a figurehead. As a branch of philosophy, the aim of ontology was to account for the essence of things. Being a philosophical discipline, this aim can be rephrased as being explicit about language ("what can be said about essence") and knowledge ("what can be known about essence"). These clarifications lead to epistemology, the official tutor of ontology tasked with the validation of ontological statements. That has been the philosopher's perspective. In practice, ontologies have been largely ignored until they reappeared alongside artificial intelligence (AI).

1.1 Ontologies and Artificial Intelligence

1.1.1 A Brief and Personal Perspective on Artificial Intelligence Past and Present

From a pragmatic perspective, intelligence can be broadly defined as the ability of agents, whether living or mechanical, to address issues presented in real, formal, and/or philosophical terms. When issues are self-contained, they can be tackled through problem-solving. However, when they are unbounded and involve environments, parties, and stakes, they must be addressed through broader cognitive processes involving observations, reason, and judgment. That double perspective, realms concerned and parties involved, should provide a benchmark for AI.

1.1.1.1 Mechanical Brains

While the Golem, Mechanical Turk, and other anthropomorphic creatures may be seen as forerunners of AI, and more particularly agentic AI, factual and rational

evidence points to computing machines evolving into digital brains. In this regard, two pioneers are particularly representative of two basic lineages of AI: Alan Turing and Claude Shannon.

Turing may be considered first among equals for his wide-ranging contributions, which encompass the mathematics of computing machines, his quasi-prophetic vision of their learning capabilities, and the design of an eponymous test aimed at distinguishing between human and mechanical speakers. Shannon, for his part, represents the second native lineage of AI, focusing less on digital brains and more on their core processing capabilities. As a counterpart to Turing's work on the theory of computation, Shannon concentrated on information theory, distilling information into its elementary components.

1.1.1.2 Symbolic vs Connectionist Tracks

The distinction between AI symbolic and connectionist tracks can serve as a compass for exploring AI developments. First, because the divide can be clearly identified at both conceptual and technical levels; second, because it has been determinant from the beginning; and finally, because the crossing of paths is now at the core of critical advances.

In outline, symbolic AI operates through languages and considers intelligence as a capability of the mind that harnesses communication and cognition. In contrast, connectionist (nonsymbolic or neural) AI views intelligence as a capability of the brain that can be fully understood in terms of computation. In between, the aim of generative artificial intelligence (GenAI) is to apply connectionist methods to symbolic content.

The opening of the symbolic track is a collective achievement of, among others, Allen Newell, Herbert A. Simon, Marvin Minsky, John McCarthy, and J.C. Shaw, who sought to address the mind/body issue. To that end, they brought together a wide array of disciplines, some computational – typically logic (e.g., the "Logic Theorist" developed by Newell, Simon, and Shaw) and linguistics (e.g., Noam Chomsky's generative grammar) – and others cognitive, from psychology and philosophy (Minsky, McCarthy) to neurosciences. The connectionist track first appeared through neuroscience, a comprehensive approach to human cognition addressing symbolic as well as connectionist dimensions, with perceptrons (Frank Rosenblatt) serving as an ephemeral landmark. But the use of neural networks and machine learning to train digital brains was constrained by a lack of computing power and training materials. For a time, symbolic approaches were the kings of the road. In hindsight, Shannon's information theory appears as a signpost for an alternative, still unexplored, cybernetic path merging symbolic and connectionist AI tracks.

1.1.1.3 Language and Common Sense

To a certain extent, symbolic approaches to AI have turned paths into useful roads with expert systems combining a corpus of domain knowledge with rule engines. Following the development of convincing applications in domains like organic

chemistry (DENDRAL) or blood diseases (MYCIN), large companies began to invest in expert systems targeting a wide array of domains. Alongside unequivocal successes like R1, developed for the Digital Equipment Corporation, the outcomes were mixed depending on the scope: positive for siloed expert systems, debatable for the extension of siloed solutions, and a patent failure for attempts to build multipurpose intelligent systems, as epitomized by the Fifth Generation Computer Systems (FGCS) project launched by the Japanese Ministry of International Trade and Industry in 1982.

In a specific way, and setting apart systemic flaws, the collapse of the FGCS highlights two of the key challenges in AI: language and common sense. The FGCS has been designed as a reasoning system built using Prolog, a logic programming language developed by Alain Colmerauer, but intelligence cannot be reduced to logic, as it requires common sense to serve as the glue between domain-specific reasoning. As far as AI is concerned, the limitation in expressiveness is not specific to logical languages like Prolog; it also pertains, albeit to a lesser degree, to Lisp, a purely symbolic and declarative (as opposed to imperative) programming language developed by McCarthy. That's because expressing common sense goes beyond language and highlights the way language relates to knowledge.

The question of common sense, or innate reasoning ability, arises from two directions: in linguistics, it pertains to the semantics that underpin grammatical structures, and in knowledge, it marks the limits of symbolic representation. While Chomsky has been instrumental in establishing a continuum between the grammars of computer languages and the semantics of natural languages, his generative grammars fall short in capturing the expressiveness of natural languages. The two-pronged questioning about common sense, from both linguistics and knowledge, brings some clarity to the AI debate: if common sense goes beyond grammatical expression, typically through pragmatics, little room remains for a comprehensive and consistent symbolic representation of discourses. As a corollary, AI systems would have to accommodate nonsymbolic content, bringing connectionism back into play.

1.1.1.4 Mimicry and Rote Learning

From its inception, AI has been hinged on the assumption that machines could learn, either through self-improved symbolic reasoning (e.g., using Lisp or Prolog languages) or nonsymbolic neural networks (connectionist approaches). In practice, advances have long been significantly hampered by the limitations in available resources, particularly in computing power and training data. As these limitations were first gradually and then drastically reduced – through dedicated chips for the former and accumulated internet content for the latter – connectionist solutions have significantly enhanced the learning capabilities of AI systems, until enhancements become breakthroughs, enabling generative language models (GenLMs) to bypass grammars and semantics.

Initially, and not surprisingly, this connectionist boost has focused on the parallel computation of graphical contents in domains such as pattern recognition and games, as illustrated by DeepMind's AlphaGo. Then, rapid advances in algorithms and dedicated chips have driven what is commonly referred to as Deep learning into a wide range of fields, such as natural language processing, clinical diagnosis, credit rating, and autonomous vehicles. However, as these systems often rely on mimicry and rote learning methods, their rapid and often unrestrained expansion in sensitive decision-making domains has raised serious doubts regarding their reliability, not to mention ethical concerns about the alignment of artificial behaviors with human values. And these concerns are being amplified by the substitution of opaque connectionist solutions for traceable symbolic ones.

1.1.1.5 Agents in No Mind's Land

By and large, humans can learn through rote memorization, mimicry, examples, reasoning, or even faith. Machines, for their part, have a convincing track record in the first three routines, a questionable one as cogent learners, and none as believers – until they became acquainted with GenLMs.

Compared to what may already be labeled as "old-fashioned" machine learning, GenLMs do not learn from deeds but from words, which is literally a matter of faith: placing trust in words with the assumption that meanings and truths can emerge from their primal soup, provided it is massive enough. According to this AI model, and the correspondence theory of truth (mentioned in the preamble), cognitive processes can be simply bypassed, with reasoning replaced by nominal correlations and judgments by stochastic probabilities. Nevertheless, GenLMs can be very effective as long as they are operated within bounded contexts that ensure unambiguous semantics (the correspondence theory of truth) and trusted references. When these caveats are taken into account, there is a wide range of opportunities for what is commonly known as GenLM co-pilots, from software engineering (where programming languages epitomize bounded contexts) to editing (assuming siloed domains, cognizant users, and reliable references).

The way GenLMs are integrated into systems may represent a turning point for GenAI, as many have already noticed, due to significant issues being raised at different levels. Starting with engineers, a consensus is emerging about the limited reasoning capabilities of GenLMs and, consequently, the need for a "significant other" to which GenLMs could offload knowledge-intensive tasks. Next is user experience: when decisions are involved, full transparency and traceability are necessary – something that is currently out of reach with GenAI. When set in a broader perspective, these flaws point to a necessary built-in distinction between language and knowledge.

Investors are also concerned about the scale of financing and the elusive tipping point. Much like in Aesop's Fable, the performance of GenLMs (the rabbit

in the fable) initially appeared to improve so rapidly that achieving a level of reliability on par with human performance (the tortoise in the fable) was thought to be just a matter of months. However, costs began to rise as quickly, if not faster, than performance, and the goalposts seemed to shift, even if only by inches, when success appeared to be within reach. Compounding financial concerns, energy consumption (and its carbon footprint) began to raise sustainability alarms for the entire GenAI project.

Most attempts to overcome these issues rely on collaborations between GenAI and knowledge-based agents, yet without being too specific about the neuro-symbolic architecture intended to support such collaborations. As it stands, the challenging nature of neuro-symbolic architectures reflects the foundational issue of AI, namely the interplay between neural networks and symbolic processing, in other words, between brains and minds. The aim of ontological prisms is to offer an effective as well as elegant approach to that issue.

1.1.2 An Ontological Manifesto

As long as generative approaches remain at the forefront of AI advancements, they also face critical issues, the most significant of which is access to knowledge, epitomized by ontologies. In this regard, an ontological manifesto may help lay the groundwork.

1.1.2.1 Facts and Data

1.1.2.1.1 Facts Are Not Information

Actual issues are rooted in facts, but facts are not given as manna from heaven; they can only be observed through data, which usually offers a partial, biased, and/or temporal reflection of what happened and/or is happening.

1.1.2.1.2 Neural Connectors Represent Facts, Not Meanings

Regardless of the nature of the nodes in neural networks, their connectors carry quantitative values devoid of meaning. This is true for brain neurons, telecommunication nodes, and also for the weights on connectors between tokens in documents.

1.1.2.2 Language and Reason

1.1.2.2.1 Information Is Not Knowledge

Information models are one-sided (no parties directly involved) and bounded (unambiguous semantics) and can thus be expounded and dealt with independently of real or philosophical ones; to that end, they can be expressed through symbolic languages used to define categories and rules representing hypothetical instances.

1.1.2.2.2 Language Is Not Knowledge

Compared to formal issues, philosophical ones may involve different parties with stakes that are possibly set across various domains. This entails the use of natural language for communication and the representation of relevant values, intents, and outcomes. Ignoring the distinction between expressing the issues (language) and addressing them (knowledge) reduces intelligence to mere algorithms.

1.1.2.2.3 There Is No Truth in Languages

In language there are only differences. Even more important: a difference generally implies positive terms between which the difference is set up; but in language there are only differences without positive terms. Whether we take the signified or the signifier, language has neither ideas nor sounds that existed before the linguistic system, but only conceptual and phonic differences that have been issued from the system. Ferdinand de Saussure *Course in General Linguistics*.

1.1.2.2.4 There Is No Reason in Data

Data consists of labeled observations from physical or symbolic environments. At the inception of cognitive processes, data is meant to be free of any assumptions regarding meanings and/or causations; otherwise, it would introduce built-in and circular reasoning.

1.1.2.3 Judgment and Computation

1.1.2.3.1 Reason Is Not Judgment

The distinction between reason (information) and judgment (knowledge) is intended to align with the difference between mechanical and living agents. This two-pronged distinction serves as a fulcrum for the AI project and its associated ethics.

1.1.2.3.2 Judgments Begin Where Algorithms End

The distinction between judgment and algorithms should serve as a litmus test to differentiate between human and AI. Judgment is necessary for issues involving parties with intentions (decision-making), as opposed to self-contained issues (problem-solving).

1.1.2.3.3 Knowledge Graphs Are Not Knowledge

Knowledge graphs (KGs) are to knowledge what family trees are to families and wine bottles are to wine: containers. KGs are symbolic representations of real (e.g., communication networks), conceptual (e.g., entities and relationships), or virtual (e.g., intents and plans) realms. Like languages, such as the Web Ontology Language (OWL), they should not be confused with what they represent.

1.1.2.3.4 Intelligence Cannot Emerge from Canned Words

Massive increases in available computing power and data hoards have triggered a surge of interest in generative technologies. But wild and irrational expectations have come with leaps of faith and sleight of hand:

- Leaps of faith, for the belief that knowledge can emerge from canned words.
- Sleights of hand, for getting rid of the distinction between language and knowledge, notwithstanding that it is at the core of the AI project.

These flaws induce inbreeding between language and knowledge, which could drive GenAI into degenerative quagmires.

1.2 Ontologies and Knowledge

Current approaches to ontologies can be categorized into two perspectives that may eventually converge: top-down or foundational approaches aim to identify universal categories that could be relevant to all domains, while bottom-up or empirical approaches begin with specific domains and attempt to generalize to abstract categories, hopefully meeting foundational approaches halfway.

1.2.1 Foundational Ontologies

The aim of foundational (also known as upper-level or conceptual) approaches is to identify generic entities intended to be applied uniformly across knowledge domains. Beyond lexical variations, ontological languages developed by these approaches are conceptual extensions of traditional modeling languages like Unified Modeling Language (UML). We are using two main ontologies to illustrate the foundational approaches, along with additional examples for spatiotemporal and goal-driven extensions.

1.2.1.1 Unified Foundational Ontology

The Unified Foundational Ontology (UFO) framework, along with the associated OntoUML language (see Bibliography), organizes conceptual modeling in terms of identified individuals and universal classifiers (types or categories). The former refers to things that can be uniquely identified in reality, while the latter pertains to sets of features that can be jointly realized by groups of individuals. Terms may vary, but things are commonly categorized into endurant (or continuant) and perdurant (or occurrent) types. The former refers to things that can be identified independently of time, while the latter pertains to things with time-related identity (also known as phenomena), typically events and situations. A distinction is made between substantial and moment-endurant types: the former has its own

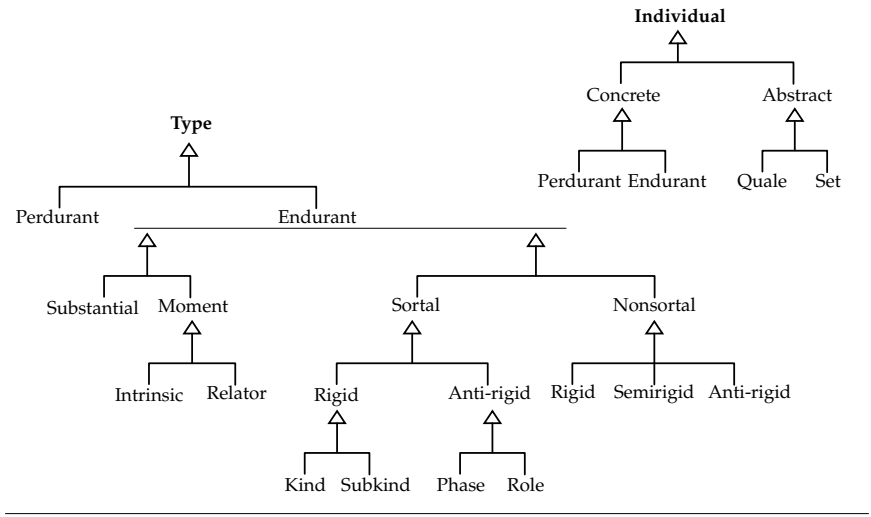


Figure 1.1 Main UFO individuals and types.

identity principle, while the latter is identified through the manifestations of the former. Moments can describe objects' intrinsic features or relationships between objects (Figure 1.1).

UFO uses the instantiation and identification principles to distinguish between rigid types, which characterize objects all along their life cycle, and anti-rigid types, which pertain to roles held by objects. Concomitantly, UFO crosses the rigidity distinction with a sortal distinction addressing the nature of instantiation, notably categories (rigid nonsortal) for types that cannot be instantiated. The identification principle is also used to distinguish between kinds and subkinds, instances of the latter being subsets of instances of the former. Besides the endurant/perdurant distinction of concrete individuals, UFO introduces abstract individuals corresponding to sets and quality spaces and structures.

1.2.1.2 Descriptive Ontology for Linguistic and Cognitive Engineering

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (see Bibliography) can be viewed as the most accomplished attempt among foundational ontologies. On the one hand, it's rooted in a formal theory in first-order modal logic; on the other hand, the more recent OWL version extends the scope of the established endurant/perdurant constructs through social and temporal dimensions while relying on the Common Logic Interchange Format (CLIF) and OWL2 to support formal consistency. More importantly, DOLCE establishes the parthood model (or mereology) as the basis for its knowledge representation,

marking a significant departure from the bias often introduced in conceptual representations through the implicit use of object-oriented (OO) inheritance semantics. Nevertheless, despite its more principled approach, the benefits that lower-level ontologies could achieve from the use of upper-level DOLCE are counterbalanced by the constraints imposed by their necessary alignment with the imperative legacy of essentially discretionary foundational distinctions.

1.2.1.3 Typical Extensions

1.2.1.3.1 Unified Foundational Ontology Extensions

Attempts have been made to go beyond foundational entities with generic endurants (UFO-A), perdurants (UFO-B), and social and intentional entities (UFO-C). While UFO-A and UFO-B add little to the UFO basic entities, that's not the case for UFO-C, which is not an extension, but a complement meant to address UFO's initial shortcomings regarding symbolic entities such as actions, agents, or commitments.

1.2.1.3.2 Basic Formal Ontology

Despite differences in terminology (such as "continuant" for "endurant" and "occurrent" for "perdurant"), the concepts of the Basic Formal Ontology (BFO) are broadly aligned with UFO, with the exception of spatiotemporal adjuncts. The significance of these adjuncts, aside from the specifics of the spatial region type, is that they highlight a critical blind spot in the conceptual paradigm behind foundational approaches, namely immaterial entities.

Compared to UFO, which addresses the issue through an unwieldy combination of sortal and rigidity schemes, BFO is explicit about immaterial entities but reduces them to mere representations of dimensions, overlooking the broader issue of symbolic entities. This original flaw in conceptual ontologies can be explained by their alignment with the classical philosophical assumption that reality is physical, a questionable premise for modeling today's social environments and business activities.

1.2.1.3.3 Tropos

Tropos does with actors and goals what BFO does with spatial regions: it introduces a subset of symbolic entities, yet without fully addressing the intrinsic caveats of the conceptual paradigm. Nevertheless, Tropos goes further by defining the actor type as an abstraction (or concept) set in three dimensions: organizational (realized through positions), behavioral (realized through roles), and intensional (realized through agents). By doing so, Tropos introduces an abstraction level between terms that are commonly used with overlapping semantics in ontology conceptual models, and thus takes a step toward all-inclusive ontological prisms and an epistemic distinction between facts, concepts, and categories.

1.2.1.3.4 Common Core Ontology

Common Core Ontology (CCO) (see Bibliography) is a mid-level ontological framework that aims to extend foundational approaches with transverse domains such as space, time, currencies, events, and agency. CCO is not intended to provide an all-inclusive knowledge engineering framework; rather, it aims to establish guardrails between foundational ontologies and domain-specific ones. Despite the intrinsic merits of its modules, CCO headways have been hampered by a mix of transverse (e.g., space, time, currency) and generic (e.g., information, artifact, relations) modules. While such overlaps are of little consequence when modules are applied separately, they introduce conflicting semantics when lineages, often implemented through inheritance, are crossed.

1.2.1.4 Limits of Foundational Ontologies

Business adoption of foundational ontologies has been hindered by their restrictive conceptual perspective, which ignores the epistemic distinction between extensional (things and phenomena) and intensional (ideas, values, and intents) realms and consequently flattens multidimensional knowledge. These shortcomings can be illustrated through three main issues: blind spots, dimensions, and modalities.

1.2.1.4.1 Foundational Blind Spot

As noted above, conceptual ontologies are rooted in a classical school of thought that assumes reality is exclusively physical. This assumption creates a conceptual blind spot when the concept of essence is questioned. While this question involves broader philosophical issues, it can be narrowed down to the UFO sortal criterion, i.e., whether individuals can be counted or not. That counting characterization raises issues for the identity principle as well as for objects' structural continuity, revealing a double-blind spot: one between natural entities and designed artifacts, and another between physical and symbolic entities. Since these intrinsic differences in identification directly affect sortal definition and, consequently, universals, the distinction between physical and symbolic entities must be explicitly accounted for and intersected with the concepts of intension (essence, independent of individual realizations) and extension (individuals, independent of their essence). A few examples illustrate the issue (Table 1.1).

As attempted by UFO-C, it's always possible to find ontological descriptions mixing rigidity and sortal yardsticks, but the solutions turn out to be discretionary and unwieldy. BFO and Tropos have made some more principled inroads into these intersections with regard to spaces and roles, respectively, but without explicitly addressing the general issue. DOLCE makes some ground addressing the flaw by introducing a distinction between physical and nonphysical objects,

Table 1.1 Intensional and extensional realms.

	Physical	Symbolic
Intensions	GM Maize (artefact)	Cook (role)
	Dish (artefact)	Recipe (design)
	Crop (artefact)	Customer satisfaction (goal)
Extensions	Virus (natural)	Order (event)
	Maize (natural)	Meal (collection)
	Person (natural)	Region (dimension)

the latter being further categorized into mental and social objects. These social objects are then divided into agentic and nonagentic subsets. However, this partitioning complicates other related partitions: the epistemic one between symbolic and nonsymbolic objects; the fuzzy one between nonsymbolic objects and abstract particulars; the pragmatic and now critical one between biological, social, and digital agents; and the foundational distinction between endurants and perdurants, which is bound to be commonly straddled by agentic objects defined by roles. These multiple overlapping partitions introduce unnecessary complexities that become overwhelming at scale, hindering the reuse of DOLCE's high-level constructs by lower-level ontologies.

The inherent flaw in multiple branching attempts to address nonphysical blind spots is best understood by comparing DOLCE's abstract particulars to John F. Sowa's foundational distinction between concrete and abstract forms: with a simple shift, discretionary and confusing classifications are replaced by an essential distinction between extensional and intensional realms, respectively.

In line with the Sowa approach, ontological prisms introduce a two-pronged epistemic partitioning. First, there is a distinction between the intensional and extensional substance of physical entities, with the former rooted in human designs (artifacts) and the latter in nature, regardless of the origin. Second, there is a distinction between physical and symbolic environments, both of which are part of the extensional realm.

1.2.1.4.2 From Dimensions to Concepts

By introducing immaterial entities as a conceptual stepping stone to spatial regions, BFO introduces a boundary without further elucidation, except for spatial regions and, more generally, quantitative dimensions. Tropos extends the concept to qualitative dimensions, introducing organizational context within which roles can be defined. Interestingly, Tropos' definition of an actor in terms of organization, behavior, and physical agency can be generalized beyond the boundaries of

conceptual models so as to be aligned with the ontological triptych of facts, concepts, and categories:

- Actor, category as modeled with regard to an activity.
- Role, concept as defined by organizations, that may or may not be associated with a category.
- Agent, individual factually identified in the physical or symbolic environment.

However, introducing agents and goals has limited benefits without an explicit conceptual distinction between actual contexts (extensions) and mental representations (intensions).

1.2.1.4.3 Ontologies Are Not Models

So far, we have examined the limits of conceptual ontologies in relation to targeted content, specifically physical entities versus symbolic ones. Goals introduce a different kind of entity, which can be referred to as planned, expected, or virtual. These references can be placed in an actual context, typically involving time and/or an agent (with plans and expectations), or detached from actual realms (virtual, fictional, etc.). This creates room for unbounded intensional modalities (intents, goals, objectives, plans, projects, etc.) that can apply to any kind of physical or symbolic entity. CCOs have made some practical steps in that direction, but further advancements have been hindered by the cross-inheritance of transverse and generic ontologies. From a theoretical perspective, the limits of foundational approaches are illustrated by UFO-C attempts to address social entities without intensional semantics. Ontological modalities may help to overcome this difficulty.

1.2.2 Empirical Approaches and Meta-models

Contrary to foundational approaches that are often confined to academic or conceptual capsules, empirical approaches are, as a matter of course, rooted in actual business concerns and, consequently, must address heterogeneous ontological issues. This, in turn, may lead to attempts at more generic developments.

1.2.2.1 Domain-specific Ontologies

Paradoxically, while operational implementations of the conceptual (top-down) approach to ontologies remain mostly peripheral, bottom-up developments targeting specific domains may point to broader perspectives. Admittedly, announcements can be misleading, as a significant portion relies on customary conceptual models established in silos and rebranded as ontologies. Healthcare represents an exception due to (i) the growing imbalance between needs and resources, (ii) compelling operational environments, (iii) the necessary interoperability

Table 1.2 The M*R/K reference framework makes a distinction between knowledge (right) and formalisms (left).

	Representation formalism	Knowledge categories and abstractions
3	Meta languages	Meta knowledge types
2	Symbolic and natural languages	Knowledge universal types
1	Modeling languages	Medical domain knowledge
0	Digital twins	Ground facts

between organizational, operational, and scientific domains, and (iv) the potential medical, social, and financial benefits of digital integration and machine learning technologies.

Without delving into the various developments addressing domain-specific knowledge in toxicology, clinical protocols, biology, demographics, traditional medicine, and so on, there are a number of attempts to address their interoperability through knowledge reference frameworks, as exemplified by the "Multilayer Metamodel for Representation and Knowledge" (M*R/K).

The M*R/K reference framework is organized into four layers: facts on the ground, domain-specific knowledge, universal knowledge, and meta-models. Each layer is represented with dedicated formalisms (Table 1.2).

That knowledge architecture offers a dual benefit for interoperability:

1. The distinction between content and representations separates shallow (representation level) interoperability between systems from deep (knowledge level) interoperability across domains.
2. The distinction between facts and knowledge enables the interoperability of domain-specific knowledge, independent of organizational as well as technical boundaries.

However, the proposed framework leaves key issues regarding layers and representations unresolved.

On the knowledge (right) side of the framework, the bottom-up mapping principle linking lower to upper layers is uniformly defined as "instances of." The semantics of the principle is straightforward when applied between ground facts and medical knowledge; however, the principle loses much of its relevance between domain-specific and universal types, and loses all relevance between universal and meta types because neither type is consistently defined.

Concomitantly, on the left (representation) side of the framework, the universal and meta layers appear to lack consistency: the former mixes programming and natural languages, while the latter combines syntactic and semantic constructs.

This lack of consistency induces confusion between semantics, which is common to both programming and natural languages, and pragmatics, which is specific to natural languages. This confusion between semantics and pragmatics undermines the universal representation layer and, consequently, the distinction between the universal and meta layers of knowledge.

These fault lines illustrate the main challenge of ecumenical ontological frameworks and, more generally, of universal knowledge representation, namely the semantics of abstractions between models and meta-models, and the juxtaposition of language and knowledge.

1.2.2.2 Abstractions, Reuse, and the Limits of Meta-models

CCOs for top-down and M*R/K for bottom-up approaches illustrate how the lack of effective and reliable abstraction schemes can prevent the development of ontological frameworks.

CCO illustrates the inherent limits of core ontologies, whose reuse is confined to generic (e.g., artifacts) or domain-specific (e.g., currency) swimlanes, without clear and comprehensive mechanisms that allow for sound combined inheritance.

Alternatively, the blurred nature of the M*R/K upper layers reflects the intrinsic shortcomings of meta-models. Just as models represent sets of actual instances represented by models' categories, meta-models are meant to represent sets of categories common in sets of models. It follows that, unlike models, meta-models are by design detached from instances in actual realms. When it's assumed that medical knowledge types are "instances" of knowledge universal types, the term "instance" takes a different meaning, just like with the semantics of universal "types." The justifications for the M*R/K universal and meta layers do not reach that point; instead, they mix operational, descriptive, and institutional arguments without establishing a clear rationale.

Nevertheless, the M*R/K framework introduces two significant points. First and foremost, the framework is built on the distinction between knowledge and representations. Even if its effectiveness is undermined by a faulty abstraction scale, that distinction represents a major improvement compared to other approaches.

Second, and paradoxically, the arguments used to justify the upper levels of the proposed abstraction scale point in fact to ontological modalities as an alternative to traditional meta-model abstractions; these arguments can be rephrased in terms of:

- Epistemic modalities, addressing the status of both facts (observed, asserted, assessed, deduced, managed, ...) and knowledge (abstract, concrete, nominal, virtual, ...).
- Reasoning modalities, addressing the realm (actual or symbolic) and truth value (nominal, empiric, formal) of discourses.

- Decision-making modalities, framing judgmental issues with epistemic and reasoning modalities.

These universals differ from the ones defined by conceptual approaches: they do not represent entities but rather ontological properties. Consequently, (i) they are not introduced through meta-models but directly through dedicated language constructs, and (ii) since these language constructs are dedicated, they can be safely mixed with modeling languages without risking semantic interference.

1.2.3 What to Expect

In contrast to foundational approaches, which are constrained by a constricted conceptual premise, the progress of empirical approaches is driven by the challenges of overlapping domains, not to mention financial incentives. The necessary integration of heterogeneous activities in industries such as healthcare or urban mobility, along with positive prospects for returns on investment, highlights the need for ontological frameworks that could support knowledge engineering across business contexts. At the same time, the pervasiveness of GenLMs has highlighted their polyvalence as well as their cognitive limitations, underscoring the need for cognizant and cogent agents to take on knowledge-intensive tasks. Such tasks are generally carried out through KGs, which can be seen as the Swiss Army knife of ontology development. While KGs provide a universal and highly effective solution for representing complex structures, they must be fleshed out with domain-specific content, which is commonly done through the Simple Knowledge Organization System (SKOS). SKOS is a knowledge-driven documentation standard developed and maintained by the World Wide Web Consortium (W3C), along with complementary standards such as the OWL and the Resource Description Framework . Besides its native complementarity with OWL, the SKOS representation of thesauri, taxonomies, and domains can be neatly aligned with the interoperability sections of the ontological prisms introduced below.

More generally, developments in KGs gain momentum from two directions. On one hand, the aim is to extend the functionalities of document management systems, typically incorporating KGs and language models. On the other hand, the aim is to package KG editors and language models with services. Both trends appear to lead toward collaborative architectures of intelligent agents, some dedicated to natural language processing, some to document management, and some to knowledge-intensive tasks. Lest they remain specific silos or turn into Babelian towers, these architectures will need ecumenical ontologies to support effective collaboration.

1.3 The Ontological Prisms Paradigm

Ontological prisms are established across three fundamental perspectives: semiotics, philosophy, and engineering.

1.3.1 Ontologies and Semiotics

As noted above, the main drawback of foundational ontologies is their limited conceptual perspective: ontological categories are directly defined from semiotic triangles that associate things (a hospital) with signs ("Hospital") to denote them, and logographic (H) or literal (Hospital) symbols representing concepts and/or categories; such a two-dimensional representation of things conflates models (categories) and ontologies (concepts). Hence, the benefits of shifting to a three-dimensional representation of things, concepts, and categories (Figure 1.2).

Unfolding a third dimension for categories enables an explicit distinction between universals (concepts) and their representations (categories), in line with the M*R/K framework mentioned above. It also allows for an explicit representation of designed and/or symbolic objects and their corresponding categories. As a corollary, the partitioning of objects can be carried out independently of the naming of concepts; the former is realized through taxonomies (things/categories),

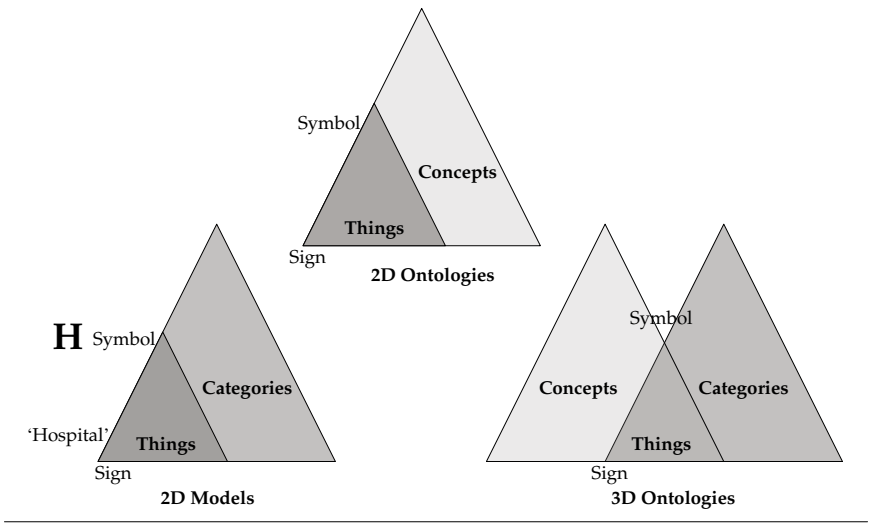


Figure 1.2 From semiotics to ontologies.

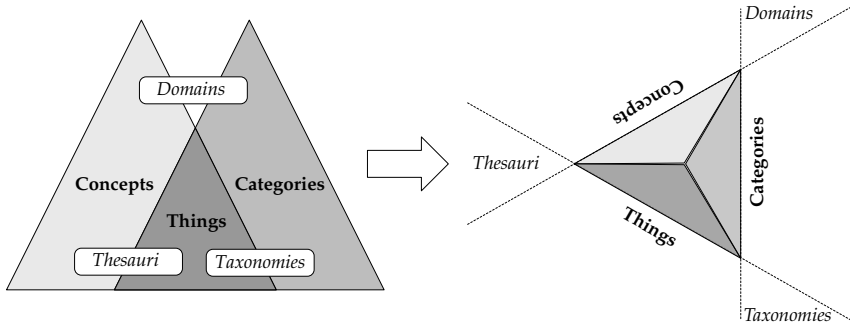


Figure 1.3 A sleight of hand rearranging pyramids into prisms.

while the latter is achieved through thesauri (signs/meanings). The third intersection pertains to domains, sometimes referred to as ontologies, which are used to align the management of concepts (for knowledge) and categories (for representations).

The significance of the interplay between realms, and the role of thesauri, taxonomies, and domains as gears realizing their alignment, appears fully through a simple sleight of hand that rearranges the pyramids into prisms (Figure 1.3).

1.3.2 Ontologies and Philosophy

Despite their roots in classical philosophy, ontologies are usually understood as upgraded conceptual models, often without further consideration of epistemic issues – the branch of philosophy that studies the nature, sources, validity, and edges of knowledge. Ontological prisms can reconcile these issues with classical philosophy.

Philosophical schools of thought can be broadly categorized into three main perspectives introduced by ancient Greek philosophers (Figure 1.4):

- Empiricism, for knowledge achieved through the observation of physical, social, and political facts.
- Idealism, for knowledge achieved through conscientious conceptual (aka mental) judgment.
- Rationalism, for knowledge achieved through reason applied to agreed-upon symbolic constructs.

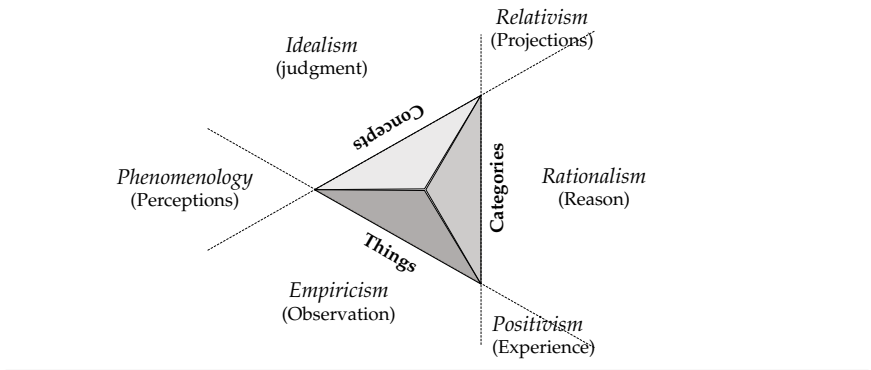


Figure 1.4 Philosophical perspectives.

These mainstays can be further refined through their overlaps, typically:

- Phenomenology, for knowledge achieved through the mental representation of perceived physical, social, and political facts.
- Relativism, for knowledge achieved by projecting judgment on reason.
- Positivism, for knowledge achieved by experiencing facts through reason.

As they stand, these perspectives can also be aligned with Kant's works on Practical Reason (things), Power of Judgment (concepts), and Pure Reason (categories); or with Spinoza's learning taxonomy of Senses, Judgment, and Reason, respectively. As a practical counterpart to that philosophical versatility, ontological prisms can also serve as an ecumenical knowledge engineering platform.

1.3.3 Ontologies at Work

Amidst the bustle generated by AI, fundamental ontological issues remain much the same as those articulated by Aristotle: are there foundational categories from which any other necessary categories could be derived? If so, how can we identify them? Is there a conceptual difference between foundational categories and top or upper-level categories? If so, where should we draw the line and stop passing the buck up the abstraction ladder? Ontological prisms can help organize these questions into three representative workflows (Figure 1.5):

- Aristotle's foundational ontologies put the focus on essential concepts ("ur-elements").
- Engineering approaches concentrate on the categories and taxonomies supporting the representation of specific domains of concern.
- Frameworks, the focus of this book, address the sharing of concepts and representations.

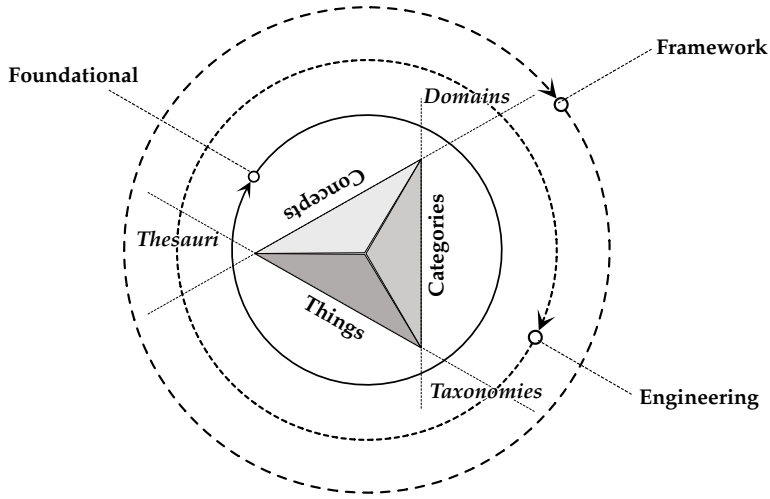


Figure 1.5 Ontological workflows: foundational concepts (upper level); engineering categories (applications); frameworks (cross ontologies).

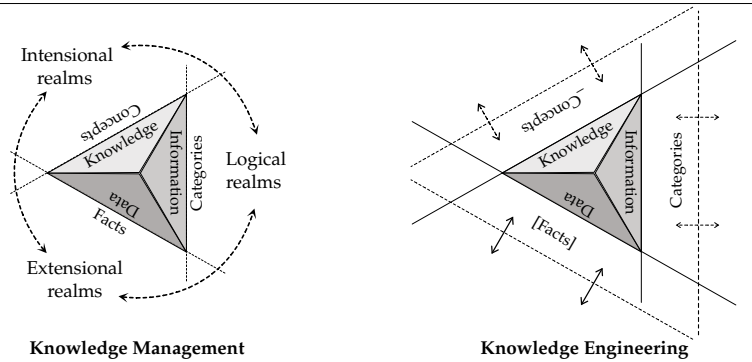


Figure 1.6 Knowledge management and engineering: pushing the edge of knowledge.

In line with the framework perspective, the next chapters will detail (i) the representation of shared concepts and (ii) their interoperability and management.

This book will alternate between ontological and systemic views, with the former focused on facts (instead of things), categories, and concepts, and the latter on data, information, and knowledge. While these views are isomorphic and can be switched depending on the issues, the distinction is particularly relevant with regard to knowledge engineering processes (Figure 1.6).

In that case, ontological prisms operate like Jacquard looms, machines used to weave complex patterns and motifs into textiles. The modus operandi, which will be detailed in Chapter 9, utilizes three rods, one for each realm, that can be moved parallel to the boundaries of the prisms. During the first phase, the inward movements determine facts, categories, and concepts from nondescript entries. In the second phase, which may overlap with the first, the inward movements rely on intersecting gears to translate facts, categories, and concepts into interoperable data, information, and knowledge, respectively.

1.3.4 Process and Case Study

The aim of the case study is to illustrate the use of ontological prisms, rather than to develop any specific ontology. In addition to the familiarity of the topics, the choice of food, cooking, and food services is intended to emphasize the sharing of knowledge and the integration of physical, functional, and organizational dimensions across ontologies (Figure 1.6).

1.3.4.1 Physical

Food and cooking are primarily physical domains, encompassing both tangible items and processes. Furthermore, culinary processes are all-inclusive, as they are designed to integrate material, social, and even molecular dimensions. In order to support a comprehensive integration of exchanges, heterogeneous ontologies must ensure the consistency of analog, digital, and symbolic representations.

1.3.4.2 Functional

Besides regulatory constraints, farming, cooking, and food services are meant to combine knowledge in fields such as chemistry, biology, culinary, and business, usually from a plurality of sources. The transparency and traceability of food chains and services cannot be achieved without a clear and comprehensive distinction between the respective contributions of the language and knowledge layers.

1.3.4.3 Organizational

From farms to diners, food is handled by various organizations serving different objectives and subject to different regulatory regimes. This results in a plurality of overlapping domains with heterogeneous authority, semantics, and rules (Figure 1.7).

While representation semantics differ for facts, concepts, and categories, their interoperability can be achieved through thesauri (meanings), taxonomies (partitions), and domains (responsibilities).

1.3.5 M*R/K and Ontological Prisms

Taking the M*R/K framework as representative of attempts to bridge the gap between empirical and foundational perspectives, we conclude the chapter with a parallel with ontological prisms (Figure 1.8).

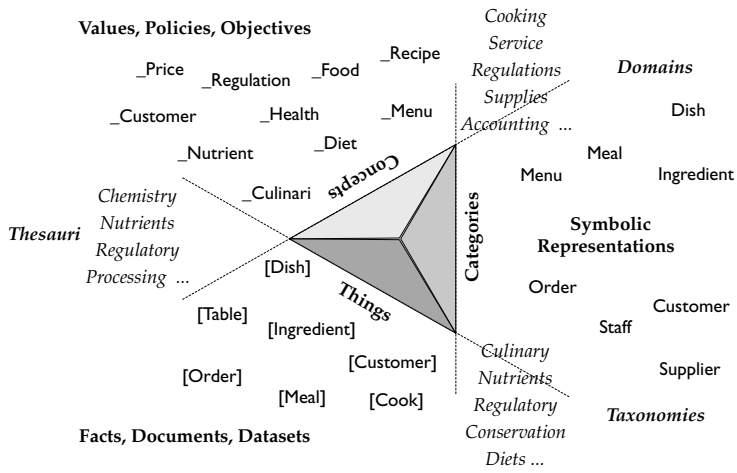


Figure 1.7 Case study: [things], _concepts, categories.

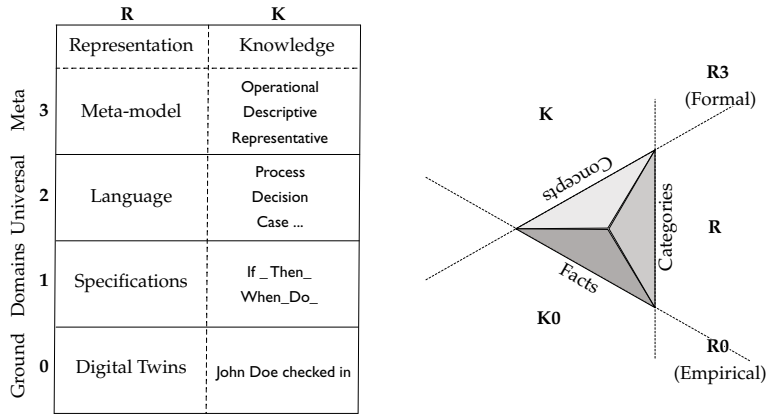


Figure 1.8 M*R/K (left) through an ontological prism (right).

While the key distinction between knowledge and representations is maintained, abstraction layers concerning knowledge are ignored, and their semantics are redefined for representations. The rationale is that the concept of abstraction can only be defined for representations, not for knowledge. Applying abstraction semantics uniformly to both knowledge and representations would imply that knowledge is something that can be neatly and definitively packaged independently of purposes and users. Moreover, once the focus shifts to representations,

abstraction layers can be defined in terms of languages: empirical (R0), modeling (R1), and formal (R3). On the knowledge side, the only distinction is between facts and concepts, which is at the core of knowledge engineering.

Key Points

The aim of this book is to introduce an agnostic and ecumenical ontology engineering framework. "Agnostic" means that it makes no assumptions about the nature or essence of things, while "ecumenical" signifies that it accommodates both connectionist and symbolic tenets regarding intelligence, whether natural or artificial.

Current approaches can be organized into two groups. Top-down foundational ontologies aim to define universal concepts intended to provide a common framework for dedicated ontologies and shared roots for domain-specific entities. However, these approaches are hindered by the difficulty of determining essential identity principles and instantiation modalities. It follows that attempts to map domain-specific entities to foundational ones tend to be at the same time unwieldy and discretionary, thereby offsetting most of the benefits of reusing sound conceptual constructs.

Bottom-up attempts to develop ontological frameworks are driven by the needs of industries that span different knowledge-specific domains, as exemplified by the M*R/K reference framework for healthcare. A two-faced lesson can be learned from M*R/K: on one hand, there are the benefits of an explicit distinction between representation and knowledge; on the other hand, there is the inherent flaw of applying abstraction to both sides when the concept can only make sense for representations. This flaw is illustrated by the inability of meta-modeling schemes, such as the OMG's Meta-object Facility (MOF), to bridge the gap between applied and foundational ontologies.

Ontological prisms, for their part, maintain the distinction between representation and knowledge, and then unfold knowledge into empirical and conceptual realms. This framework allows for bidirectional, iterative, and incremental knowledge engineering processes. It is bidirectional because developments can progress in a clockwise direction (from facts to concepts) or in a counterclockwise direction (from facts to categories). It is iterative and incremental because individual facts, concepts, and categories are identified separately, allowing developments to be carried out independently. As a corollary, identity principle, instantiation modalities, and abstraction semantics can be defined distinctly for facts, categories, and concepts, avoiding the hiatus between models and meta-models.

2

Ontological Realms

There are more things in heaven and earth, Horatio, than are dreamt of in your philosophy

William Shakespeare

Overview

For philosophers, ontologies are intended to provide a comprehensive and consistent account of the nature of things. For engineers, they must offer the conceptual background of categories that represent specific domains of concern. Ontological prisms, therefore, must be ecumenical with regard to both philosophical and engineering concerns, as well as various domains. To achieve this, an epistemic distinction must be introduced between ontological realms: extensional for environments, intensional for philosophers, and rational for engineers (Figure 2.1).

- Ontologies' extensional realm addresses whatever can be observed about symbolic or physical environments: direct observations, documents, or datasets.
- Ontologies' intensional realm pertains to mental representations.
- Ontologies' logical realm pertains to collective representations.

Overlaps between realms are managed through thesauri, taxonomies, and domains, which address the meaning of words, the partitioning of facts, and the governance of ontologies, respectively. Finally, ontological modalities provide a declarative alternative to imperative meta-models, offering a more robust and less constraining solution for the consistency and reuse of symbolic artifacts.

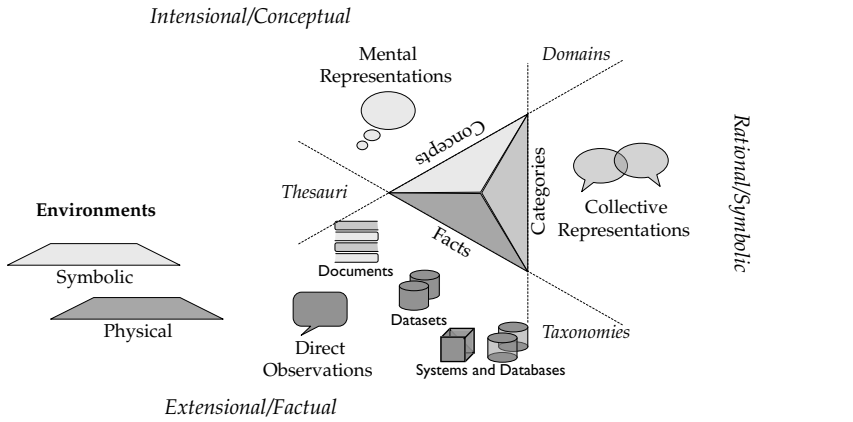


Figure 2.1 Ontological realms.

2.1 Extensional Realm: Facts

2.1.1 From Things to Facts

Ontological debates since Aristotle have revolved around the nature of existence and have continued ever since. Our first step is to place ontological prisms on pragmatic foundations by considering facts instead of things, with facts being observed directly or through documents and datasets. It must be emphasized that observations encompass objects and phenomena in both physical and symbolic environments, provided they can be identified.

2.1.1.1 Collecting Facts

As stated in the preamble, the aim of replacing things with facts is to move epistemic considerations to the background.

That point has taken on a new significance with the rapid and wide-ranging use of language models (see Chapter 5), which can process documents directly as sequences of tokens (signs), bypassing the content's native syntax and semantics. Alternatively, the Extensible Markup Language (XML) can be used for restoring the syntactic structure of document content independently of its meaning. It follows that both language models and XML address documents as reified symbolic contents, i.e., facts.

When considering a restaurant, initial observations would identify elements in both the physical (customers, cooks, tables, ovens, dishes) and symbolic (menus, orders, prices) environments. By contrast to direct observations, recipes and

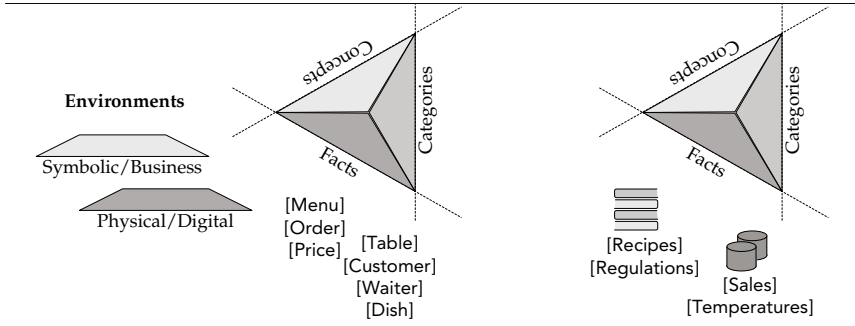


Figure 2.2 From things to facts: direct observations (left) versus documents and data (right).

regulations (documents) and sales and weather temperatures (datasets) are digital collections of detached elements (Figure 2.2).

The benefits of detaching facts and things are not only practical; it is also a divide-and-conquer approach because it corresponds to a distinction between syntax and semantics. Facts can be collected and organized, but they must remain open to semantic interpretation. This is a prerequisite for ecumenical prisms: facts, including datasets, should serve as the entry point and must not be coerced into restricted meanings.

2.1.1.2 The Naming Game

Facts must be identified and, in practice, named; however, their meaning can differ between observers and vary over time. Consequently, if facts are to remain open to different interpretations, they must be managed independently of their meaning. So, what do their names represent?

The answer to this apparent conundrum is provided by Charles Sanders Peirce's distinction between signs, which represent something else in the minds of observers, and symbols, which are signs that represent shared references (i.e., meanings) in the minds of observers. Following that reasoning, the names (or labels) on facts are signs that stand for individual things. Beyond standing in the first intention, signs can also stand in the second intention, i.e., as signs that represent other signs.

These semiotic distinctions can be neatly aligned with the naming of facts as defined by ontological prisms. Signs (or terms) are used to name direct observations without symbolic reference (a). First intention signs are used to name sets of physical individuals (physical tables, b), while second intention signs designate sets of symbolic ones (table seatings, c). Symbols are used to denote concepts (d) and/or categories (e) (Figure 2.3).

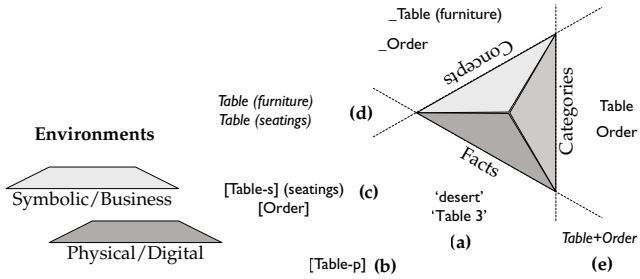


Figure 2.3 Signs and symbols.

It should be noted that these distinctions are discretionary, as their use depends on how facts are shared between ontologies and represented by concepts and/or categories.

2.1.1.3 The Semantics of Connectors

Thinking is often equated with making connections, but what should we make of connections between facts if they are meaningless? That question lingers behind the debates about large language models (LLMs) and generative artificial intelligence (GenAI), particularly concerning syntax and semantics for the former, and correlation and causation for the latter. More generally, that issue is at the core of the recurrent neuro-symbolic AI conundrum, namely, what happens between brains (computations) and minds (cognition). In order to remain ecumenical and keep all options open, ontological prisms maintain a distinction between connector syntax, applied uniformly, and semantics, which are specific to ontological realms (Figure 2.4).

Syntactic connectors pertain to intrinsic descriptions and can thus be applied uniformly to facts, concepts, and categories. In contrast, semantic connectors pertain to functional descriptions, and their interpretation depends on ontological realms:

- With facts: semantic connectors mean composition (sets and subsets) or references (individuals).
- With concepts, semantic connectors mean subsumption or contribution.
- With categories, semantic connectors mean inheritance or association.

From an ontological perspective, the semantics of connectors in fact and concept realms can be defined in terms of parthood (or mereology) relationships between parts and whole and between parts within a whole.

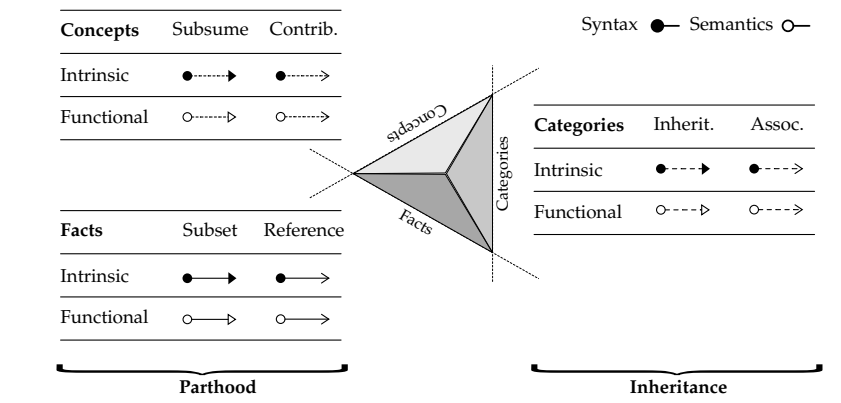


Figure 2.4 Connectors syntax and semantics.

That dichotomy is at the core of the effectiveness of ontological prisms and will thus be recurring. For now, two direct consequences can be mentioned:

- The congruence of the respective connector semantics greatly facilitates the mapping of concepts to facts.
- In contrast, the distinction between parthood and inheritance schemes highlights the specific scope of inheritance and the critical flaw introduced by a misguided uniform application, regardless of the nature of the representations.

Nevertheless, the dichotomy makes room for interoperability of representations:

- Parthood accommodates the implicit inheritance of the identity principle between elements of sets and intrinsic subsets.
- Parthood can be applied to categories provided they are at the same level of abstraction, and thus can serve as a bridge between symbolic representations of facts or concepts on the one hand, and categories, on the other hand.

The overlapping of schemes is typically used in system design to replace subtyping with delegation (see Chapter 8).

Back to our example, consider the §Gourmet restaurant (individual) and its functional references to [Cook] and [Table] (sets), both identified independently (Figure 2.5).

First, the cardinality of connectors between facts is business-specific and consequently falls under ontological radars. What matters are the conditions of their

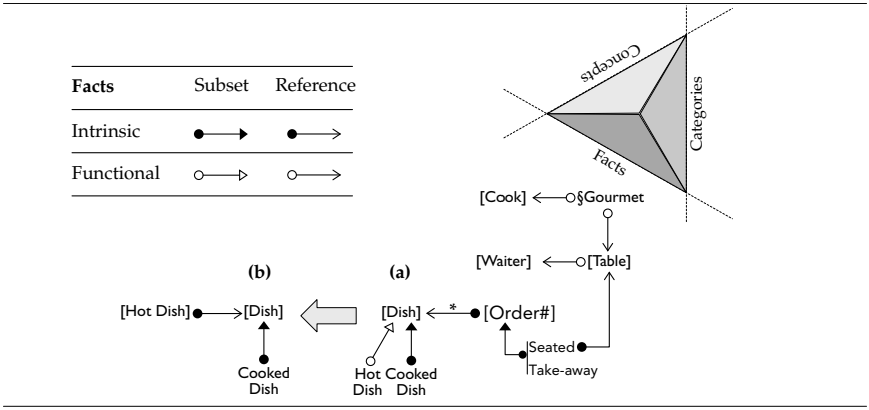


Figure 2.5 A functional subset (a) transformed into an intrinsic association (b).

existence (intrinsic or functional) and their collective arrangement (*), for example, the association between [Order#] (identified as an anchor) and [Dish].

Next, consider the initial hierarchies of [Order#] and [Dish] (Figure 2.5a). The former consists of two intrinsic subsets for seated and take-away orders, while the latter comprises independently defined subsets: one functional for dishes that can be served either hot or cold, and the other intrinsic for cooked dishes. Alternatively, the hot dish functional subset can be replaced by a standalone [Hot Dish] set with an intrinsic reference to [Dish] (Figure 2.5b).

It must be noted that the examples provided are not meant to serve as reference models but to illustrate how ontological prisms can support comprehensive, consistent, and unambiguous descriptions of facts, concepts, and categories.

2.1.2 Facts, Data, Modalities

2.1.2.1 Facts and Data

As far as knowledge engineering is concerned, facts are not simply given; they must be observed, which requires some medium, ranging from human senses to specially designed scientific instruments. For ontological prisms, and considering the semantic constraints mentioned above, this means that facts must first be captured and managed as analog or digital inputs, commonly referred to as data: texts, images, sounds, etc. When data is managed in flat (or 2D) "data lakes," the challenge lies in adding semantic layers to the data, which are typically represented by various types of networks.

Ontological prisms can be used to outline the available options (Figure 2.6):

- Facts can be obtained through applications, organized through schemas, and managed through databases (a).

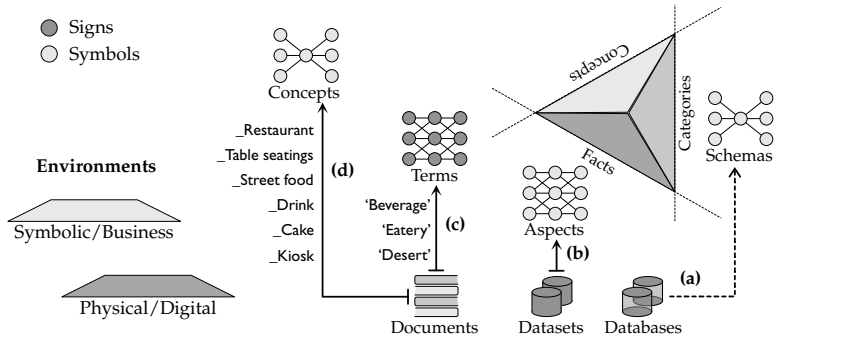


Figure 2.6 Facts and data: (a) information, (b) analytics, (c) nominal, and (d) conceptual.

- Analytical and machine learning technologies can be applied to datasets in order to build networks representing facts, attributes, and dependencies (b).
- Generative methods (see Chapter 5) can be applied to multimodal documents in order to build networks representing content (c);
- Natural language parsing methods can be applied to documents and observations in order to build conceptual networks (d).

Both analytical and conceptual methods produce symbolic networks; the former addresses aspects (or perdurants), while the latter focuses on individuals (or endurants). In contrast, generative methods operate on nominal networks that address terms, which are signs rather than symbols. Facts can thus be exploited in different ways: in relation with prior categories (schemas of managed entities) or concepts, mapped into aspects, or processed nominally (terms). The aim of ontological prisms is to enable their conjunction while ensuring traceability. To that effect, it must be emphasized that, contrary to analytical and conceptual swimlanes, the generative one does not originate from actual things but from reified discourses that are supposed to accurately reflect those things.

It should also be noted that these options address different characteristics of facts and could therefore be combined, provided that the respective outcomes are kept in swimlanes and do not affect the integrity of the sources.

2.1.2.2 Facts and Modalities

Facts can be "known" in various ways, depending on the time and the reliability of supporting data. Observed facts are meant to be obtained independently of prior knowledge (e.g., previous wine or dish orders). Assessed facts are also observed, but their understanding is contingent on what can be known about the reliability of the data (e.g., price increases). Typically, though not necessarily, the reliability of

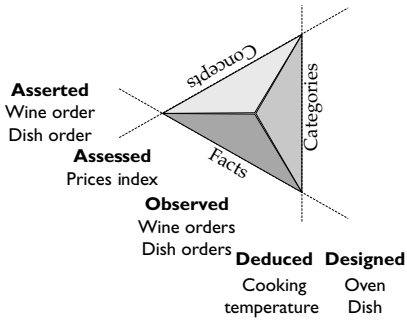


Figure 2.7 Facts modalities.

data tends to improve over time until it reaches its useful shelf life or expiry date (Figure 2.7).

In contrast to observed or assessed facts, some facts may encroach on epistemic boundaries. Asserted facts could as well be defined as embodied concepts, ranging from implicit beliefs (e.g., menu choices) to explicit ones (e.g., religious acts). Deduced facts are derived from observations and causal chains, and could thus fall into specific categories (e.g., cooking temperature). Designed facts pertain to artifacts that are supposedly described by categories such as appliances (e.g., ovens) or dishes; designed facts may be managed as surrogates or remain as unrealized designs.

2.1.3 Facts and Models

The term "data model" epitomizes the conundrum of current 2D ontologies: on one hand, data reflect facts; on the other hand, a model is supposed to be a symbolic representation of those facts. It follows that data models would be a second-degree representation of facts. In practice, the term is used with two accepted meanings: one top-down (enterprise architecture), for data (or logical) models derived from conceptual ones, and one bottom-up (business analysis), for data models extracted (or emerging) from facts. What often occurs, then, is a sleight of hand that creates a make-believe narrative, ignoring intermediate logical steps and directly weaving facts and concepts into a single flat fabric.

The third dimension of prism ontologies eliminates the illusory shortcut of the "flat" 2D modeling paradigm by reinstating the key ontological issue: how to bridge the gap between observed individuals and designed universals. The solution lies in utilizing the dual nature of partitions – conceptual and logical – to span the fault lines between concepts, facts, and categories. More on that later.

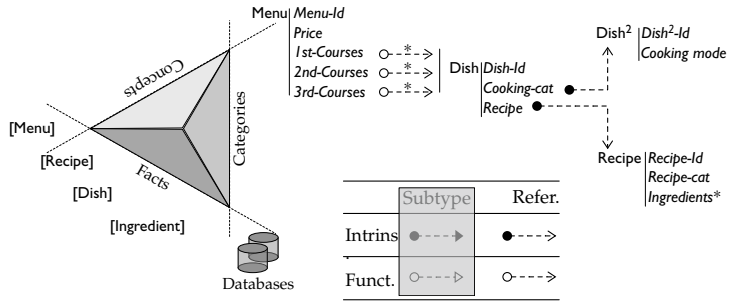


Figure 2.8 Data partitions and relational tables.

2.1.3.1 Data Models

In the traditional 2D modeling paradigm, the term "data model" often refers indifferently to conceptual (business domains and organization), logical (symbolic representation), and physical (technology-related implementation) models. In practice, however, the term is primarily associated with the relational version of logical models, which present a dual, albeit limited, advantage when aligned with ontologies: tables and relations can be represented by straightforward categories, and functional dependencies can be translated as partitions, for instance, dishes can be partitioned with regard to cooking modes (Figure 2.8).

Partitions and logical models are further discussed below with categories, and their use in relation to powertypes and subtypes is detailed in Chapter 8 on engineering.

2.1.3.2 Generative Language Models

While the huge transformational impact of GenAI is being felt across the whole of numerical realms, it also raises a number of issues regarding the boundaries between language and knowledge, and consequently, ontologies. That issue is best illustrated by the fusion of numerical and language GenAI technologies applied to data analytics, undermining the distinction between unstructured and structured data.

Before we take a closer look in Chapter 5, it's important to place GenLMs in a functional perspective. A subset of GenAI technologies, GenLMs are specifically designed to process multimodal documents. To that end, they encode massive amounts of resources into signs (tokens or terms) and then apply deep learning algorithms to morph them into symbolic models that can be trained and then further tuned to domain-specific contexts (Figure 2.9).

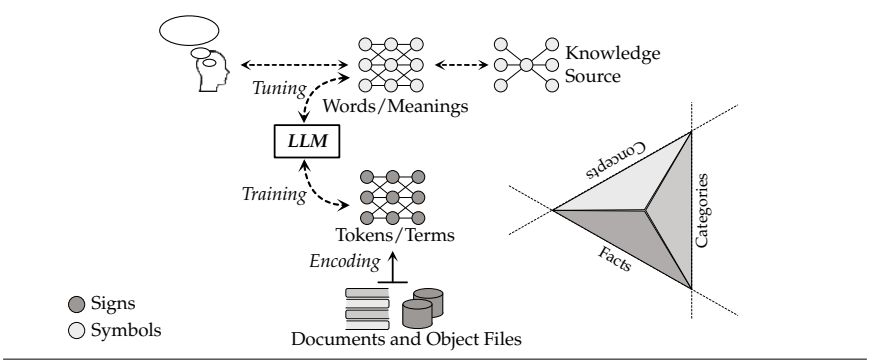


Figure 2.9 Overview of GenLMs.

On that account, GenLMs seem to provide a shortcut from facts to concepts that bypass categories. However, that assessment comes with the assumption that words can be squared with things. As they currently stand, particularly in terms of understanding and processing meaningful content, GenLMs have established a convincing track record, albeit with significant caveats regarding bounded semantics and verified references to facts. Consequently, when operating in unbounded contexts, GenLMs are compelled to outsource most of their value-adding operations to knowledge sources, typically ontologies.

2.2 Intensional Realm: Concepts

The paradigm shift introduced by ontological prisms is their third dimension, namely the distinction between concepts and categories. From a philosophical perspective, the main benefit is the ability to disengage ontologies from the debates between Materialism and Idealism. From an engineering perspective, the distinction between concepts and categories extricates ontologies from the exclusive and exhaustive foundational ur-elements that are meant to account for the entirety of reality.

2.2.1 Relinquishing Aristotelian Ur-elements

Aristotelian approaches, and thus foundational ontologies, assume that all things can be understood as a material footprint of essential categories, the so-called ur-elements. While such a radical assumption may appeal to philosophers, it severely restricts engineers. A third ontological dimension can satisfy both,

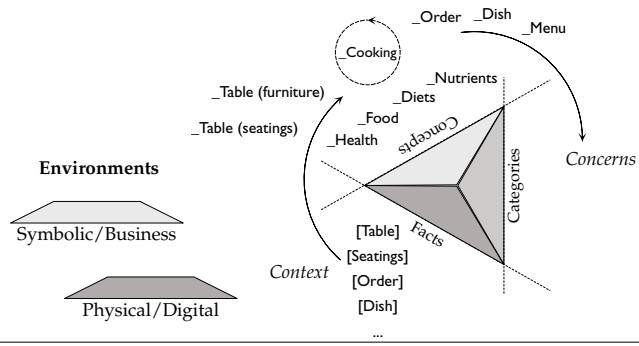


Figure 2.10 Origins of concepts.

providing concepts for the former, categories for the latter, and room for common ground.

Taking food as an example, a concept-driven approach could introduce abstract concepts such as _Food, _Nutrients, _Health, _Diet, and _Cooking. Concomitantly, _Table and _Table seatings could be added to reflect pragmatic observations. Finally, concepts for _Order, _Dish, and _Menu could be included in preparation for subsequent realization (Figure 2.10).

Regardless of their origin, the relevance of concepts must be assessed against targeted contexts (facts) and concerns (categories). To prevent dual relevancies from going astray, such as the religious or health concerns related to diets, some guardrails are needed to replace Aristotle's fundamental elements.

2.2.2 Concepts Modalities

Ontologies should be ecumenical in order to encompass the entire range of scientific and cultural knowledge. As demonstrated by the endless controversies surrounding categories of substance, this cannot be achieved within siloed 2D conceptual models rebranded as ontologies; 3D ontological prisms help to clear these issues.

Starting with nominal things, which Aristotle defines as things without substance, it is easy to see the cultural, if not religious, intractable issues raised by the Aristotelian definition. Yet, when viewed through ontological prisms, "things without substance" can be categorized into two kinds: pure nominals (e.g., 'Apicius'), which are signs viewed as facts and commonly known as tokens, and abstract concepts (e.g., *Health*), which are not meant to be reified as facts. In contrast to abstract concepts, virtual ones are currently realized by hypothetical facts

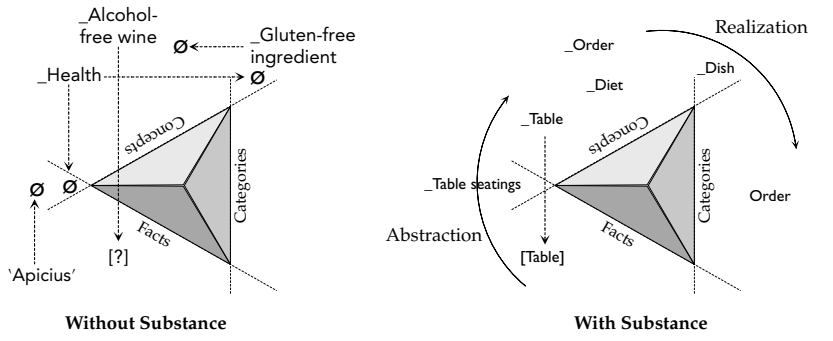


Figure 2.11 Concepts modalities: (left) without substance, (right) with substance.

(e.g., *Alcohol-free wine*), whether asserted or deduced, but could eventually be realized in deeds. Finally, there are ghost concepts, which are not concepts at all but rather categories, typically defined negatively (e.g., Gluten-free ingredient) (Figure 2.11).

In practice, ontological prisms make room and time for epistemic decisions about concepts which can be introduced without substance (Figure 2.11, left) and eventually realized by categories or abstracted from facts, or introduced simultaneously with associated fact and/or categories (Figure 2.11, right). That declarative (as opposed to imperative) approach enables incremental and iterative engineering processes as well, and facilitates the interoperability between prism ontologies and standard conceptual models.

2.2.3 Concepts and Models

As already noted, there is a broad consensus about distinctions between conceptual, logical, and physical modeling tiers:

- Conceptual, for the representation of domains, intents, and objectives, as well as their embodiment in organizations and individuals.
- Logical, for the symbolic representation of conceptual entities independently of the technologies employed.
- Physical, for technology-dependent implementations of symbolic representations.

That taxonomy can also be aligned with the model-driven architecture layers proposed by the object management group: computation-independent models, platform-independent models, and platform-specific models, respectively (Figure 2.12).

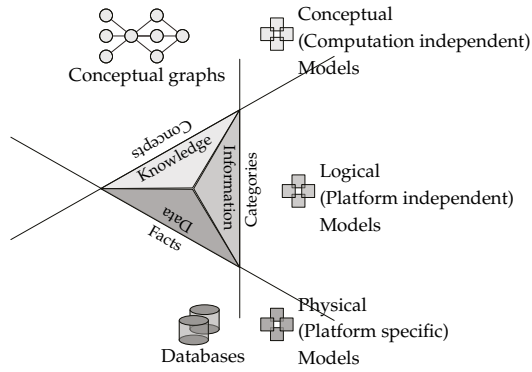


Figure 2.12 A taxonomy of models.

In practice, due to the straightforward transformation of database schemas into graphs, these models are often used independently and sometimes confusedly, especially when the conceptual and logical tiers are merged or when the logical and physical tiers are combined to generate such schemas. These misunderstandings can be avoided by framing these models within ontological prisms, i.e., aligning conceptual and technical models with concepts and facts ontological representation, respectively. More broadly, bringing models and ontologies under the same roof highlights the implicit alignment of corresponding realms: facts, categories, and concepts on one hand, and data (which reflects facts), information (data organized into categories), and knowledge (information serving purposes) on the other hand.

2.2.3.1 Entity–Relationship Models

Beyond naming and graphical variants, most conceptual modeling methods rely on entity–relationship (ER) concepts, with models that can be directly translated into conceptual graphs (CGs). ER models utilize two types of nodes: one for entities and the other for relationships. The former represents individuals identified independently, while the latter represents aspects identified by their association with entities (Figure 2.13).

Assuming that ER models address domain-specific content, their semantic alignment with ontologies should not be an issue, with nodes representing endurants, and relationships perdurants. However, the integration of syntactic constructs, typically inheritance (is-a) and parthood relationships (part-of, *), with semantic ones is likely to increase complexity for large models. More critically, it introduces a conceptual confusion between the semantics of categories such as is-a or part-of and the semantics of instance connectors (e.g., manage, attend).

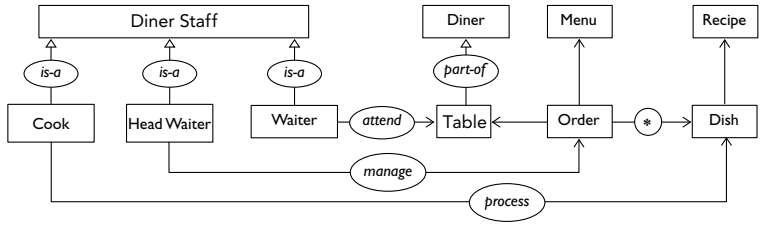


Figure 2.13 Category (closed arrow) and instance (simple arrow) connectors in ER models.

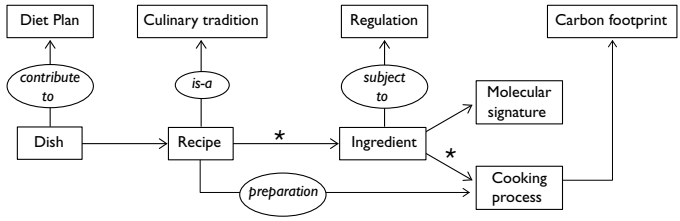


Figure 2.14 CGs as extended ER models fusing the semantics of concepts, categories, and instance connectors.

2.2.3.2 Conceptual Graphs

CGs can be broadly defined as extended ER models with an ecumenical embrace of entities and concepts. In addition to the fusing of ER syntactic (is-a, part-of) and semantic (e.g., subject to, contribute to) constructs, CGs bring under a common roof different layers of abstraction. Such an all-inclusive approach comes with two major drawbacks. First, it blurs the boundaries between heterogeneous realms and consequently muddles semantics, typically between meanings (concepts) and representations (categories). For instance, the is-a relationship between recipe and culinary tradition can be interpreted as a functional reference between category instances, or as an abstraction connector between concepts. Also, if dishes, ingredients, and regulations were managed as categories, but diet plans and culinary traditions were not, there would be no consistent semantics for connectors. Furthermore, it would make no distinction between business (concepts) and engineering (categories) concerns (Figure 2.14).

Second, when woven into the same models, the leveling of perspectives induces exponential growth in complexity, which, in practice, imposes drastic limits on the scalability of CGs. Shortcomings related to semantics and scalability reflect the intrinsic contradiction between heterogeneous content and siloed, 2D containers.

These issues can be addressed using ontological prisms, first by exposing homogeneous conceptual and logical layers, and then by employing dedicated connectors to differentiate between parthood (concepts) and inheritance (categories) on the one hand, syntax (intrinsic) and semantics (functional) on the other hand.

Models can thus combine concepts and categories, with the former representing meanings and the latter representing entities. Regarding concepts, the semantics of connectors are based on the foundational distinction between substantial and moment (or accident) types. Abstraction connectors between levels of representation are defined by subsumption: meanings in higher levels encompass meanings in lower levels. Intrinsic subsumption pertains to substance and is employed for sub-concepts which would be meaningless without the root concept, e.g., the intrinsic variants of _Cooking function, or the intrinsic cooked variant of _Recipe. In contrast, functional subsumption pertains to moments and is employed for sub-concepts that could be meaningful on their own, e.g., assuming temperature may be discretionary, the functional cold variant of _Recipe (Figure 2.15).

Conversely, contribution connectors are established between concepts defined at the same level of representation. Intrinsic connectors denote necessary contributions; for example, _Hot Recipe, which, like Cold Recipe, could be equally defined as a functional subset, entails an intrinsic reference to the meaning of _Recipe when defined in terms of contribution. Functional contributions, on the other hand, are used when semantic contributions are not essential; for instance, the meaning of _Recipe is not intrinsic to _Cooking, as _Hot Recipe is to _Recipe.

As a concluding remark, while the meaning of _Cooking can be functionally extended through a reference to _Cooking function, there is no symmetry because _Cooking function is meaningless without the intrinsic contribution of _Cooking.

It is important to remember that (a) meanings are not universal but determined by domains and (b) that while the intrinsic (substance) versus functional (moment) distinction is applied uniformly to fact, concept, and category connectors, their

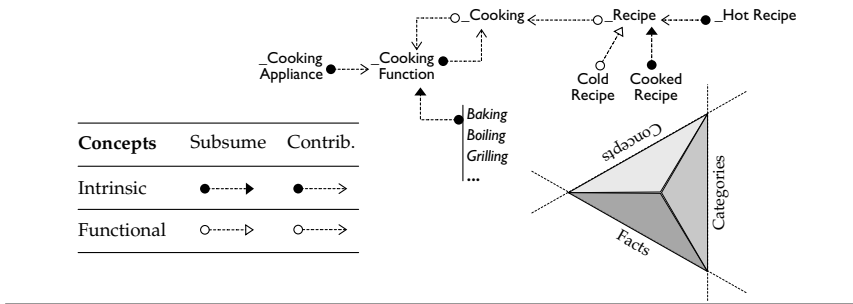


Figure 2.15 Conceptual connectors.

respective semantics differ. Ontological prisms can then ensure that alternative meanings of concepts can be consistently realized through categories.

2.2.3.3 Knowledge Graphs

Graphs are arguably the jack-of-all-trades of symbolic representations, serving as a universal modeling tool that can be configured to represent almost any kind of knowledge. When specifically applied to knowledge itself, rather than, for example, communication networks, their versatility may be misleading, as content (knowledge) becomes confused with the container (graphs), much like conflating family trees with families.

When knowledge graphs (KGs) are used to represent ontologies, they are generally built using the Resource Description Framework (RDF) and the Web Ontology Language (OWL), both of which are standards set by the World Wide Web Consortium (W3C). Originally designed to describe interconnected data on the web, RDF operates at two levels:

- Definition and management of standards for resources such as addresses (Uniform Resource Identifier), documents (XML), or data (JavaScript Object Notation).
- A generic graph modeling language, which has emerged as the primary option for ontology languages.

Taking advantage of their versatility, KGs can be used to represent all kinds of ER models, semantic networks, or CGs, ensuring their seamless integration with ontologies. Yet these benefits come at a cost: when effective and multipurpose tools like RDF are available, methodological concerns are often relegated to the back burner, if not completely ignored, as can be observed in the development of ontologies based on KGs and RDF. Given their powerful and wide-ranging expressiveness, using languages like OWL without principled guardrails inevitably produces indiscriminate representations that jumble epistemic modalities, domain-specific semantics, and syntactic constructs, thereby undermining the very fabric of knowledge. Not to mention the exponential complexity generated by any scaling beyond sandboxes.

With regard to ontological prisms, issues of neutrality and scalability, which are barely manageable for CGs, become crippling for KGs that are meant to concomitantly and consistently represent abstract, common, and specific entities, along with their corresponding individuals. These issues can be overcome by using RDF to implement ontological prisms. In practice, this can be achieved with an OWL kernel built on principled realms:

- The OWL native subclass hierarchy, used to organize things without further precision about their semantics.

- The RDF's Simple Knowledge Organization System (SKOS), reserved for users' domain-specific ontologies.
- Kernel built-in modal categories for facts, concepts, and categories
- Kernel built-in object properties (ontological connectors) pertaining to concepts, categories, individuals, semantics, and logic.
- High-level templates for aspects, facts, concepts, and categories, defined in terms of modalities.

Knowledge engineering with ontological prisms (KEOPS) is described in Chapter 9 and the corresponding kernel architecture in Chapter 10.

2.3 Logical Realm: Categories

Models are symbolic representations, and their value depends on how well they align with the context and concerns. Nevertheless, such alignments can be doubly elusive: For business ontologists working bottom-up from facts, there is a risk of getting lost in flights of abstraction, resulting in models that lack relevance. For foundational ontologists working top-down from concepts, the risk is losing sight of their goals and settling in waste or no-man's land.

Ontological prisms provide a triangulation method to establish guardrails against both of these pitfalls. To that end, categories mark the empirical departure between prism and foundational ontologies: by using facts as a positional base, ontologists can determine baselines by aligning knowledge engineering purposes with specific domains of concern.

2.3.1 Anchors and Connectors

Adding an explicit distinction between concepts and categories requires some extensions to ensure the interoperability and consistency of the triptych.

2.3.1.1 Anchors and References

In contrast to concepts, which pertain to meanings, categories pertain to representations, namely types that describe individuals independently of their nature – whether actual or virtual, logical or digital. As such, categories can be abstracted from facts, a realization of concepts, or a reflection of legacy systems. Categories, or the types that implement them, address both features and behaviors, some of which are intrinsic (tied to identities) and some functional. Like facts and concepts, categories can be identified by anchors (#) that are used to reference elements across realms, whether as abstractions from facts to concepts or categories (e.g., [Ingredient]) or realizations from concepts to categories (e.g., _Diet#) or facts, or from categories to facts (Figure 2.16).

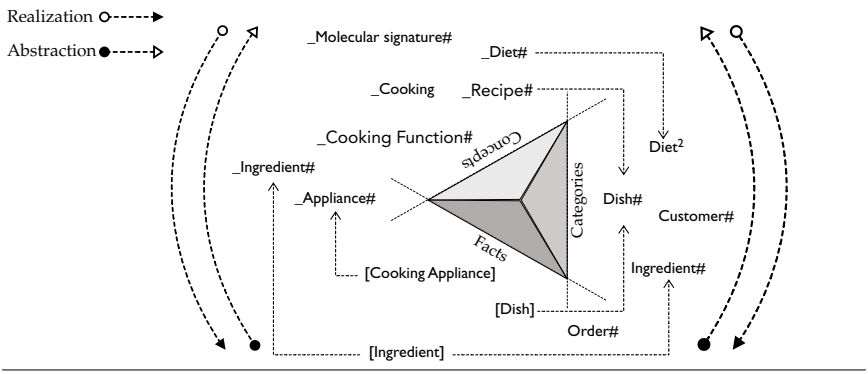


Figure 2.16 Anchoring realizations and abstraction.

Abstraction and realization references set across epistemic realms are inherently heterogeneous and should therefore not carry specific semantics. They can be explicitly represented using the kernel's dedicated connectors (e.g., _Recipe# to Dish#) or implicitly through homonyms when anchors can be directly aligned (e.g., Order#).

Anchors are management tools used by ontologists to identify pivots of abstraction or realization. While they can only be employed with identified elements, they don't have to when elements are not used as anchors between realms. The alignment of anchors can be carried out at two levels: nominal (or shallow) alignments consider the anchors' names independently of associated elements, while comprehensive (or deep) alignments encompass the whole of intrinsic and functional connectors. Comprehensive alignments and interoperability will be discussed in Chapter 3.

Explicit references are used when names differ, whether temporarily (e.g., during iterative development) or permanently (e.g., during incremental development). Implicit references can be used with homonym anchors, providing decisions on representations have been settled. Here are some examples of possible developments (as far as ontologies are concerned, anything goes) (Figure 2.17).

Referenced anchors are the default option when alignments are pending, whether it's for realization (e.g., _Client to Customer#), abstraction (e.g., from [Cook] and [Waiter] to Staff#), or both (e.g., between [Dish#] and _Recipe#).

Nominal alignments allow for a simplified representation of unique anchors set at the intersection of realms, for instance:

Nominal realization: Surrogates for Order# and Customer# are obtained from schemas; the Compliance# category represents the concept; employees that are meant to be classified as Manager#.

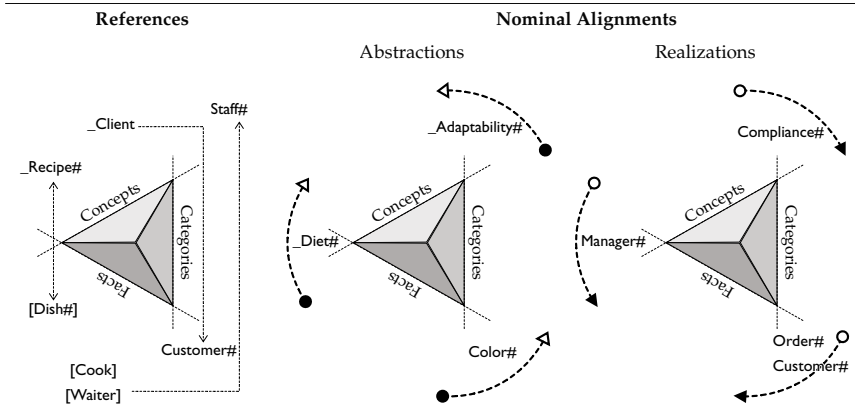


Figure 2.17 References and anchors alignment.

Nominal abstraction: The _Diet# concept is abstracted from data analytics; the Color# category is defined empirically; the concepts of adaptability are defined based on relevant categories and aspects.

2.3.1.2 Logical Connectors

Prisms and the separation-of-concerns principle may seem to be at odds, with one aiming to be a jack-of-all-trades and the other focusing on specific issues. Ontological prisms solve the dilemma by applying syntactic connectors uniformly while using different semantic connectors for facts, concepts, and category realms. As explained above, the semantics for fact and concept connectors are based on the theory of part-hood (or mereology) applied to things and meanings, respectively. In contrast, the semantics of logical connectors is based on object-oriented design (OOD) principles, which address symbolic representations using inheritance and associations.

Similar to the connectors defined for facts and concepts, connectors defined for categories can be intrinsic or functional; that is, they can apply to the essence or to the manifestations of their respective marks. Intrinsic inheritance allows subtypes to inherit identification principles and structures from base types. Functional inheritance allows subtypes to inherit aspects (features and behaviors) independently of the identification principle. For instance (Figure 2.18), Cooked dish represents dishes identified as such (intrinsic inheritance), while Hot dish represents dishes whose status is meant to be discretionary (functional inheritance).

However, if the Hot dish category is defined as a functional subtype, this points to a potential misalignment with the _Hot Dish concept, which is intrinsically connected to the _Dish concept. That apparent discrepancy illustrates the difference between concept and category semantics: as long as representations are

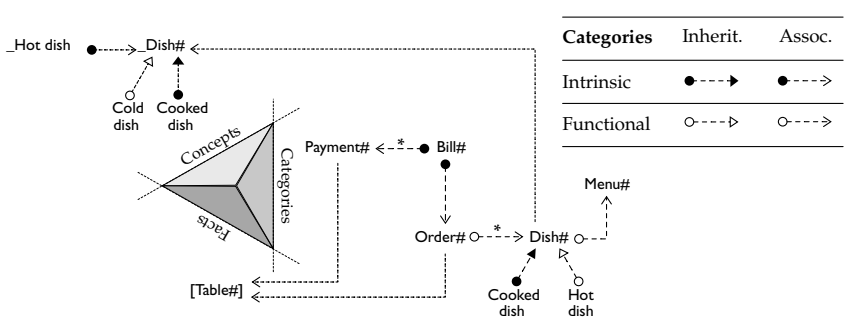


Figure 2.18 Category connectors.

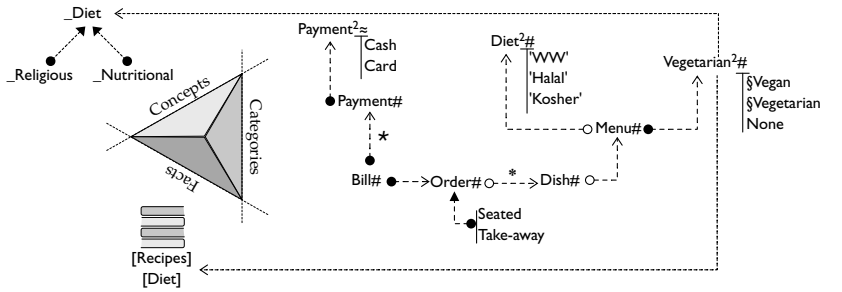


Figure 2.19 Intrinsic (#) and functional (≈) powertypes.

concerned, they are not affected by the way dishes are served; but when meanings are considered, _Hot Dish doesn't make sense without _Dish.

Regarding associations, intrinsic ones are used to address the identification principle and structure, such as those between bill, order, and payment. Functional associations are used in other cases, for example, associations between orders and dishes (assuming orders can be modified). In the example, references to [Table] are managed manually (facts).

2.3.1.3 Powertypes

Inheritance and associations are also used to define powertypes that represent partitions, i.e., subsets of individuals that may eventually evolve into fully entitled subtypes. The properties of powertypes are common to all subsets, but the values are specific to the subsets, i.e., to powertype instances. For example, a powertype representing grape varieties will indicate the regions and colors for each variety (Figure 2.19).

Intrinsic powertypes are used for exclusive partitions (e.g., vegetarian options), while functional powertypes are used for overlapping ones (e.g., dietary options). These options can be defined nominally (e.g., 'WeightWatcher'), by concepts (e.g., _Vegan), or by categories (e.g., Cash or Card). Associations are used to define the partitioning of instances, with intrinsic associations applied when the partitioning is exclusive and functional associations applied otherwise. It should be noted that partitions and partitionings are not necessarily congruent.

At this point, it is worth reminding that the objective is not to replace modeling tools but to ensure a seamless integration of ontological representations with traditional ones.

2.3.2 Aspects

2.3.2.1 Objectives

Aspects, which are the equivalent of moment types in foundational ontologies, straddle ontological realms, first observed as facts before being represented with categories. From an ontological perspective, aspects are entities that cannot exist on their own and only make sense as parts of fully defined entities. When defined through ontological prisms, aspects can be managed as floating representations that remain unsubstantiated until they achieve meaning through their attachment to entities. This allows for a level of indirection with both domain semantics and factual observations.

This offers a dual benefit: the use and reuse of aspects with different semantics (also known as polymorphism) and the sharing of the same semantics between categories. To that end, the distinction between semantic and syntactic constructs introduced for concepts must be reproduced and refined for categories: syntactic constructs, which pertain to representations and bear no meaning regarding what is represented; and semantic constructs, which pertain to meanings and should not be affected by representations. The absence of such a built-in distinction, along with the resulting ambiguities, is a primary cause of the unconvincing track record of meta-models in operational business contexts.

2.3.2.2 Taxonomy and Syntax

The duality of aspects, combining syntax (when floating unanchored) and semantics (when construed by anchors to individuals), mirrors the intrinsic (syntax, neutral) versus functional (semantics, domain-specific) cross-section of ontological prisms. This explicit separation of concerns helps ensure that aspects are used consistently across domains without delving into their semantics. It can also significantly enhance the integration of ontological prisms with data and document management systems, as well as modeling environments.

Ontological prisms organize aspects into three tiers according to their footprint. Local aspects relate to tokens (literals or digital objects), ranged values, or coded

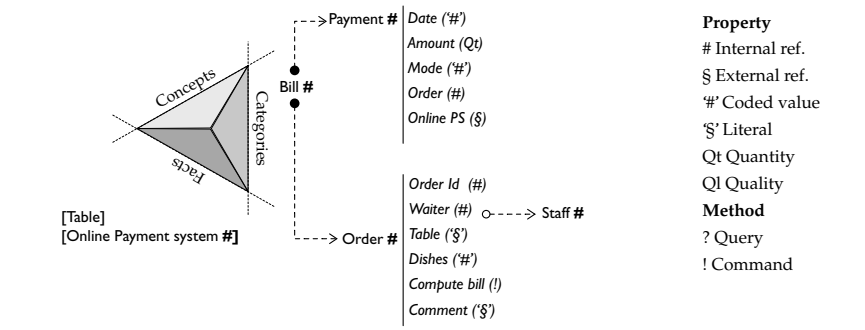


Figure 2.20 Basic aspects.

values. Internal (or endogenous) references point to individuals within the same realm, while external (or exogenous) references point to individuals in different realms (Figure 2.20).

For instance, aspects for dates, payment modes, and dishes are represented by coded values ('#'), while aspects for names and addresses could be represented by literals ('§'), and phone numbers and emails would be represented by external references (§).

Methods are defined by name and modalities (see 2.4.1), ensuring polymorphism, i.e., the possibility of a method's semantics being determined by context.

As far as ontological prisms are concerned, the definition of aspect primitives should be as ecumenical as possible to ensure interoperability while allowing for as much semantic latitude as possible for domain-specific ontologies.

2.3.2.3 Ontological Constraints

Regardless of their source, the objective is to ensure a seamless yet transparent integration of domain-specific rules and constraints with intrinsic and functional ontological ones. To that end, aspect representation follows a parthood model with part-to-whole (P2W) connectors related to the intrinsic or functional presence of aspects, and part-to-part (P2P) connectors related to their content. Intrinsic constraints on aspects indicate mandatory presence or content; presence refers to whether the semantics of the aspect are always relevant for the category being considered, while content refers to whether the value of the aspect must always be determined (Figure 2.21).

Examples of intrinsic presence are the date of payments or their mode, or the table of seated orders. Examples of intrinsic content are payments' reference to order or orders' reference to waiters. Functional constraints on aspects indicate optional presence or content. Examples of functional presence are emails and

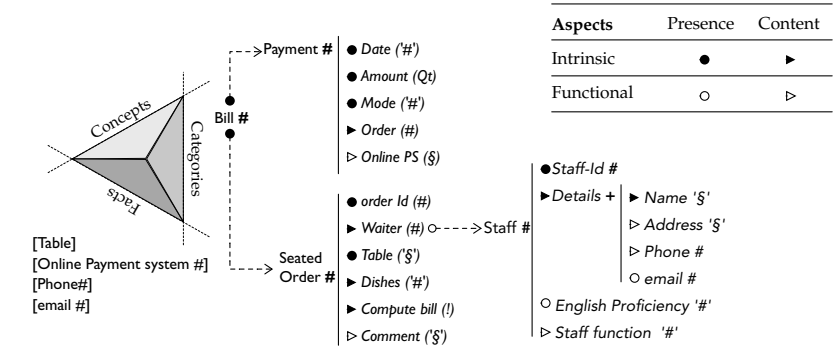


Figure 2.21 Aspects with ontological constraints (see notation on Figure 2.20).

English proficiency. Examples of functional content are staff functions and the online payment system. It must be noted that domain-specific constraints like intervals between the respective time stamps of order and payment, or the range of attributes' values, are best managed under ontological radars. Additionally, established standard operators can be used for collections (*), composition (+), and alternatives (/), as well as abstract data types such as tables, graphs, and so on.

Taking advantage of ecumenical syntax and semidetached semantics, aspects can serve as a glue between entities across domain-specific ontologies. This approach aligns with the way aspects are implemented through domains in relational models, allowing their semantics and constraints to be defined independently of their attachment to entities, thereby ensuring the consistency of shared semantics.

2.3.2.4 Aspects and Relational Schemas

Just like ER models and CGs provide a bridge between standard modeling approaches and ontological concepts, relational models can do the same for logical models and ontological categories. Relational models are by far the most popular enterprise database models, accounting for almost three-quarters of the market. Their importance may be further reinforced as a reliable and cost-effective companion for AI applications, given their formal foundations and extremely efficient implementation, in sharp contrast to the conjectural reliability and high operating costs often associated with AI.

As introduced above, relational database models (or relational schemas) consist of a number of tables, each defined by a set of attributes that draw their semantics from data domains. The aim of relational models is to define keys and dependent attributes and identify the constraints and dependencies on their values. Equating attributes to aspects, "flat" ontological categories can be derived

from tables composed of attributes whose values are set by the same key. This could be achieved either directly from populated databases or by using schemas designed from concepts to organize unstructured data. In both cases, the objective is to sort variants according to intrinsic and functional dependencies, the former set by types, the latter by aspects. Alternatively, schemas can also be translated into bare-bones KGs that could then serve as a basis for ontologies. The object-role modeling (ORM), a reference for fact-based modeling, can be used as a bridge between conceptual schemas and ontological categories (see Chapter 3). In any case, the mapping of relational schemas into KGs can take advantage of normalized schemas, typically using Boyce–Codd Normal Form (BCNF), which ensures their functional homogeneity. Simply put, non-key attributes are fully dependent on primary attributes. Additional normal forms can also be used to address different levels of abstraction. Tables can thus be transformed into nodes, references into edges, and attributes into aspects.

2.4 Ontological Modalities

As noted before, would-be universal approaches to ontologies fall victim to the "rainbow syndrome" of elusive metamodels, characterized by enticing abstractions that can never find solid ground. In terms of software engineering, this issue is addressed by the Liskov Substitution Principle, which states that the sets of identified individuals being represented should not depend on the level of abstraction. When applied to ontologies, this would mean establishing unique and uninterrupted ascendancy. However, there is a caveat: while software engineering starts with symbolic designs set on purpose, knowledge engineering can only rely on discretionary abstract constructs. Representation modalities provide a formal as well as practical alternative.

2.4.1 Meta-models vs Modalities

2.4.1.1 Objective

Modalities are at the core of ontological prisms; from a theoretical perspective, they organize substances and, in practice, replace meta-models. Foundational approaches to ontological substance rely on the assumption that definite substances can be discovered, initiating an endless quest for the Grail without a yardstick by which progress can be assessed. In contrast, instead of confining meanings into square and rigid boxes, ontological modalities shape concepts using vectors set in a finite number of foundational dimensions, such as realm, identity, or agency. Some of these have already been introduced with facts and concepts, while those pertaining to categories focus on abstraction and the reuse of representations.

2.4.1.2 The Underlying Flaw of Meta-models

In principle, abstraction is the alpha and omega of ontology representation; in practice, it is the key to reuse. Without it, ontologies are little more than rebranded conceptual models. Notwithstanding their inconsistent track record in software engineering, meta-models are often presented as a panacea for abstract representations. However, as mentioned in Chapter 1, they fall short when dealing with heterogeneous ontologies and multiple inheritance, which complicate the semantics of domains and the syntax of representations. This is because meta-models suffer from an underlying flaw: they attempt to bridge the gap between the downstream syntax and semantics of domains and the upstream syntax and semantics of abstractions. Consequently, meta-models only succeed when they represent pure abstractions, i.e., categories that are not meant to be instantiated. Otherwise, models fall victim to the confusion between the respective syntactic and semantic constructs of top and bottom representations. In contrast, leveraging the formal distinction between syntax and semantics introduced above (see Figure 2.4), ontological modalities can provide blueprints to untangle this confusion by replacing prescriptive forms with declarative designs.

2.4.2 A Taxonomy of Ontological Modalities

Ontological modalities constitute the rational backbone of prism ontologies and can be seen as a substitute for Aristotle's taxonomy of substances and accidents. Their aim is to ascribe intrinsic and functional characteristics to things meant to be represented, thereby conditioning the alignment of designs (categories) with theory (concepts) and empirical experience (facts). The proposed ontological prism modalities are organized into intrinsic and functional groups (Figure 2.22).

It should be noted that modalities remain discretionary; it is up to ontologists to make the commitments.

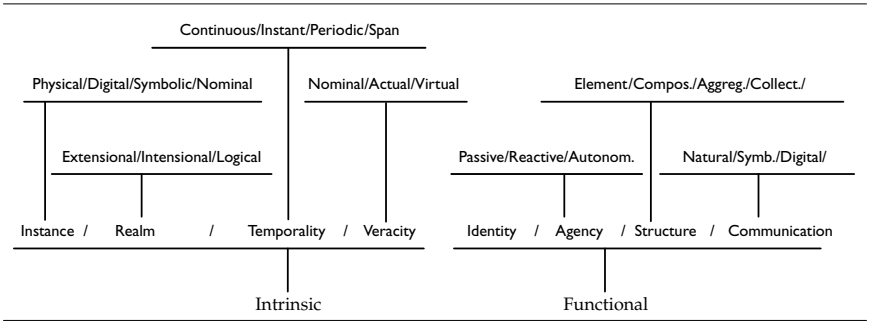


Figure 2.22 Ontological prisms modalities.

2.4.2.1 Intrinsic Modalities

Intrinsic (or creational) modalities characterize the immutable nature and identity of individuals.

2.4.2.1.1 Temporality

Temporality is an intrinsic and exclusive modality that epitomizes the difference between foundational concepts, such as perdurants and endurants, and ontological modalities, the former prescriptive, the latter declarative. Ontological prisms introduce four basic temporal modalities to characterize life cycles: continuous, instantaneous, periodic, and timespan. These modalities can be detailed or extended, provided they remain exclusive and consistent with instantiation and identification modalities.

2.4.2.1.2 Instantiation

Instantiation is an intrinsic and exclusive modality that characterizes the nature of individuals: physical, symbolic, digital, or nominal. Instantiation modality must not be confused with realm (facts can be symbolic) or with identity (which are not exclusive). For instance, an ontologist could consider actual diners (extensional, physical), regular diners (intensional, symbolic), and customer records (logical, digital).

2.4.2.1.3 Realm

Realm is an intrinsic and exclusive modality that characterizes the origin of individuals and determines the footprint of representations: extensional (physical or symbolic), intensional (symbolic), or logical (digital). Realm and instantiation modalities are best determined jointly, with the former used to refine the latter according to domains such as menus versus pricing, written orders (posted on the kitchen wall) versus billed orders (for payment), etc. (Figure 2.23).

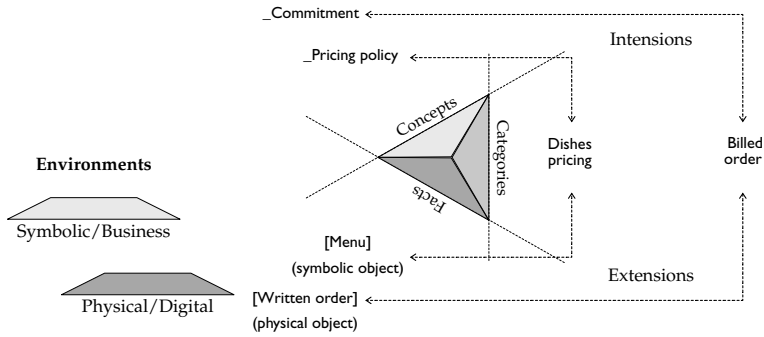


Figure 2.23 Epistemic realm of representations: what is to be known.

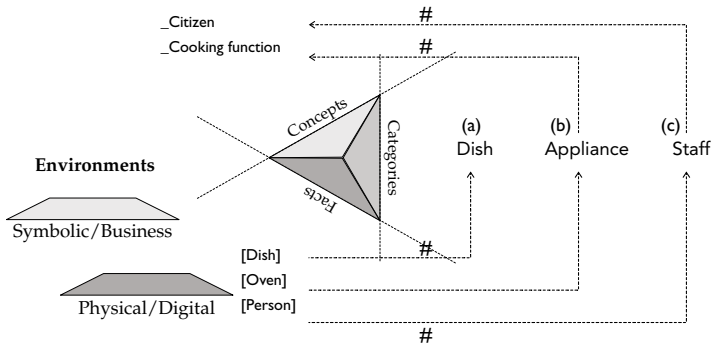


Figure 2.24 Instantiation blueprints: (a) facts, (b) concept, (c) fact and concept.

2.4.2.1.4 Identity

Concomitantly, identification modalities (or principles) define how individuals are meant to be identified: uncountable, anonymous (John Does), temporal (time-stamped), social (personal identities), biological (DNA), mechanical, and so on. For instance, one could identify dishes physically (a), appliances functionally (b), and staff both physically and socially (Figure 2.24).

Identification modalities should not be confused with identification itself: the former pertain to categories, the latter to individuals. In the example above, the identity of ovens will contain a reference to the cooking function of the oven's device category; by contrast, given that intrinsic associations don't necessarily contribute to identification, the identity of dishes may be determined by their instantiation, namely, the cooking process. Moreover, identification modalities are not exclusive; for example, the identity of ingredients can first reference their molecular signature (biological), but they can also make a reference to nutritional functions, typically represented by a code.

2.4.2.1.5 Veracity

Veracity modalities are loosely defined to address ontic modalities of facts (actual, asserted, expected, etc.), concepts (nominal, actual, virtual, etc.), or categories (planned, shelved, operational, etc.).

2.4.2.2 Functional Modalities

2.4.2.2.1 Structures

Structural modalities address the constraints between individuals and aspects: elementary (no structure), collection (identified elements) or container (unidentified elements or uncountable content), aggregate (detached identities) or composite (tied identities). They can be defined on their own or be related to facts

and/or concepts. For instance, dishes could be simply associated with collections (*) of ingredients, while recipes would need more details with regard to ingredient composition (+) and options (/). Beyond the standard constructs for composition, aggregation, alternatives, and collections, structural modalities are best left to implementation, in our case OWL (see Annex).

2.4.2.2.2 Agency and Communication

Agency modalities describe the behaviors of individuals represented by a category independently of domain specifics. This objective has become a priority with the development of distributed architectures of cognizant agents collaborating through natural language. To that end, the basic distinction between passive, reactive, and proactive behavior can be extended for autonomous agents with cognitive capabilities.

Concomitantly, communication modalities characterize the ways agents can conduct actions: symbolic (e.g., billing) or physical, asynchronous (e.g., order) or synchronous (e.g., oven). Autonomy distinguishes between passive (no effect, e.g., Recipe agent), reactive (e.g., Service and Billing agents), and proactive (e.g., Cooking agent) agency. Agency can be further characterized with regard to operations: observation of environments (<), orientation (?), decision (!), and action on environments (>). Structured blueprints can be used to define control modalities.

For instance (Figure 2.25), a waiter passes an order to a service agent (a), which transmits it directly to the cooking agent (b); the cooking agent gets orientation from the Recipe agent (c) before instructing the actual kitchen to proceed and be kept informed (d). Based on its observation, the cooking agent follows with instructions (e) to the service agent to dispatch dishes to the waiter and be kept in the loop in order to trigger further courses as required (f). When the waiter asks

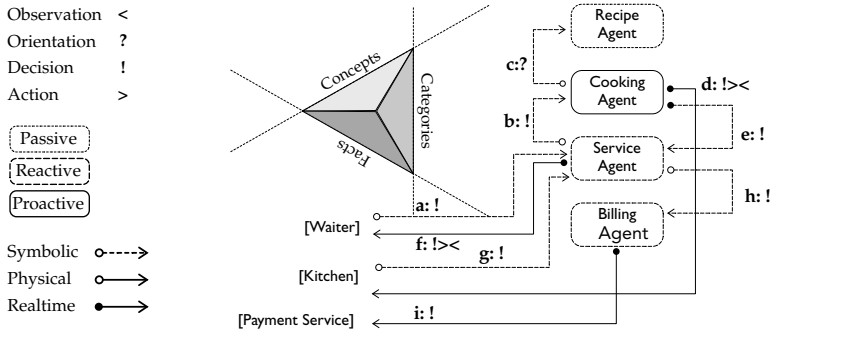


Figure 2.25 Agency.

for the bill (g), the service agent instructs the billing agent to compute the bill (h) and execute the debit operation (i).

Agency modality is taking center stage with the development of digital agents that possess GenAI capabilities. In this context, the ontological distinction between human and digital agents becomes critical, especially concerning their cognitive capabilities of observation, reasoning (Chapter 6), and decision-making (Chapter 7).

2.4.2.3 Modalities and Ontological Commitments

This chapter has addressed the first two requirements proposed by Davis, Shrobe, and Szolovits mentioned in our preamble: ontological commitments and representation of surrogates. In both cases, ontological modalities used as blueprints can replace prescriptive meta-model prisms such as M*R/K (see Chapter 1) with declarative patterns. This enables a two-pronged approach to the representation of surrogates: a top-down (clockwise) approach from concepts and a bottom-up (counterclockwise) approach from facts, which raises the issue of their alignment.

Ontologies are not about truths but about commitments to knowledge architectures built from materials (facts), intents (concepts), and designs (categories). Set with ontological prisms, these commitments are by nature heterogeneous, and their consistency can be managed at two levels: first by ensuring the interoperability of representations, then using blueprints in support of tiered iterative development, one tier pertaining to intrinsic characteristics and the other to functional ones. Chapter 3 will address the interoperability of heterogeneous representations, and knowledge engineering issues will be addressed in Chapters 8 (systems) and 9 (ontologies).

Key Points

This chapter has established the three pillars of ontological prisms as the basis for the proposed knowledge architecture.

First and foremost, there is a foundational shift in paradigm that establishes a three-pronged modeling framework pertaining to facts (extensional representations), concepts (intensional representations), and categories (logical representations).

Second, the introduction of ontological modalities ensures a comprehensive and consistent integration of representations, avoiding the meta-models' double bane of flights to abstraction or falls to no-man's land.

Third, there is a built-in interoperability of resources (data), assets (categories), and intents (concepts) to support an iterative and incremental engineering of knowledge across business domains and time spans.

3

Ontological Prisms and Interoperability

Logic takes care of itself; all we have to do is to look and see how it does it
Ludwig Wittgenstein

Overview

Chapter 2 presented the organization of ontological prisms, along with the principles of their integration with other information systems. This chapter will look inward on the principles and mechanisms that support the alignment and interoperability of heterogeneous ontologies. The focus will be on the gears designed to operate the exchanges across the proposed three-pronged architecture, ensuring the alignment of ontological commitments regarding facts, concepts, and categories across domains and life cycles. First, we will consider the alignment, integration, and interoperability of homogeneous ontologies. Then, the focus will shift to heterogeneous interoperability, the benefits of ontological prisms, and how the interoperability of heterogeneous ontologies can support collective learning and enhance collective intelligence.

3.1 Alignment and Integration

Ontologies may be specific by nature, but their use often requires interoperability across multiple domains, which in turn necessitates matching and alignment methods and routines. Current approaches focus on the alignment and integration of different ontologies based on homogeneous representations, with alignments intended to be carried out a posteriori, and interoperability introduced as an afterthought. In contrast, the aim of ontological prisms is to introduce interoperability

as a built-in capability that can be enacted a priori on ontologies set across facts, concepts, and categories, and consequently heterogeneous representations. In line with the ecumenical purpose of ontological prisms, the objective of this chapter is to present mechanisms that could be applied to the matching, alignment, or integration of homogeneous as well as heterogeneous ontologies.

3.1.1 Homogeneous Alignment

Homogeneous alignment pertains to ontologies addressing domains represented at the same conceptual level and using the same modeling semantics. Taking advantage of that homogeneity, alignment methods and routines can focus on domain forms and semantics. Such methods can be applied as a posteriori to the alignment, matching, or integration of actual ontologies, assuming that mapping yardsticks can be used to assess semantic spans between entities, either nominally (shallow or simple alignment) or comprehensively (in-depth or complex alignment). Given such yardsticks, probes can be oriented or fine-tuned. The former involves changing tack between sources and targets, while the latter adjusts granularity, ranging from single entities to selected sets and complete ontologies.

3.1.1.1 Alignment Principles

The principle of simple (or shallow) matching (or alignment) is straightforward as its aim is to decide on the semantic equivalence of two terms taken in isolation. That can be achieved with traditional tools or generative language models (GenLMs). Complex (or in-depth) matching is more challenging because it takes into account words' context, and consequently the representation of contexts. Without delving into the details of formal definitions and algorithms, the principle is to compare the graphs representing the semantic neighborhood of the targeted entities. A simple example can illustrate the principle: assuming weights of 4, 3, 2, and 1 are assigned, respectively, for intrinsic and functional cross- and same-level connectors to matching targets, the length of the semantic span between "Recipe" and "Dish," which are supposedly in the same semantic space, would be calculated as follows: 4 (cooked, intrinsic abstractions) + 2 (ingredients, intrinsic contributions) = 6. This principle can be applied to named or anonymous constructs (see Figure 3.1).

In practice, thesauri and/or GenLMs do provide a wide range of methods and algorithms to define and tune the weights assigned to conceptual connectors. Assuming homogeneous modeling semantics, ontological prism connectors could also be used to formally define alignment patterns. Intrinsic patterns would be defined by intrinsic connectors and functional patterns by functional connectors (Figure 3.2).

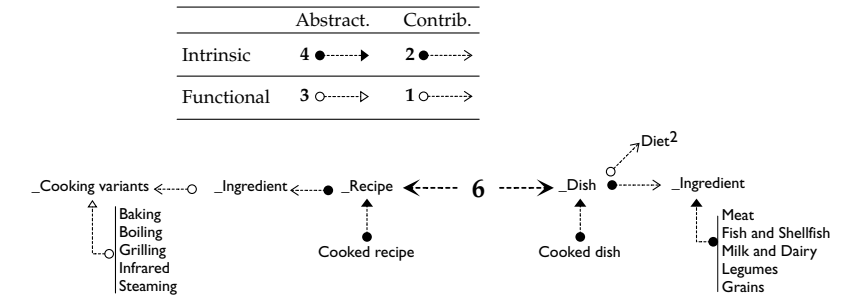


Figure 3.1 A simple yardstick for measuring semantic spans.

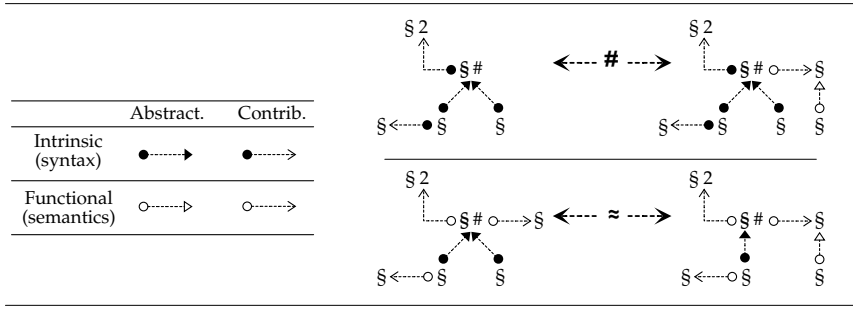


Figure 3.2 Intrinsic (#) and functional (≈) alignments.

Yet there is a caveat: as explained in Chapter 2, while intrinsic connectors pertain to syntax, functional ones pertain to semantics. This distinction makes no difference for homogeneous alignments because they are based on the same representational semantics; however, it does matter if the semantic distances are not defined uniformly, as it's the case when definitions pertain to observed facts, mental representations, and designed categories.

3.1.1.2 Limits of Alignment and Integration

While local and fine-grained matching and alignment can be achieved reliably and offer practical benefits, challenges arise when dealing with entire ontologies. From a formal perspective, even small ontologies can experience exponential complexity due to the merging of semantic graphs, which critically affects the reliability of outcomes. From a practical standpoint, modular and interoperable ontologies are generally much more effective than full integration. This highlights the importance of a selective integration of heterogeneous sources. Driven by language models' contextual retrieval needs, some advances are being made on this

issue, typically regarding the semantic and logical alignment of ontologies based on the Ontology Web Language (OWL) and the Resource Description Framework (RDF). However, progress is hindered by discrepancies of representation semantics and the resulting exponential complexity of alignments, much like geographic maps representing topography, demography, public transport, businesses, etc., on a single layer. Utilizing ontological prisms can remove this obstacle, enabling selective alignments according to purpose (analytical, conceptual, functional, etc.), orientation (abstraction or realization), and/or granularity (aspects, entities, domains, ontologies, etc.).

3.1.2 Integration of Heterogeneous Ontologies

Assuming that simple and complex alignments can be achieved between homogeneous ontologies, the next step is to consider the integration of heterogeneous ontologies. To that end, we first consider the interfaces between realms (facts, concepts, categories), and then the semantic layers involved (Figure 3.3).

3.1.2.1 Ontological Gears

With ontological prisms, exchanges between realms are performed through three mechanisms: thesauri (which match the meanings of facts and concepts), taxonomies (which match the partitions of facts and the specialization of categories), and domains (which establish responsibilities for concepts and categories).

3.1.2.1.1 Thesauri

The role of thesauri is to support the alignment of extensional and intensional commitments, with the former assigning meanings to facts and the latter assigning meanings to values, intents, and goals. Intrinsic alignments achieve this for

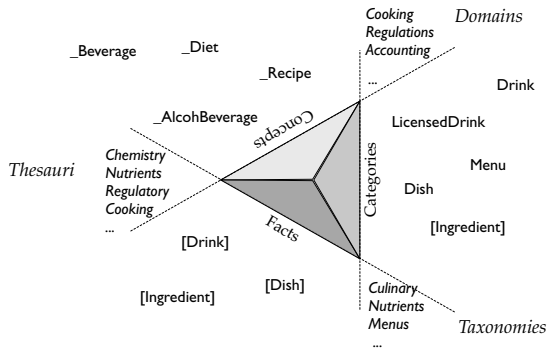


Figure 3.3 Ontological alignments.

anchors and representation levels through exclusive subsumption (for concepts) and subsets (for facts). In our example, the aim would be to align a regulatory concept of _AlcohBeverage with the labels on marketed drinks. Functional alignments serve the same purpose for functional levels, utilizing overlapping subsumption (for concepts) and subsets (for facts). In our example, the general concept of _Diet can be easily aligned functionally with alimentary practices; however, intrinsic alignment between dietary and religious partitions would be more problematic.

In principle, the congruence of parthood semantics (subsumption or subsets) that supports the representation of facts and concepts (see Chapter 2) facilitates the alignment of representation levels, ensuring that meanings at higher levels can be defined independently of those at lower levels.

In practice, document management systems or tools like National Information Exchange Model (NIEM), now an OASIS® Open standard, could serve as a prequel in 2D traditional modeling of thesauri in a 3D prism. The core of NIEM, for instance, is a semantic reference model comprising agreed-upon terms, definitions, relationships, and formats that are independent of their use in information systems. Surrounding that core, NIEM Open includes domain-specific elements shared across the NIEM community, along with associated development rules and methodologies. With regard to ontological thesauri, the main benefit of such tools is their institutionalized management of semantic commons and a built-in distinction between names and name-spaces.

3.1.2.1.2 Taxonomies

With ontological prisms, the role of taxonomies is to ensure the alignment of sets and subsets (facts) with types and subtypes (categories). However, there is a caveat: contrary to the representation of facts and concepts, both of which use parthood semantics, the representation levels of categories rely on inheritance semantics. This distinction prevents a direct alignment of the respective representation levels, except at the bottom between ground level, where instances of facts and categories are identified, and the initial level of their respective representations. In that case, intrinsic alignments entail that individuals in intrinsic subsets are uniquely represented by corresponding instances in intrinsic subtypes; functional alignments do the same for aspects regardless of the individual concerned. For instance, one may assume that a packaged food company would intrinsically align the typing of their products (categories) with the animal or vegetal origin of their ingredients (facts). In contrast, given the unbounded and changing variety of fad diets, the most one could expect from the same company is a label ensuring transparency (functional alignment).

Despite the differences in representation semantics between facts and categories, taxonomies can still provide an alignment mechanism at higher levels of representation by employing a bottom-up partitioning of individual facts alongside

a top-down subtyping of categories. For example, a top-down refinement of aquaculture products (categories), combined with a bottom-up partitioning of seafood (facts), may lead to an alignment of taxonomies such as fish and shellfish.

3.1.2.1.3 Domains

Domains are where concepts are intended to be projected as categories. At this point, one can assume that the scope and boundaries have been agreed upon, and semantic discrepancies have been resolved. As such, domains can be viewed as ontologies, and the objective is to distribute responsibilities regarding commitments (concepts) and designs (categories), as well as the realization of the former by the latter. Considering object-oriented (OO) design as the default standard for category representation, alignment should adhere to the single responsibility principle, which states that changes should originate from a single domain.

The alignment of intrinsic responsibilities can be achieved independently of domain semantics, either through organizational hierarchies – typically, base domains managing concept anchors and sub-domain category anchors – or through delegation from conceptual to category domains. In contrast, the alignment of functional responsibilities should be managed through aspects whose semantics are shared between concepts and categories. For instance, responsibilities for diets and menus can be assigned, respectively, to conceptual and category domains, with intrinsic responsibilities realized through hierarchy and functional ones through shared aspects. However, the caveat mentioned above remains: the fault line between the representation semantics of concepts and categories prevents alignments across representation levels. For instance, there is no reason to assume that the types and subtypes of managed diet categories should mirror the unbounded and expanding conceptual galaxy of ethical, religious, culinary, or nutritional diets.

3.1.2.2 Semantic Layers and Interoperability

Rapid and effective advances in biological and medical ontologies, by nature heterogeneous, have raised the dual issue of the semantic layer and explainability. On one hand, there is a necessary distinction between semantics (user experience) and logic (ontological engine); on the other hand, there is another necessary distinction between language (representation) and knowledge (content). Because these distinctions are not fully congruent, attempts to align heterogeneous ontologies often result in epistemic confusion between "what is known," "how it can be known," and "how to explain it." Some clarification can be achieved by crossing logic and semantic tiers.

On the one hand, logic operates on four different tiers:

- Nominal, when words are dealt with as character strings (signs); e.g., "01," "blue," "Emperor," "Without," "clothes," "random."

- Empiric, when words are dealt with as symbols representing categories defined in bounded semantic contexts with attachments to actual environments; e.g., carbon footprint, wavelength, molecular signature.
- Formal, when words are dealt with as symbols representing abstract categories defined in bounded semantic contexts, regardless of any attachment to actual environments, e.g., circle, isomorphism.
- Deontic, when words are dealt with as symbols representing categories defined in bounded semantic contexts and attached to symbolic environments, e.g., royalty, elections, profit.

On the other hand, logic is expressed through four linguistic forms (see Chapter 6): lexical (signs used to label facts), syntactic (forms used to structure commitments), semantic (symbols used to convey meanings), and pragmatic (meanings tied to contexts).

Logic and grammatical modalities can thus be combined to define ontological prisms' semantic layers (Figure 3.4):

- Thesauri must support nominal, empirical, and deontic reasoning.
- Taxonomies must support formal and empirical reasoning.
- Domains must support formal and deontic reasoning.

The issue of semantic layers is typically addressed in relation to the consistency of domain-specific thesauri and/or in terms of the consistency between thesauri and taxonomies. Ontological prisms take the thesauri/taxonomies approach further by explicitly distinguishing between the logical foundations of taxonomies and the semantic foundations of thesauri, while simultaneously spanning the interval between the semantic spaces defined by taxonomies and the semantic vectors used by language models to represent meanings.

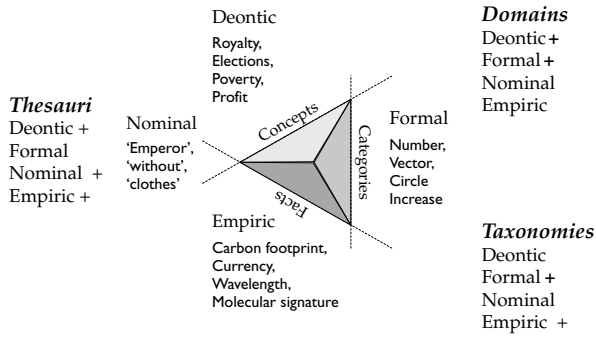


Figure 3.4 Ontological prisms semantic layers.

That point, and more broadly the roles of direct and mediated communication, can be illustrated by a thoughtful homonym mentioned by Wegerif and Casebourne in their insightful article about the role of direct (dialog) and mediated (writing) communication in education (see Wegerif R., Casebourne I. (2025)). In ancient Greek, "pharmakon" means both poison and remedy. Thesauri address the issue by referencing the different uses of words according to actual contexts, while taxonomies tackle the issue by organizing labeled facts and memorizing the outcomes.

These interfaces and semantic layers facilitate the alignment and integration of heterogeneous ontologies, which is a necessary condition for reasoning (Chapter 6), decision-making (Chapter 7), and engineering (Chapters 8 and 9).

3.2 Interoperability of Heterogeneous Ontologies

Interoperability is the dynamic counterpart of alignment and integration. Through ontological prisms, interoperability can accommodate the heterogeneity of ontological content as well as their varying levels of maturity. To that end, ontological prisms work as an engine with three types of gears associated with thesauri, taxonomies, and domains, which address meanings, individuals, and responsibilities, respectively.

These gears can operate at two levels (intrinsic and functional) and in two directions (abstraction and realization). When combined, these gears can support collaborative knowledge engineering processes (Figure 3.5).

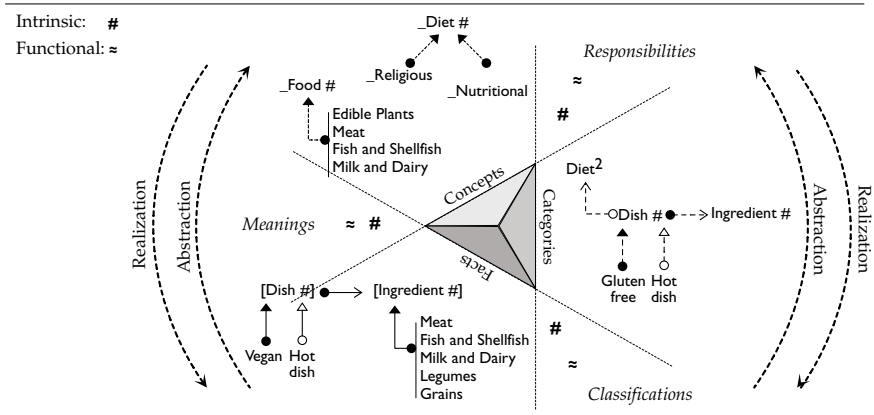


Figure 3.5 Ontological gears and interoperability of contents.

3.2.1 Interoperability and Reuse

From an ontological perspective, alignment pertains to knowledge management, while interoperability pertains to knowledge engineering. Drawing from engineering principles, our primary objective is to identify the conditions that must be met for the reuse of concepts, categories, and facts.

In contrast to physical or software engineering processes, knowledge engineering operates iteratively in both directions: top-down, where abstractions (concepts or categories) can be realized differently across domains, and bottom-up, where realizations (facts) can be abstracted differently across domains, whether as concepts or categories. That multipronged use is exemplified by data mining (from facts to concepts), process mining (between facts and categories), counterfactuals (from concepts to facts), or archeology. However, there is a critical difference: in actual engineering, changes to artifacts are intended to be phased and managed, allowing for organized reuse. In contrast, knowledge changes are meant to be piecemeal and continuous, which undermines the very principle of reuse. Ontological modalities offer a way out of this conundrum by establishing intrinsic and functional guardrails supporting reuse.

3.2.2 Facts, Concepts, Meanings

Meanings are like mythological chimeras, as their contours and colors encapsulate shifting semantic commitments to both concepts and facts. The challenge for ontologists building thesauri of such heterogeneous assortments is to weave them into a consistent fabric to ensure the continuous interoperability of the ontologies considered, whether in abstraction or realization. Taking our restaurant as an example, customer orders can be used to abstract customer profiles, while customer relationship policies are meant to be realized through menus.

3.2.2.1 Abstraction: Entity Resolution

When considering facts, ontologists begin by labeling groups of identified things or phenomena and then proceed with more abstract descriptions based on what they can learn from observations. At some point, new learnings must be compared and possibly matched to prior perceptions; however, the pairing, broadly known as entity resolution, is not symmetric (or homogeneous) as it involves an epistemic discrepancy between domain-specific observations and shared symbolic assets.

The aim of entity resolution is to identify intrinsic threads in unstructured datasets and use them to construct the meaning of concepts by weaving these threads together with functional associations. For instance, _AlcoholicBeverage appears

as a straightforward concept abstracted from observations and characterized by intrinsic (alcoholic) and functional (beverage) dimensions. That's not the case for the _Diet concept, which lacks a clear intrinsic identity.

Considering integration and assuming _Diet and _AlcoholicBeverage are known concepts, [Vegan] and [Absinthe] are terms introduced for new facts. Abstraction can be used to align the meanings of these new terms with the meanings of established concepts: [Vegan] could be interpreted as a "kind of" _Diet (functional alignment) until more is known about respective intrinsic modalities, typically identification principle and instantiation. In contrast, assuming [Absinthe] is known to denote an "is-a" (intrinsic alignment) alcoholic drink duly identified as such, the assumption can be used to integrate the new meanings with the semantics of wine and alcoholic beverages.

Considering interoperability and the planning of our restaurant's menus, the objective would be to introduce new nutritional diets in line with customers' seasonal leanings. In this case, a new concept representing the influence of weather on customers' choices will be added, and its intrinsic and functional modalities will be determined from observed changes (facts) in customers' orders in relation to weather. To achieve this, analysts could utilize internal resources such as sales datasets and recipes, along with one external weather dataset. The new concept, _WeatherCulinaryFactor, characterized, e.g., by factors such as temperature and sunshine, would serve as a bridge between facts and the semantic commitments relating to _Recipe, _CulinaryTradition, and _CookingFunction. Statistical inference could then be employed to determine the concept's identification principle (#), and to weight functional connectors (≈) with _Recipe, _CulinaryTradition, and _CookingFunction (Figure 3.6, left).

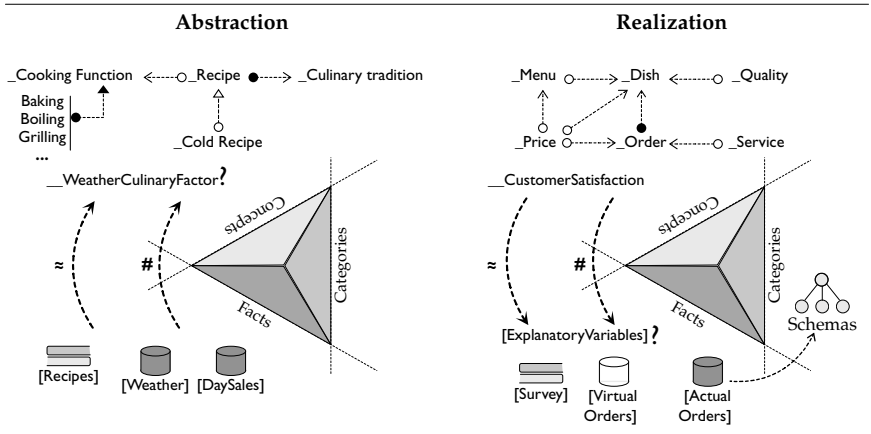


Figure 3.6 Interoperability of facts and concepts: (left) abstraction, (right) realization.

3.2.2.2 Realization: The Matter of Names

Shifting the orientation from abstraction (from facts to concepts) to realization (from concepts to facts), the objective now is to rearrange the labeling of facts in line with the newly introduced concepts. Intrinsic realization applies when the intrinsic modalities of known concepts, typically realm and instantiation, can be applied to new facts; otherwise, the same routine can be attempted using functional modalities, such as identification. For instance, while the [Diet] label can be attached to a number of observations, the _Diet concept remains without intrinsic modalities.

Considering integration, the _Diet (culinary) concept could be realized through religious dietary rules documented as facts. However, if the _Diet concept has already been functionally realized by nutritional diets, the integration of religious dietary rules would require a review, if not an update, of the _Diet concept, which in turn could affect the prior meaning of nutritional diets. It must also be noted (Figure 3.3) that while [Drink] can be intrinsically aligned with _Beverage, the alignment of Dish (category) with _Recipe (concept) can only be functional due to the differences in identity and instantiation modalities.

Considering interoperability and the restaurant's customer relationship policy, analysts would seek facts that could materialize (realize) customer satisfaction. Compared to the descriptive methods adopted for abstractions, realizations rely on prescriptive methods to build simulacra of factual and counterfactual realities. In our example, let's assume analysts would opt for a two-pronged approach: a quantitative analysis of orders to identify culinary patterns and a qualitative review to identify explanatory variables for customer satisfaction (Figure 3.6, right). Using the concepts of _Price, _Quality, and _Service to define explanatory variables, analysts would combine reviews and predictive methods to create virtual orders that could be used to assess an (unobserved) variable representing customer satisfaction.

The example highlights the importance of thesauri for the integration of heterogeneous ontologies. First, by introducing a level of indirection between the meanings of signs (observations) and symbols (representations), prism thesauri enable iterative and incremental knowledge engineering processes. Then, iterations can build momentum by shifting orientation back (abstraction) and forth (realization), leveraging the asymmetry between the two moves: abstraction advancing with the meaning of representations (symbols) and realization advancing with the meaning of words (signs).

3.2.3 Facts, Categories, Digital Twins

As emphasized in Chapter 2, taxonomies play a pivotal role in foundational ontologies, from Aristotle's intrinsic ur-partitions that bridge the gap between concepts and categories to functional classifiers that do the same between facts

and categories. With their intrinsic hand, partitions address substance and consequently aim to define immutable, exclusive, and all-inclusive entities. With their functional hand, partitions deal with observations, which are inherently limited to apparent features and functions. Mirroring the role of thesauri for meanings, taxonomies are intended to serve a similar purpose for individuals. These individuals can be identified as observed individuals on one hand (e.g., actual customers or orders) and as instances of categories (or surrogates) on the other hand (in this case, customer or order records).

3.2.3.1 Interoperability with Engineering Solutions

We begin with two examples of interoperability featuring representative current engineering solutions.

3.2.3.1.1 From Categories to Facts: Unified Modeling Language

Interestingly, a language like Unified Modeling Language (UML), whose OntoUML extension often serves as a basis for foundational ontologies, was initially explicit about the distinction between individuals and their representations, only to dismiss the concept itself in its later developments. That misguided afterthought has been of little consequence as long as models were meant to represent artifacts, and the flaw could even be tolerated for homogeneous ontologies, because in both cases, representation semantics are intended to be uniform. However, the confusion of representation semantics for facts and categories becomes critical when dealing with heterogeneous ontologies that combine different kinds of representation and maturity levels. As before with meanings, we consider interoperability in terms of the abstraction from facts to categories and the realization of categories as facts.

3.2.3.1.2 From Facts to Categories: Object-role Modeling

The object-role modeling (ORM) method, developed by Terry Halpin, provides a formal basis for mapping facts to categories, supported by its explicit reference to set theory for the former and to the relational model for the latter. Without delving into ORM, it is sufficient to say that it is both comprehensive and formal, addressing all modeling issues pertaining to the mapping of facts into categories (Figure 3.7).

Given that ORM can be translated into the Web Ontology Language (OWL), it can be used to organize properties along with ontological prisms. However, ORM can only be effective in well-circumscribed semantic contexts, not to mention heterogeneous ones.

3.2.3.2 Abstraction and Artifacts

3.2.3.2.1 The Matter of Time

As William Faulkner famously wrote, “The past is not dead. In fact, it’s not even past.” Even without considering virtual facts, observations at any moment

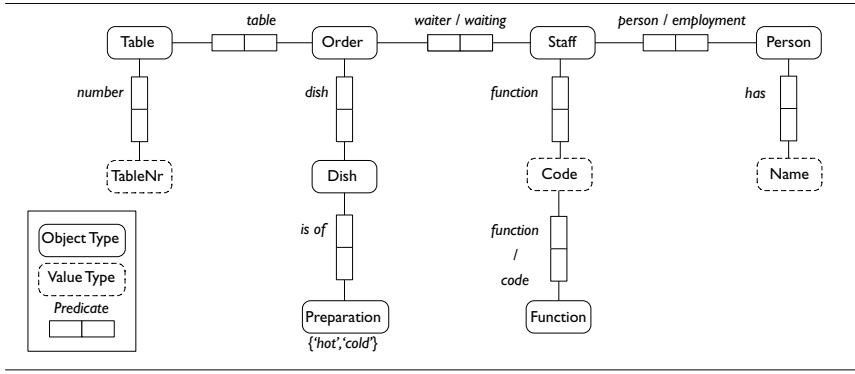


Figure 3.7 Overview of ORM constructs.

blend imprints of past phenomena with ongoing ones, and natural entities with artifacts. It's not only facts that mix time scales; our knowledge about these facts and, consequently, their representation is also changing, creating an infinite number of timelines and capsules. When set between facts and concepts, the issue may be left to scientists, historians, or philosophers. However, it has very practical consequences when considered between artifacts and the categories used for their design, particularly for legacy systems. In that case, achieving interoperability may require something akin to a time machine capable of reversing engineering processes. We will return to time-related issues in the following chapters.

3.2.3.2.2 Schemas and Reverse Engineering

Consider two types of artifacts: applications and databases. Applications are by far the more challenging of the two, as they consist of a large number of executable files (the individuals) that have, over time, been subject to haphazard modifications, obscuring their functional contributions. This has led to a disconnection between the intrinsic individuals and their actual functional contributions, which can only be observed through the execution of the applications.

By contrast, databases epitomize the use of ontological prisms for reverse engineering. On the factual side, one finds "real" individuals ([Order#], [Customer#]) and their surrogate counterparts (§) in databases, with their intrinsic and functional components neatly aligned; while on the category side, one finds the models, known as schemas, of the surrogates. When functional changes are required, schemas can be updated and surrogates replaced without affecting their alignment with actual individuals (Figure 3.8, left).

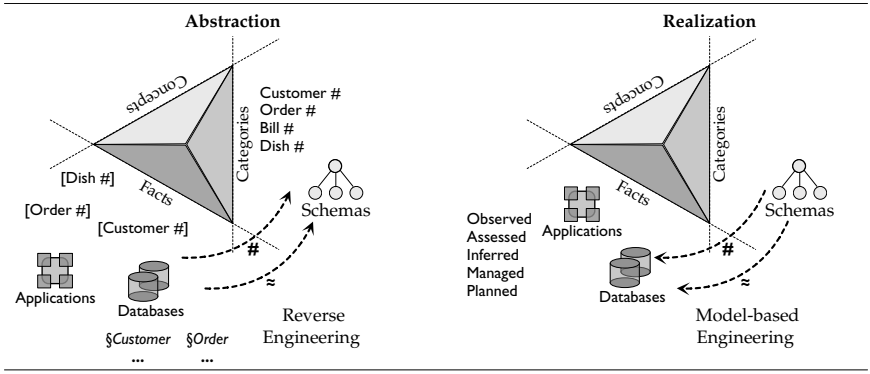


Figure 3.8 Interoperability of facts and categories: (left) abstraction, (right) realization.

3.2.3.3 Realization of Artifacts

The ontological distinction, or lack thereof, between "real" individuals ([Order#], [Customer#]) and their surrogates (§Order, §Customer) points to a broader issue: the lack of ontological status for artifacts. This ontological blind spot leads to both foundational and practical consequences. On a foundational level, it generates a kind of regressive loop: being designed, artifacts are by nature embodiments of knowledge; ontologists trying to understand the substance of things should be aware that when they consider artifacts, which they more than often do, they are looking in a mirror. On a practical level, taking for granted that most of our environments are made of artifacts, efficiency would suggest designing ontologies accordingly. Model-based systems engineering (MBSE, see Chapter 8) illustrates that point (Figure 3.8, right).

While reverse engineering focuses on individuals, MBSE concentrates on the corresponding categories (schemas). In this context, interoperability is not about the intrinsic or functional alignment of facts (the executable files) and categories (the models) – a built-in feature of MBSE – but rather about the alignment of the timelines. At the enterprise level, the objective is to align engineering and business time frames, the former focused on intrinsic identities (#), the latter on functional contributions (≈). Since there is no reason to assume any kind of native congruence between the respective granularity and timelines of business expectations and engineering capabilities, ontological modalities (e.g., observed, assessed, planned) can greatly enhance the dual alignment of time frames (business vs engineering) and deliverables (intrinsic vs functional).

3.2.3.4 Viable Systems and Digital Twins

For enterprises competing in digital environments, change is the rule of the game. Their architecture must undergo both emerging and planned evolutions. Emerging evolutions are driven by unforeseen and/or forced changes, which may conflict with the current state of their organization or systems. Planned evolutions involve designed changes to organizations or systems, which may be misguided or out of sync with changes in the environment. Balancing these momenta can be achieved through versatility and plasticity: the former for the ability to change functions without changing form, and the latter for the ability to change form while maintaining functions. In ontological terms, this means balancing intrinsic and functional changes; in engineering terms, it means managing legacy systems while also pursuing new developments. Such challenging objectives embody Stafford Beer's vision of viable systems at the core of his cybernetic approach to enterprise management, which he developed fifty years ago. Today's digital twins can be seen as a modern avatar of Beer's viable system, now equipped with a full range of digital and artificial intelligence (AI) technologies. Broadly speaking, digital twins are designed to support:

- A comprehensive and consistent symbolic representation of the system in question.
- Real-time synchronization between internal and external time frames.
- Full transparency regarding resources, traceability regarding operations, and explainability regarding the rationales behind actual and virtual activities.

When supported by ontological prisms (Figure 3.9), digital twins would be represented as proactive surrogates with both physical and digital modalities

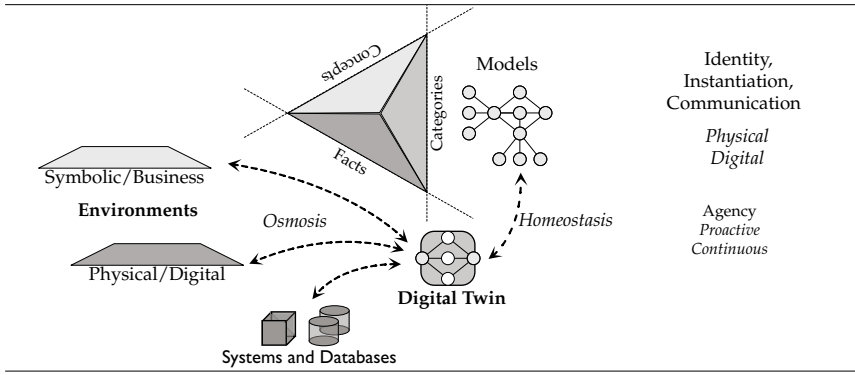


Figure 3.9 Digital twins.

with regard to identity, instantiation, and communication. That would ensure full transparency and interoperability across the realms of environment (facts), business (concepts), and systems (symbolic representations).

The comparison with viable systems highlights two aspects of the interoperability of ontological prisms: osmosis (between surrogates and environments) and homeostasis (between environments and representations):

- Osmosis: Machine learning technologies, such as data and process mining, will enable a seamless exchange of information between systems and their environment.
- Homeostasis: Knowledge graphs (KGs) and ontologies will enable the continuous alignment of business models, governance, and operations with enterprise architectures.

We will return to these points in Chapter 4.

3.2.4 Concepts, Categories, Organization

The primary consequence of the digital revolution has been the dismembering of established boundaries between enterprises and their environments, accompanied by the weaving of organizational and system threads. Domains have thus emerged as a key organizational unit responsible for aligning business objectives with system capabilities, in ontological terms, concepts and categories. While the new responsibilities of domains have prompted a renewed interest in conceptual models (often rebranded as ontologies), this has also led to some confusion between meanings (concepts) and representations (categories). Such oversight has limited consequences as long as ontologies are homogeneous and semantics are congruent; however, this is not the case for heterogeneous ontologies. In fact, the primary purpose of heterogeneous ontologies is to bring a plurality of contexts and concerns under a common roof to ensure the transparency of intentions and the explainability of outcomes. To achieve this, ontological prisms leverage the distinction between concepts and categories, with conceptual remits centered on values, intentions, and objectives, and category remits focused on the logical and functional representations supporting the realization of these values, intentions, and objectives. Once these remits are clearly established, domains can be assigned responsibilities.

Similar to thesauri for meanings and taxonomies for individuals, domains consider the alignment of responsibilities defined in terms of orientation and level. The orientation determines which concepts or categories are at the origin of the moves, realization, or abstraction, respectively. The level determines the scope of the responsibilities: substance (intrinsic alignment) or aspects (functional alignment).

3.2.4.1 Abstraction: Alignment of Emerging Changes

One way or another, organizations are shaped by the nature of activities and the role of human contributions. The question is: what happens to organizations when activities are immersed in digital environments, and technologies blur the origin of contributions? To begin with, the emerging changes brought about by intelligent systems through digital channels may carry more weight but offer less transparency than changes that are planned in advance. Hence, the benefits of an explicit representation of their crossing paths. Emerging changes can occur locally at the system level through process mining and tuning, or induce broader changes, and consequently alignments, at the organization level. Typically, such bottom-up organizational alignments occur when new concepts are needed to make sense of changes in managed categories. For instance, a merger between a restaurant and a packaged food company may require a redesign of supply chains as well as a redefinition of product lines. In this case, some realignment will be necessary between the respective responsibilities of the Dietary domain (concepts) and the new (merged) Cooking domain (categories). In line with object-oriented design principles, the ontological distinction between intrinsic and functional responsibilities is intended to ensure that changes at lower levels of representation do not affect meanings at higher levels. For example, the intrinsic meaning of the _Diet# concept should not be impacted by changes in the Cooking domains (Figure 3.10, left).

This example illustrates the centrality of ontological commitments: while the meaning of some nutritional diets may be derived from categories, that's not possible for religious dietary rules.

3.2.4.2 Realization: Alignment of Designed Changes

Designed changes of categories may be homogeneous, e.g., creating new dishes for specific diets, or heterogeneous, such as reorganizing our packaged

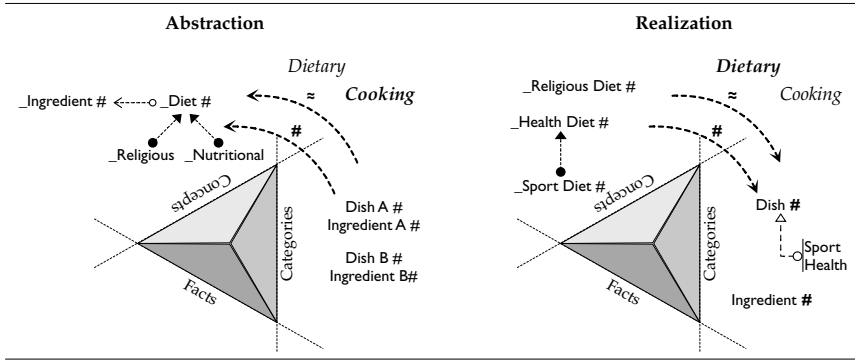


Figure 3.10 Emerging (left) and planned (right) changes.

food business into separate product lines, one addressing religious concerns, the health and sport ones. In this case, changes will affect all domains and responsibility, concepts, and categories, intrinsic and functional (Figure 3.10, right).

As before, the alignment of syntactic and semantic connectors with intrinsic and functional ones, respectively, means that responsibilities for intrinsic changes, e.g., Dish# and Ingredient#, can be aligned across domains independently of their semantics. This is not possible for functional changes, e.g., *vegan, halal, kosher, light weight, feel good, high performance*, which pertain to aspects shared within homogeneous representations or reused across heterogeneous ones. In both cases, the alignments can be achieved in two steps. The first step takes advantage of parthood syntax to compare the structure of aspects. The second step relies on polymorphism to allow the meaning of aspects to be determined differently depending on domain semantics, for instance, a campaign marketing a diet good for body and soul.

3.2.4.3 Architectures: Alignment of Time Frames

The decoupling of ontological realms and their representation throughout their life cycles allows for architectural changes to be managed across different time frames. Regarding enterprises (Figure 3.11, left), intrinsic alignments will address the mapping of the conceptual and logical representation of enterprise assets (human, financial, technical, data, information, knowledge), while functional alignments will do the same for business capabilities (communication, reasoning, control, operations, decision-making). Concerning time frames (Figure 3.11, right), prism modalities (actual, expected, planned, etc.) will be used to define

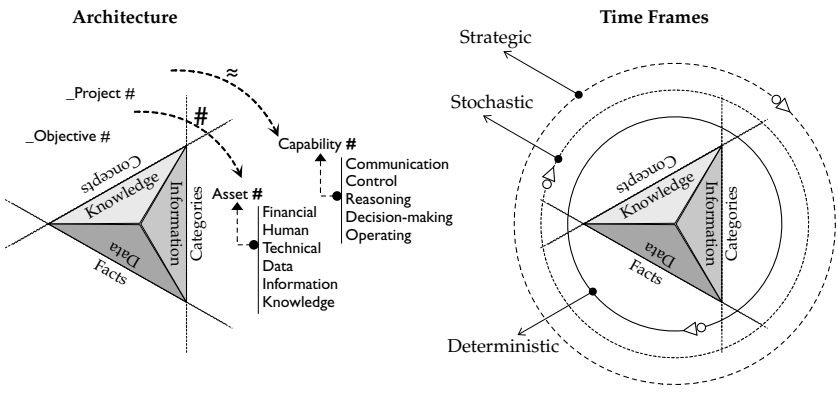


Figure 3.11 Architecture planning (left) and time frames (right).

operational, tactical, and strategic planning, depending on what can be known about present and future business environments:

- Deterministic time frames: when business events can be immediately and unambiguously attributed to relevant information and utilized as knowledge; corresponding alignments can be formalized as procedures.
- Stochastic time frames: when business events involve random variables, the potential range of which can be accounted for by available information; corresponding alignments can persist as long as the reliability of the data can be improved or until the "last responsible moment," when further delay would be detrimental.
- Strategic time frames: when business events include boundless developments, which may or may not be accounted for by available information; corresponding alignments can persist as long as risk management schemes can mitigate the impact of unfavorable outcomes.

Time frames and horizons will be discussed further in Chapter 7, and engineering processes in Chapter 8.

3.2.4.4 Transparency, Traceability, Accountability

The digital revolution and the pervasiveness of AI technologies have undermined the traditional pillars of enterprise governance, calling for a new generation of tools aimed at knowledge-based systems. As a result, most tools now claim to integrate various ontologies and/or GenLMs to extend their platforms' capabilities. However, many of these tools rely on homogeneous ontologies designed to function within bounded contexts, in contrast to knowledge-driven systems that are meant to operate in unbounded environments involving different semantics, nondeterministic modes of operation, and unpredictable outcomes. Hence, the importance of transparency, traceability, and accountability. While these issues can be addressed, at least partially, through homogeneous ontologies (e.g., Tropos), this approach only covers the conceptual side and, due to semantic fault lines, can only ensure nominal traceability across organizational and operational environments.

These caveats can be addressed by using ontological prisms that establish explicit purposes for: transparency, regarding commitments to the objectives and resources behind the changes; traceability, for tracking their symbolic footprints; and accountability, for the decisions that depend on both (Figure 3.12).

In that case, intrinsic alignments (#) will be used between use cases (for individual business objectives), projects, and categories, while functional alignments ($\approx$) will concern aspects.

Transparency, traceability, and explainability will be discussed further in Chapter 6, decision-making in Chapter 7, and use cases in Chapters 8 and 9.

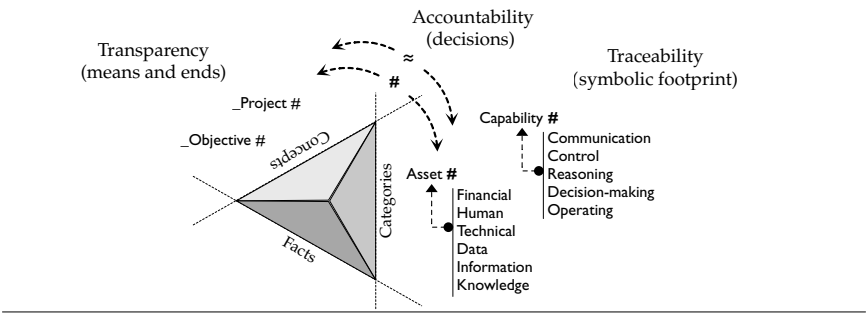


Figure 3.12 Transparency, traceability, accountability.

3.3 Interoperability and Collective Intelligence

Enterprises immersed in digital environments give new relevance to Stafford Beer's vision of viable systems that rely on osmosis and homeostasis to thrive in competitive settings. However, this requires a built-in capability to learn from internal and external interoperability, which can be best achieved through heterogeneous ontologies.

3.3.1 Pushing the Edges, Moving the Lines

Ontologies are most often used to organize domain-specific knowledge into KGs. While this approach may be sufficient for document management systems and conceptual models, it falls short when it comes to knowledge itself. The very nature of knowledge is to be transverse and expansionist, with its edges continuously pushed, its lines redrawn, and its modalities redefined. To ensure such holistic interoperability with knowledge galaxies, ontological prisms rely on perceptions, projections, and experience (Figure 3.13).

3.3.1.1 Perceptions

Perceptions mark the pivot between reality, represented by environment models, and meanings, represented by business models. Perceptions are shaped by the interplay of observations and understanding, with observations informing our understanding, which in turn redefines how we conduct further observations. Such shifts are a critical part of knowledge elicitation and could not be achieved without the interoperability of heterogeneous ontologies. This can be illustrated by the feedback loops between data mining, which feeds environmental and analytics models based on changes in the data landscape (e.g., sales of food products,

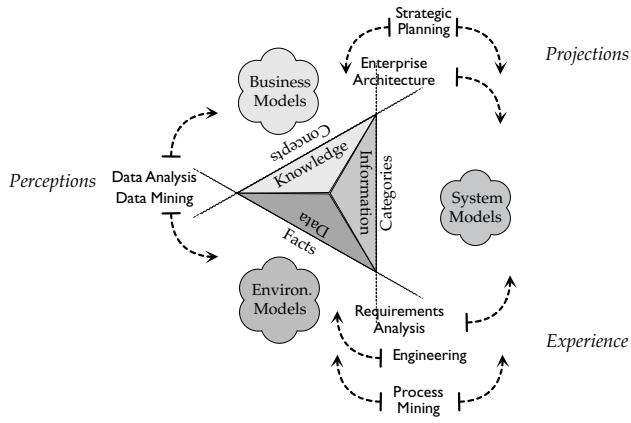


Figure 3.13 Perception, projections, experience.

preferred dishes), and data analysis, which attempts to make sense of emerging patterns (e.g., consumer whims, health concerns, ecological awareness).

3.3.1.2 Projections

Projections mark the pivot between business models and their symbolic realization through system models. Within organizations, they are shaped by the interplay of individual reckonings and collective undertakings, with individual reckonings feeding into collective efforts, which in turn induce a realignment of individual projections. While shifts in the (baffling) lines between individual and collective intelligence are known to be at the core of innovation, their intangibility makes it hard to substantiate. In practice, enterprise architecture and strategic planning can provide a more tangible illustration of the interoperability of heterogeneous ontologies. The former embodies business models into organizations and systems (e.g., food distribution channels), while the latter projects reckonings and undertakings into conjectured realms (e.g., sustainable farming business models).

3.3.1.3 Experience

Experience, whether individual or collective, is the third pillar of knowledge engineering. It serves as a springboard for knowledge expansion and provides feedback to cognizant agents. As such, experience can confirm or refute perceptions and projections, thereby inducing changes in both. Moreover, experience introduces a nonsymbolic dimension to interoperability, at individual as well as collective levels. Beyond individual experience, enhancing organizations'

ability to learn from their collective experience emerges as a primary benefit of heterogeneous ontologies. With ontological prisms, benefits can be achieved through the direct mapping of facts into categories (e.g., requirements analysis), categories into facts (e.g., engineering), or probing the imprints of actual categories (e.g., process mining). From a broader perspective, the multidimensional nature of experience – both individual and collective, symbolic and nonsymbolic – points to a holistic approach to interoperability.

3.3.2 Collective Intelligence and Holistic Knowledge

Framing interoperability in terms of learning and knowledge highlights the importance of the collective dimension, which has been brought about by the digital revolution and the pervasiveness of generative artificial intelligence (GenAI) technologies. However, given the growing gap between the surge of these technologies and the inadequate attempts to control their employment, the question is whether a fusion of organizations with opaque technologies will lead to holistic knowledge or collective blindness?

3.3.2.1 Interoperability and Holistic Knowledge

For all intents and purposes, light and colors epitomize holistic integration, whereby wholes can be repeatedly decomposed into and recomposed from elements, with both elements and rules being redefined indefinitely. Just as prisms do this for light and colors, ontological prisms could serve a similar purpose for knowledge. In that context, interoperability would fulfill holistic tenets about nonseparability, causality, and maturity.

3.3.2.1.1 Nonseparability: Symbolic vs Connectionist Knowledge

The nonseparability principle, which states that spatial and temporal modalities cannot be apprehended separately, is often used as an excuse to overlook any further characterization of holistic features. This oversight is particularly damaging to the integration of symbolic and connectionist knowledge, assuming that this integration is a prerequisite for any form of holistic approach. By contrast, ontological prisms use symbolic layers to combine nonseparability with transparency and traceability, ensuring the integration of symbolic and connectionist contributions through thesauri (which bring together facts with concepts) and taxonomies (which bring together facts with categories).

3.3.2.1.2 Causality: Semantic Layers and Reasoning Tiers

As a corollary of holistic nonseparability, holistic entanglement may also serve as a pretext to ignore causal chains, even though such oversight would undermine

any attempt at traceability. With ontological prisms, the disentanglement of holistic causal chains can be achieved through the semantic layers of thesauri, taxonomies, and domains serving as sieves for nominal, empirical, and formal causalities, respectively (see Chapter 6).

3.3.2.1.3 Maturity: Knowledge Modalities

When considered together, a shallow understanding of the holistic nonseparability of elements and the entanglement of causal chains may seem to leave little room for learning and knowledge engineering; this could be seen as a critical flaw for ontologies. However, the outlook becomes more favorable if holistic principles are applied to the integration of heterogeneous ontologies rather than to the ontologies themselves. In this context, the role of ontological prisms is not to explore domain-specific knowledge but to assess their maturity with regard to the transparency, traceability, and accountability of truth commitments. In simpler terms: what is known, what can be known, how it could be known, and why it cannot be known.

3.3.2.2 Mechanical Brains vs Collective Minds

Setting aside the hype that often surrounds technological breakthroughs, it seems that, for the first time in history, technological advances are outpacing socio-economic and cultural contexts. This observation is supported by various indicators at social, corporate, and institutional levels. At the social level, technologies like GenLMs are widely used irrespective of their intended purpose and capabilities. At the corporate level, the rush for GenAI technologies is driven more by fears of being left behind and herd mentality than by economic rationales. As for institutions, they are still trying to regulate the backlog of "traditional" AI. These converging trends appear to stem from the same root cause: a mind-boggling technology that can serve as a brain prosthesis or a mind enhancer. Addressing the misguided social or institutional uses of GenAI is clearly beyond the scope of this book. Nevertheless, the functioning of brainy organizations immersed in digital ecosystems may still offer valuable lessons for the sustainable and socially responsible deployment of GenAI technologies.

Key Points

Chapter 2 focused on the organization of ontological prisms, while this chapter addresses their mechanics.

First, the terms alignment, integration, and interoperability have been defined in relation to homogeneous and heterogeneous ontologies.

Second, ontological gears have been introduced to enable the dynamic integration of heterogeneous ontologies across three dimensions: meanings (through thesauri), classifiers (through taxonomies), and responsibilities (through domains). Ontological gears have also been used to identify semantic layers in terms of grammar and logic.

Third, abstraction and realization have been defined by the orientation of interoperability and characterized in relation to time frames and knowledge architectures.

Finally, the chapter has redefined the interoperability of heterogeneous ontologies in terms of experience, perception, and projection, establishing its significance for collective intelligence, to be examined in Chapter 4.

4

Ontological Prisms and Governance

An organization's ability to learn and translate that learning into action is the ultimate competitive advantage

Jack Welch, Former CEO, General Electric

This chapter concludes Part I, which presents the organization of ontological prisms (Chapter 2) and their mechanisms (Chapter 3). Its objective is to align the management of data, information, and knowledge with their ontological counterparts of facts, categories, and concepts.

4.1 Environments and Organizations

As social entities, enterprises are set at the cross of three symbolic dimensions: statutory, organizational, and economic. While the digital revolution has brought them into a single symbolic fold, the aim of ontological prisms is to restore these dimensions through the alignment of enterprises with their environments.

4.1.1 Environments: Data and Documents

Before the digital flood, exchanges between organizations and their environments could be monitored through dedicated gates. Nowadays, digital immersion and the erosion of traditional gateways have undermined the established distinctions of origins, calling for explicit management of sources according to their intrinsic footprint and principles of change (Table 4.1):

- Institutional contexts and regulatory authorities are meant to be stable, with changes to content subject to established procedures, such as those of the US Food and Drug Administration (FDA) or the European Union's General Data Protection Regulation (GDPR).

Table 4.1 A taxonomy of environments according to intrinsic footprint and principle of change.

	Environments				
	Institutional	**Professional**	**Corporate**	**Social**	**Personal**
Perimeter	Bounded	Bounded	Closed	Open	Fixed
Changes	Statutory	Agreed	Autonomous	Arbitrary	Set
Examples	US FDA EU GDPR	WC3 UML BPMN	Meta Carrefour McDonald	Facebook LinkedIn	Books Articles Statements

- Professional contexts are established by agreement between the parties. Unlike institutional contexts, their stability is not statutory but contractual, as are their scope and procedures.
- In contrast to professional contexts, corporate contexts have an intrinsic perimeter, and changes in content are not subject to established procedures.
- Social contexts, on the other hand, lack clear perimeters, authority, or rules, and their contents are therefore subject to continuous, random, and informal changes.
- Personal contexts are defined by identified individuals, and their contents are recorded accordingly (e.g., research papers).

These distinctions should be used to characterize the nature and role of data and document inflows, as well as to define the principles of their management.

4.1.2 Enterprise Architecture: Information

Regarding enterprise architectures, the primary objective of heterogeneous ontologies is to ensure the technical and functional integration of assets shared across business units. In our case, these assets are referred to as information.

4.1.2.1 Technical Integration

Like most ontologies, ontological prisms are implemented with Web Ontology Language (OWL), itself implemented with Resource Description Framework (RDF); this will allow for a straightforward integration with enterprise architectures. Regarding ontological content, the primary objective should be a syntactic (or shallow) integration, ensuring the importation of external resources independently of domain-specific semantics. That can be achieved directly with OWL and RDF, or indirectly with Unified Modeling Language (UML) and Extensible Markup Language (XML), for knowledge, information, and data, respectively.

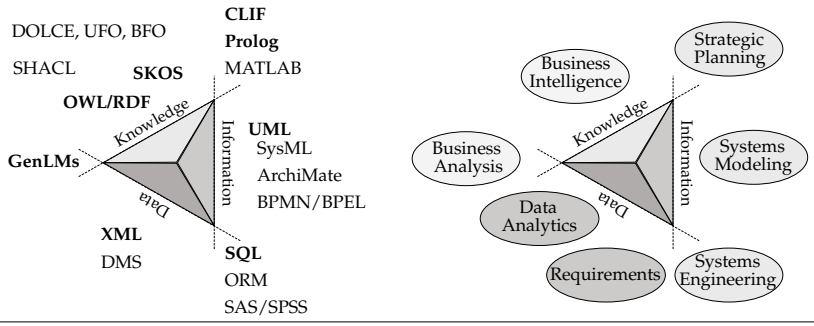


Figure 4.1 Standard interfaces (left) and typical use cases (right).

Leveraging its shared RDF implementation, the Shapes Constraint Language (SHACL) can complement OWL in ensuring the consistency of representations. More specific interfaces could also be supported with RDF-based ontologies (e.g., DOLCE, UFO, or BFO), document management systems (DMSs), general modeling methods and tools (e.g., SysML, ArchiMate, or BPMN/BPEL), or specific ones (e.g., SAS/SPSS or MATLAB). Regarding interoperability, besides generative language models (GenLMs) which will be considered in Chapter 5, four languages are suggested as references: Simple Knowledge Organization System (SKOS), implemented with RDF, for the meaning of facts and concepts, Common Logic Interchange Format (CLIF) and Prolog for formal reasoning, and Structured Query Language (SQL) for the mapping of data into categories. Object-role Modeling (ORM) can also be mentioned with regard to the mapping of facts into categories (Figure 4.1, left).

4.1.2.2 Functional Integration

The functional integration of ontological prisms within enterprise architecture can be defined in terms of typical basic use cases (Figure 4.1, right):

- Requirements, as expressed from the users' perspective and set in actual contexts.
- Data analytics, from a business perspective, aiming at virtual contexts.
- Business analysis, for the alignment of data analytics and business models and objectives.
- Business intelligence for the definition of business models and objectives.
- Strategic planning, for the alignment of business models and objectives with organization and supporting systems.
- Systems modeling, for the alignment of planned, legacy, and currently required systems.
- Systems engineering, for the realization of system models.

Data analytics, business analysis, and business intelligence will be discussed in Chapter 6; strategic planning in Chapter 7; and requirements, use cases, systems modeling, and engineering in Chapter 8.

4.1.3 Organization: Knowledge

From a governance perspective, the role of organizations is to manage data (resources), harness information (assets), and leverage knowledge along the value chain. In this regard, ontological prisms should be ecumenical, and their organizational impact should align as closely as possible with the governance of enterprises and systems.

4.1.3.1 Enterprise Level

Organizations are symbolic architectures, combining authority structures and collaboration mechanisms. Assuming that the former is embodied in the latter, the objective is to determine how collaboration can enhance collective knowledge and decision-making. On that account, a primary distinction should be made between three levels of collaboration:

- *Personal* collaboration is carried out between identified individuals, in person or through communication channels.
- *Functional* collaborations are personal collaborations carried out within socially defined contexts (e.g., a committee) or objectives (e.g., a project team).
- *Organizational* (or institutional) collaborations are carried out between collective entities through representatives and documents.

As suggested by psychology (see Robin Dunbar), personal and functional levels correspond to direct communication. The former is set for around 10 trusted personal contacts, while the latter is for approximately 150 untested social connections. These numbers broadly align with empirical studies of personal and network clusters in both professional and social contexts. While functional collaborations may or may not be mediated by systems, organizational collaborations necessarily rely on system mediation. These collaboration tiers can be used to characterize knowledge-driven organizations in terms of agents (people, organizational units, systems) and cognitive capabilities (observation, reasoning, judgment). Ontological prisms can then ensure a clear functional autonomy between judgment, which is to be carried out by people or organizational units at the functional or organizational level, and observation and reasoning, which are supported by people, organizational units, or systems. These issues will be addressed in Chapter 6 (Reasoning) and Chapter 7 (Decision-making).

4.1.3.2 System Level

With regard to information systems, the aim of an organization is twofold. On one hand, it is to separate concerns: to know the facts (data), define values and intents, set objectives, establish plans (knowledge), and manage the respective representations (information). On the other hand, ontological prisms must remain neutral to ensure that their adoption can proceed smoothly, regardless of the idiosyncrasies of the host organizations. For enterprises regarded as viable systems immersed in digital environments, the immediate objective is to enhance their ability to track changes and exploit opportunities. In the long term, the goal is to improve their sustainability. To that effect, enterprises must transform their organization into a nervous system that can seamlessly integrate the perception of their environment, the projection of their means and ends, and returns on experience. This can only be achieved through a tight integration of organizations and supporting systems, and, as a corollary, flattened organizational structures. It follows that ontological prisms should not introduce additional layers to organizations but should instead aim to align their remits with existing ones (Figure 4.2).

Data, document, and information managers, as well as enterprise architects, are well-defined roles whose responsibilities should naturally extend to ontological prisms. The case of the knowledge manager is different, as their role involves managing the knowledge shared among various ontologies, which is a relatively new function. While they rely on existing skills and competencies, the cross-functional aspects associated with thesauri, taxonomies, and domains must also consider the dual responsibilities introduced by ontological prisms: content management and conceptual modeling (thesauri managers), data analysis and system modeling (taxonomists), and ontologies and system modeling (domain managers).

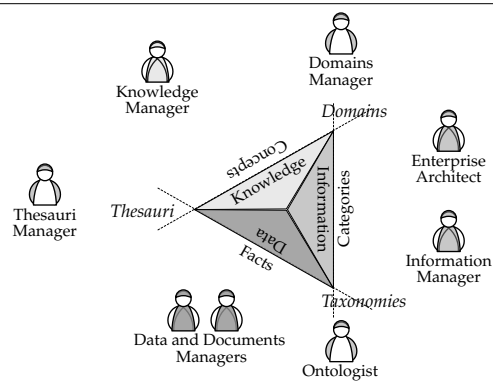


Figure 4.2 Ontological prisms' main remits.

4.1.3.3 Learning Organization

If learning means pushing the edges of knowledge, collective learning can be achieved through a twofold extension of knowledge: from individuals to organization, and from implicit to explicit. Taking into account enterprises' comprehensive digital transformation and the ubiquity of systems' artificial intelligence (AI) components, the learning capability of organizations can be represented by crossing the kind of agent (people or systems) with the kind of knowledge (implicit or explicit) (Figure 4.3).

The fabric of organizations' collective knowledge can thus be developed through the weaving of four threads:

- An in-depth integration between personal and direct collaboration (implicit knowledge) and functional (nonhierarchical) and mediated collaboration (explicit knowledge) (a).
- Transparent and traceable processes across connectionist and symbolic tiers of representation, typically involving machine learning and knowledge graph (KG) technologies (b).
- Enhanced user experience and direct collaboration through feedback from intelligent systems (c).
- Integration of organizational and systems levels that secure a functional distinction between judgment and decision-making, which should be exclusively carried out by people at the functional or organizational level, and observation and reasoning, which can be carried out at all levels (d).

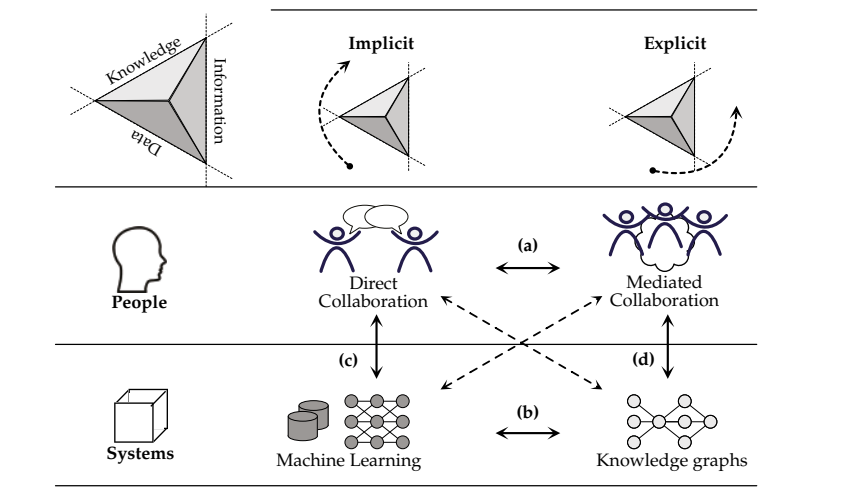


Figure 4.3 Knowledgeable organizations.

The weaving of this fabric will determine the effectiveness of knowledge value chains.

4.2 Data and Value Chains

Starting with ontological concepts, our focus has shifted to facts for a general perspective, and then to data for an enterprise perspective. When viewed through ontological lenses, data have two key characteristics: they reflect observed facts free of interpretation, and they can be managed, provided that their management does not introduce interpretative biases. In practice, data come in one of two forms: datasets and documents.

4.2.1 Data and Metadata

4.2.1.1 Datasets

From an ontological perspective, datasets can be defined as containers of labeled digital content. Datasets can be characterized by their origin: institutional, professional, corporate, social, or personal; and their status: open, proprietary, or marketed. These distinctions are doubly critical as they can be used to:

- Draw the line between data that can be legitimately associated with individuals and data subject to privacy regulations.
- Identify the data products (see Section 4.2.1.3.2) that companies would like to buy, e.g., from data factories, or to package data products that companies would like to market and sell.

It must be noted that datasets derived from managed categories are best understood as designed data. Ontological prisms can also introduce a third kind of dataset for unlabeled digital contents – things still lacking names and can only be pointed to. Such datasets can serve as entry points for both thesauri and taxonomies, the former to introduce new meanings, the latter to elucidate business requirements. As facts, datasets can be characterized by modalities also documented through metadata.

4.2.1.2 Documents

From an ontological perspective, documents can appear as physical objects (a), as symbolic containers of signs (b), as references to symbolic representations (c), or as containers of symbols (d) (Figure 4.4).

DMSs are white boxes whose contents must be managed in every dimension: capture, storage, integration, retrieval, security, life cycle, etc. When paired with ontological prisms, DMSs should present documents as facts that are as

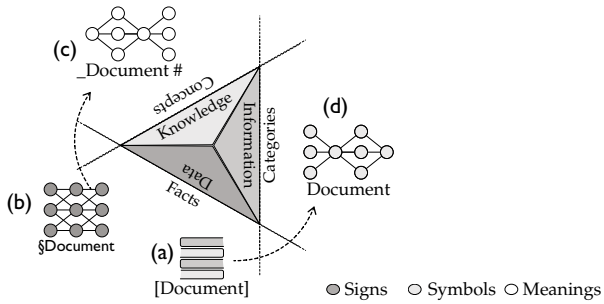


Figure 4.4 Documents: (a) physical objects, (b) symbolic containers of signs, (c) as references to symbolic representations, or (d) as containers of symbols.

fine-grained as possible, while ensuring that whatever can be known about a document context can also be known for users of its detailed content. This information is meant to be encapsulated in what is commonly referred to and managed as metadata.

4.2.1.3 Metadata

As the primary resources of ontologies, datasets, and documents must be managed in a way to ensure their consistent and continuous mapping with the relevant context as defined above: institutional, professional, corporate, social, or personal. That's the objective of master data management in general, and of metadata in particular.

4.2.1.3.1 Objectives

Words can be misleading, as illustrated by metadata, which has nothing to do with data abstractions and everything to do with data management. The rising importance of metadata stems from a two-pronged challenge: the vast amount of available data and its exponential growth, and the fact that data has become a critical resource for a wide range of knowledge-based industries and generative artificial intelligence (GenAI) applications. However, the challenge does not arise from the scarcity of the resource, but rather from its abundance and its biased, partial, and ephemeral nature. For ontological prisms designed to support the engineering of knowledge across various contexts and concerns, managing the flows of data resources is essential; but there is a catch: how to manage data commons without encroaching on the wide range of users' domains, semantics, and purposes.

4.2.1.3.2 Representation

As already noted, semantics, and therefore language, mark the limits of facts; beyond that line, interpretations cannot be avoided. To ensure regulatory compliance, data resources must be initially (i.e., until becoming data products)

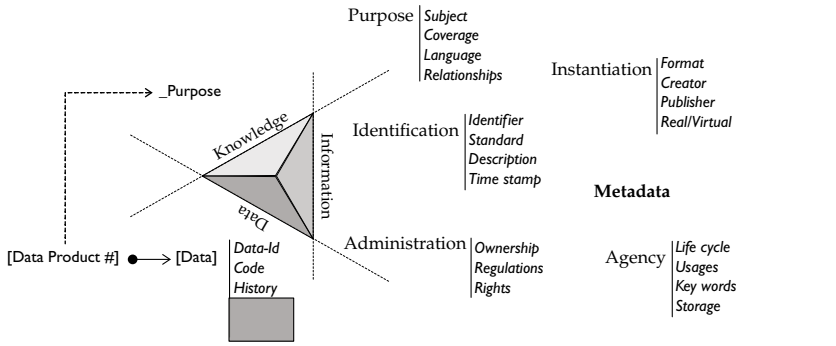


Figure 4.5 Metadata as information about data.

managed as black boxes that display characteristics related to provenance and modus operandi, without providing specifics about the nature of their contents. That typically includes the origins, form, past and present ownership, life cycle, reliability, format, security and confidentiality constraints, and destinations. On that account, metadata is best understood as information about data (Figure 4.5):

- Intrinsic modalities: instantiation (provenance) and identification.
- Structure, format (how to access the contents), e.g., relational schemas.
- Life cycle, keywords, processes
- Ownership, regulatory constraints, authorized accesses, etc.

Along with internal management needs, the widespread surge in GenAI and its dependence on massive, boundless, and reliable quantities of data are driving a customer-centric approach to metadata. This is illustrated by the FAIR initiative, which ensures that data products are findable, accessible, interoperable, and reusable. However, FAIR-like approaches can only go halfway in meeting the needs of GenAI, particularly concerning data, statutory, and temporal modalities.

4.2.1.3.3 Data Products

Most metadata approaches focus on data as a corporate resource best managed like a commodity, independent of its domain-specific characteristics. However, for systems immersed in digital environments and permeated with GenAI technologies, this understanding comes with a double caveat: the role of data in the knowledge value chain cannot be properly assessed, nor can the role of documents in corporate environments. Data products are designed to stand somewhere between raw and managed data: the product is managed but not individual elements. They serve as reusable, self-contained packages that combine datasets, documents, metadata, structure, and semantics, enabling their integration into business value chains.

4.2.2 Knowledge and Value Chains

While there is now a general consensus that data is at the cornerstone of any effective AI strategy, attention is mostly focused on quality issues, while the role of data in the knowledge value chain is generally overlooked. This shortsighted approach can be attributed to the flat (2D) modeling paradigm and its corollary: value chains can be constructed from only one type of link, from data to representation, which fails to accurately depict where the value originates.

4.2.2.1 Added Value

Framed by ontological prisms, value chains can be constructed from three types of links connecting resources (data), assets (information), and business value (knowledge). Gains in added value can thus be explained by improvements in the management of resources and assets (e.g., through better data products and/or modeling methods and tools), and a more effective alignment of products and services with business environments (Figure 4.6).

Such improvements are arguably the bread and butter of enterprise performance, and they can be achieved with or without ontologies. However, when considered on their own, they only explain a fraction of the decisive and continuous gains in added value that stem from synergies and cross-effects and, more broadly, holistic improvements. Using ontological prisms, such improvements can be assessed in terms of interoperability as defined in Chapter 3: perceptions, projections, and experience.

4.2.2.1.1 Perception and Added Value

Perception is where conceptual commitments are made about the meaning of observations and their relevance to business models. Added value primarily arises

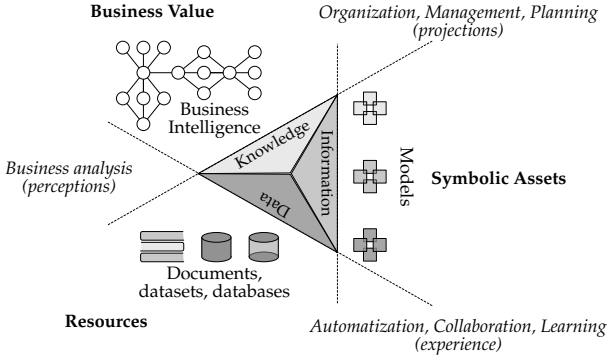


Figure 4.6 Sources of added value.

from a better understanding of business opportunities, which can be achieved through a more effective use of GenAI capabilities (such as refactoring, computing, pattern matching, etc.). In turn, these improvements can enhance the back-and-forth between the definition of business concepts derived from data analytics on the one hand, and the design of analytic schemes based on business objectives on the other hand. Ultimately, these improvements are meant to be embodied in the design and use of data products.

4.2.2.1.2 Projection and Added Value

Projection is where business commitments are made, translating business intelligence into collective endeavors. These commitments are shaped by the interplay of individual assessments and collective evaluations before evolving into collective engagement. Added value stems from the soundness of these commitments and from the substance of the collective support backing them. Both can be significantly enhanced through ontological prisms: the former by the dynamic association of decision-making processes and knowledge chains, and the latter by the merging of individual and collective reasoning, which enhances transparency, traceability, explainability, and accountability.

4.2.2.1.3 Experience and Added Value

Experience is where operational commitments are made based on the assessment of outcomes and lessons learned from process execution. Added value can thus come from direct improvements in the quality of outcomes and/or processes, a better assessment of operational risks, and, from a broader perspective, enhanced collective learning. By leveraging the interoperability of ontological prisms, these benefits can typically be achieved through a systematic integration of quality procedures in product and service development, leading to higher standards and improved anticipation and reactivity. More broadly, transparency, traceability, and explainability will contribute to smoother learning curves and enhanced collaboration.

4.2.2.2 Transaction Costs and Value Chains

Given business processes and the sources of added value, the objective is to organize the components of value chains to maximize total added value. Transaction costs can serve as a yardstick.

4.2.2.2.1 Transaction Costs

The brainchild of Nobel Prize winner Ronald Coase (see Coase R.H., (1990)), the concept of transaction costs pertains to the evaluation of outsourcing products or services in comparison to their internal costs. To that end, transaction costs consider the primary costs of obtaining all the necessary data, information, and

knowledge, along with the secondary costs associated with the negotiation and successful execution of transactions. Coase's concept becomes particularly relevant when all enterprise transactions are conducted in digital environments and involve a mix of people, organization, and systems. Most importantly, it emphasized the pivotal role of information in the design of value chains. When framed by ontological prisms, transaction costs can be fine-tuned to address the trading of data, information, and knowledge.

Regarding the costs of data, a distinction must be made between primary data directly obtained from environments and secondary data obtained from business activities. The former can be obtained freely from public sources, with additional refining costs for data products, or purchased from so-called data factories. The latter is already refined and may be available at no cost or be priced internally based on upstream acquisition and management costs. Both should rely on an agreed-upon definition of a data product, enabling comparisons between traded options.

In contrast to data and knowledge, which can be traded and consequently priced externally, information is intended to be priced internally. This pricing will take into account the management of supporting systems, as well as the cost of asset depreciation. It will also cover the costs of ensuring the security and confidentiality of information, along with compliance with privacy regulations.

While the role of knowledge in value chains is becoming a critical issue, its pricing remains challenging, as we will see further with the issue of intellectual property.

4.2.2.2.2 Outsourcing Knowledge-intensive Activities

Advances in GenLMs and, more generally, agents with AI capabilities have triggered attempts to outsource certain information processing tasks to such agents, whatever they may be, potentially making people redundant in some positions along the knowledge value chain. However, delegating tasks to supposedly intelligent yet opaque AI agents raises functional and economic concerns. From a functional perspective, outcomes have limited value without transparency (resources) and traceability (reasoning) that can explain margins of error, no matter how small they may be. From an economic perspective, in addition to the costs of computing, the net returns must take into account the costs of skilled human resources needed to train the agents and supervise the outcomes. When all induced costs are considered, economic incentives appear to be primarily for agents operating in bounded contexts where full transparency and traceability can be achieved. Still, as discussed in previous chapters, these issues, along with accountability, can be partially addressed through ontological prisms, assuming that knowledge-intensive tasks can be outsourced according

to cognitive capabilities: observation, reasoning, and judgment. Such tasks could thus be aligned with corresponding segments of value chains: resources (data), assets (information), and services (knowledge), respectively.

4.2.2.3 Documents and Blockchains

As already mentioned, current metadata approaches focus on managing data resources uniformly without taking into account their role in broader environments. While this blind spot may seem primarily relevant for documents, the distinction between data products and documents is undermined by the pervasiveness of GenAI technologies, which calls for a more discerning approach.

4.2.2.3.1 Documents as Currency

When viewed in terms of exchanges between organizations and their environment, data products and documents can be compared to currencies, with their meaning, status, or role defined accordingly:

- Direct exchange (barter) of nondescript datasets or documents.
- Data products as a medium of exchange between organizations with different business or technical concerns.
- Documents or datasets stored for future reference or reuse over time, such as time series, models, and test cases.
- Documents with statutory value established by environments, such as regulations and contracts.

As with currency, data and documents are often associated with overlapping options, leading to exponential metadata complexity. At the corporate level, ontological prisms can address the issue by extending metadata with explicit references to environments. At the professional or institutional level, blockchains may serve as clearinghouses.

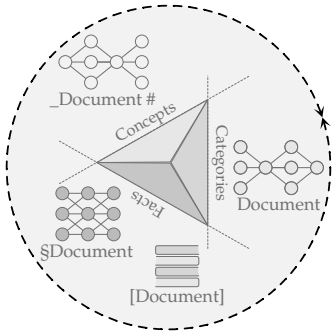
4.2.2.3.2 Blockchains

Blockchains are encrypted digital ledgers that can be used to track and authenticate the provenance of physical or symbolic artifacts without disclosing their contents. Whatever happens in a domain of concern is embodied in a new block along with a cryptographic hash of the previous block and a timestamp. Since each block contains information about the previous block, blockchain transactions cannot be changed retroactively without all-inclusive network agreement.

Blockchains thus epitomize documents as facts and ensure their intrinsic identity and functional aspects over time (Figure 4.7).

They can be designed based on specific needs and environments. For instance:

- Public blockchains allow anyone to request or check transactions, add new blocks, and view every block on the blockchain.



Block	Institutional	Professional	Corporate	Social	Personal
Public	+	+	-	+	-
Consortium	+	+	-	-	-
Private	-	-	+	-	+
Sidechains	-	-	+	+	+

Figure 4.7 Documents and blockchains.

- Consortium blockchains are governed by business or professional associations, with access based on rules.
- Private blockchains are governed by individuals or groups of individuals, with access defined on an individual basis.
- Sidechains are secondary blockchains designed to complement primary ones.

Blockchains can be applied to a wide range of fiduciary contexts, including finance, artworks, and intellectual property.

4.3 Data Protection and Intellectual Property

In addition to the wide range of functions supported by established information management systems, ontological prisms may offer key benefits regarding regulatory compliance. In line with the rationale presented in Chapter 2 and the holistic perspective introduced in Chapter 3, regulatory issues can be framed within a two-pronged ontological perspective: data protection in relation to data and information, and intellectual property in relation to knowledge. Data protection, as commonly understood, pertains to the data itself (e.g., privacy) as well as its representation through categories, i.e., information (e.g., integrity). Intellectual property, on the other hand, relates to documents (e.g., copyrights) and concepts (e.g., patents). Complexity and maturity are introduced as overall benchmarks.

4.3.1 Data Protection

Data protection encompasses the continuum from raw data to data products and managed data, i.e., information. As far as ontological perspectives are concerned, data is positioned at the entry point of knowledge engineering processes, where it

is meant to be shared and interpreted by different ontologies. In order to be equally available for all while being specifically relevant for each, data management must remain ecumenical, with domain-specific semantics and purposes managed separately for data products. Taking that distinction into account, ontological prisms should only consider characteristics deemed necessary for their use: typically, format (how to open the boxes), provenance (who handled the boxes), and shelf life, while ignoring specifics about the meaning of their contents. Nevertheless, ontological prisms must address constraints related to access and usage that cannot be managed without opening the boxes, typically concerning privacy and confidentiality. To that end, ontological prisms should maintain a formal and functional distinction between data products and managed data (information), with the former tied to external instances and the latter defined by the categories of managed surrogates.

4.3.1.1 Principles

Habits are hard to change, and mental habits are the hardest, as illustrated by the confusion between data and information. First, there is the evidence: data is public, whether it is freely available or marketed, while information is proprietary. Then there is the law: compliance with data protection regulations means that enterprise architectures must support a built-in distinction between what is legitimate and what is not. Finally, there is ontological substance: data is amorphous and needs structure and semantics to be transformed into information. Ontological prisms address all these issues by introducing a conceptual distinction between data and managed data. Based on that distinction, confidentiality regulations are meant to protect managed data, while privacy regulations aim to do the same for unmanaged data, data products included.

4.3.1.2 Institutional Approach: General Data Protection Regulation

Adopted in 2016, the European Union's GDPR represents a significant legislative milestone for data privacy. Its implementation has also revealed the inherent challenge of regulating both private and corporate realms simultaneously. Difficulties begin with terminology. Despite a broad consensus regarding the nature of data protection issues, controversies persist over specific terms, likely due to confusion between data and information. From a pragmatic perspective, a distinction between data and information can be neatly aligned with the distinction between private and corporate realms: the former pertains to identified individuals, while the latter relates to categories of managed surrogates.

Given that terminologies constitute the cornerstones of regulations, the confusion between data and information has direct consequences for the GDPR's definition of "personal data" as "any information relating to an identified or identifiable natural person ('data subject')." Logically, that definition implies an

equivalence between "data" and "information," an assumption that undermines the very purpose of the GDPR. This is because, given enough anonymized data, systems can produce personal information, even without the use of AI technologies. Ontological prisms could help address these issues.

4.3.1.3 Ontological Approach

Taking advantage of the distinction between data and information, privacy and confidentiality issues can be defined in terms of the osmosis between information systems (managed data) and individuals identified in various environments. Based on that rationale, data protection can be expressed in terms of the alignment of taxonomies introduced in Chapter 3.

Taking for granted that no personal data can be managed without owner's explicit consent, confidentiality (Figure 4.8a) means that managed personal data (e.g., customers' files) cannot be obtained directly from environments, and privacy (Figure 4.8b) that database schemas cannot be used to identify individuals from datasets (e.g., all John Does who like white wine with hamburgers).

It follows that, in addition to the data security directly provided by systems, ontologies should ensure that any linking of features and taxonomies (encoded through parameters) with identified individuals can only be achieved through legitimate business applications. This is typically accomplished by preventing the direct or inferred association of features (≈) with identified individuals (#). It is important to note that confidentiality and privacy issues can be represented in terms of intrinsic (#) and functional (≈) properties in both enterprise (data, information, knowledge) and ontological (facts, categories, concepts) perspectives.

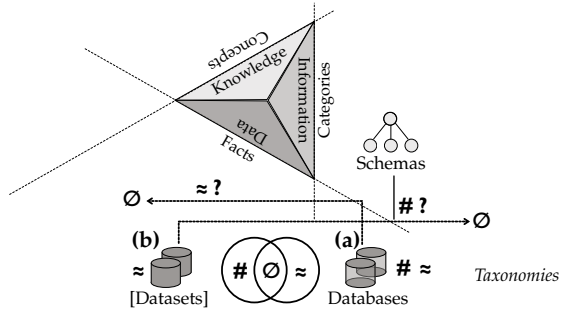


Figure 4.8 Confidentiality (a) and privacy (b) of identities (#) and personal features (≈).

4.3.2 Intellectual Property

Ontologies are designed to question the nature of things, raising the issue of artifacts, which are things that originate from design. Furthermore, there is the question of the digital existence of all things: for all intents and purposes, reality has become digital, blurring the distinction between tangible and intangible assets. For creative works, whose legitimacy relies on intent and provenance, this represents a clear threat. This threat is now heightened by the ubiquity of GenLMs (see Chapter 5), which can erase any documented provenance of meaningful content, undermining the foundations of the intellectual property regulatory framework. Ontological prisms address this issue by focusing on facts instead of things and by introducing a conceptual distinction between facts and data. This differentiated approach can help manage the intellectual property of creative works (copyrights) and useful designs (patents).

4.3.2.1 Creative Works and Copyrights

Copyrights address the ownership of creative works, which can be succinctly defined as singular and nominal: singular because creative works are meant to be self-defined objects or phenomena that cannot be reduced to established designs; and nominal because creative works are intended to be identified as a whole, rather than by detached features.

When framed by ontological prisms, creative works can be characterized in terms of taxonomies and thesauri, allowing for copyright management: taxonomies to ensure singularity, and thesauri to ensure exclusive, nominal, and holistic identification (Figure 4.9).

A routine defined in this way could be applied to any type of documentary piece, whether it be music, text, graphics, or a combination of all of them: first, check that the piece cannot be categorized under an existing category (Figure 4.9a); second,

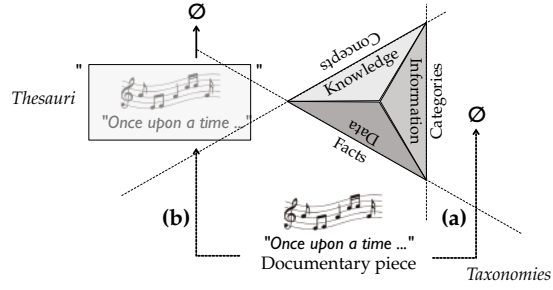


Figure 4.9 Copyrights apply to creative works with singular (a) and nominal (b) characters.

ensure that the piece cannot be associated with any name other than the whole its own instantiation (Figure 4.9b).

4.3.2.2 The Quagmire of Patents

Once limited to physical apparatuses, patents nowadays pertain to the design and modus operandi of the whole range of physical or symbolic artifacts. It ensures that:

- Compared to copyrights and privacy rules, which pertain to actual documents or datasets, patents may involve the three ontological realms: facts, concepts, and categories.
- Compared to copyrights, patents do not secure the rights to use what is described, only the rights to exclude others from using patented descriptions.

That puts patents in a conundrum: the first characteristic should place them at the core of the knowledge value chain, while the second hinders a sound assessment of business opportunities. This conflict could have been overlooked as long as knowledge wasn't bought and sold for its own sake; however, the patenting of software has paved the way for the patenting of any kind of symbolic construct, most notably representations of biological structures (e.g., stretches of DNA), financial products, and business processes.

As a simple illustration from our culinary context (Figure 4.10, left), a patent could be submitted for the recipe of a molecular dish (a), with a description of the ingredients and cooking steps (b). The patent would protect the realization (c) of actual dishes using that representation.

Extending the scope of patents to include symbolic representations themselves, rather than just actual targets, has introduced an intrinsic bias in the intellectual property legislative framework. This shift has direct consequences for both defensive and offensive corporate patent strategies. On the defensive side, the application of multiple preventive patents relies on the confusion between intent and feasibility, which is inherent when symbolic representations are patented. On the offensive side, patent trolls are created by shell companies whose only

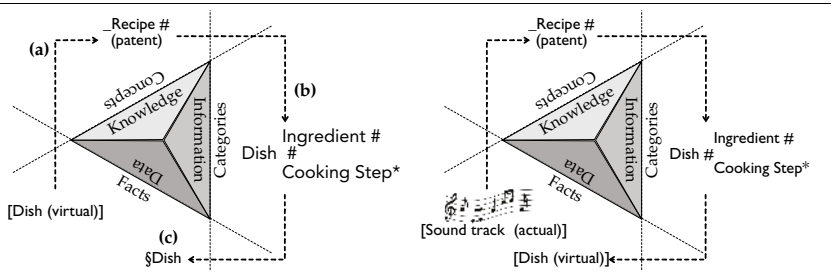


Figure 4.10 Proper (left) and troll (right) patents.

purpose is to acquire as many seemingly pointless patents as possible in the hope that one of them will turn into a lucrative industrial stranglehold.

Once patents are allowed to mix intents and representations, distinguishing between legitimate and spurious patents first requires an explicit differentiation between intents (concepts) and representations (categories). Patents could then associate domains, encompassing concepts (patent purposes) and categories (patent implementations), characterized by symmetrical intrinsic and functional alignments of remits (see Chapter 3). In our culinary context, one can envision a troll patent for a GenAI agent capable of transforming musical pieces into recipes (Figure 4.10, right).

4.3.2.3 Intellectual Property and Generative Artificial Intelligence

GenAI introduces a practical, business-oriented dimension to cognitive and philosophical issues: how to draw the line between inspiration and creation, and how to distinguish exploration from discovery.

Plagiarism and its more respectable cousin, inspiration, have always been integral to creative work; it could not be otherwise, given the intrinsic social nature of the human species. The novelty with GenAI is that agency, until now a human privilege, has been granted to digital pens, paintbrushes, and musical instruments, breaking the personal lineages between creative works and their sources of inspiration. In theory, assuming that GenAI agents can be trained on controlled resources, ontological prisms could be used to deconstruct creative lineages. In practice, however, this would require full transparency regarding resources and complete traceability concerning algorithms.

Conversely, GenAI may provide a way out of the current patent quagmire by untangling the intensional, extensional, and logical aspects of patents. This is because patents do not pertain to perceptions like copyrights do, but to projections (see Chapter 3). Set within ontological prisms, perceptions mark the boundary between reality and meanings; as such, they are shaped by culture and cannot be reduced to algorithms. In contrast, projections do not pertain to meanings but to representations shaped by categories that can be processed through algorithms. When supported by ontological prisms, GenAI could thus help to sort out the patterns of intents, designs, and virtual realizations underlying submitted patents.

4.4 Complexity Management

The flattening of established system boundaries brings the chaos of external environments to enterprises' doorsteps. Unless a robust and flexible organization can serve as a buffer, external disorder risks progressively clog enterprises' organization and systems with sequences of piecemeal ad hoc adjustments. To prevent

such debasement (or entropy) of their information systems, enterprises need to leverage the benefits of digital osmosis and homeostasis across their architectures. Ontological prisms could help with that.

4.4.1 The Fabric of Complexity

From an ontological perspective, complexity can only be addressed through symbolic representations. When ontologies are homogeneous, their complexity directly matches the complexity of their representation, typically KGs, reducing complexity management to a technical issue. In contrast, when heterogeneous ontologies must be represented, managing complexity requires a distinction between extensional and intensional complexity, the former set directly between observations, documents, and datasets (facts) and supporting systems (categories), while the latter is set indirectly through enterprise values, policies, and objectives (concepts) (Figure 4.11).

4.4.1.1 2D Complexity

Extensional (or 2D) complexity pertains to bounded factual, semantic, and organizational contexts: there are no uncertainties with regard to environment, meanings are unambiguous, and remits are aligned, respectively. In theory, extensional complexity is minimized when all relevant facts can be accounted for by categories. This can be accomplished through the alignment of classifiers (see Chapter 3). For instance, a comprehensive and consistent mapping would ensure that all that is known about the ingredients used in dishes, e.g., their molecular signatures, can be reflected in categories, subtypes, and powertypes for gluten, calories, proteins, and carbohydrates (Figure 4.12).

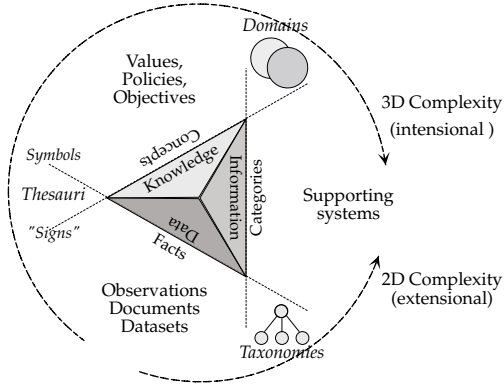


Figure 4.11 Extensional (2D) and intensional (3D) complexity.

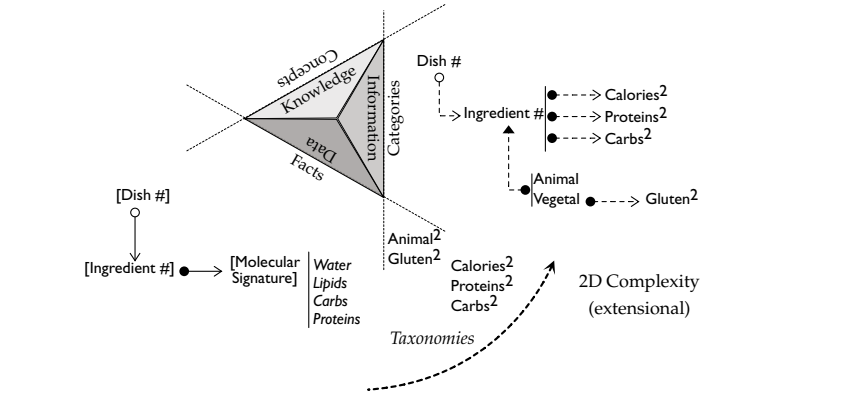


Figure 4.12 2D complexity.

In practice, actual extensional complexity can be estimated by applying established function point methods to models.

4.4.1.2 3D Complexity

Contrary to extensional (or 2D) complexity, intensional (or 3D) complexity takes into account the uncertainties of observations, unbounded semantics, and alternative or conflicting remits. While these factors are intertwined and their outcomes cannot be sorted unequivocally, their management can still be differentiated by their origin and aligned with the time frames defined in Chapter 3 (Figure 4.13):

- Deterministic (or operational) complexity coincides with extensional (2D) complexity. It can be managed through experience, with perceptions and projections remaining unaffected.
- Stochastic complexity arises from changes in the environment that are subject to probability distributions. It cannot be managed solely through operational adjustments (experience), and adjustments in perceptions may also be necessary.
- Non-stochastic complexity arises from changes in the environment that cannot be described through probability distributions. Managing this complexity requires adjustments to projections.

In our culinary context, operational complexity would reflect weekly routine adjustments to meteorological conditions, tactical complexity would reflect monthly adjustments to observed dietary preferences, and strategic complexity would address potential misalignments between business models, such as those arising from mergers or new regulations.

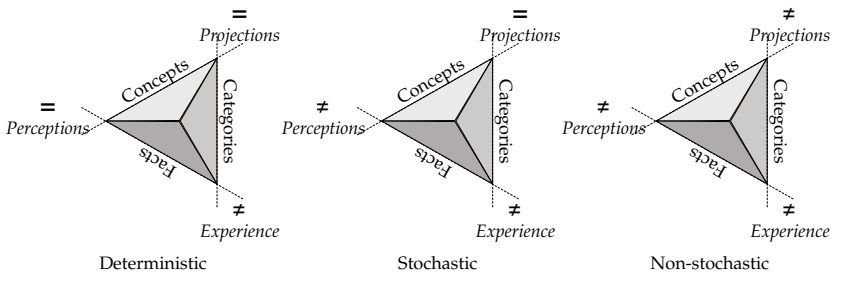


Figure 4.13 3D complexity.

4.4.2 Organization and Complexity

Traditionally, i.e., before the digital revolution, organizations were designed to manage people and responsibilities. Nowadays, data floods and knowledge-driven competition are undermining the relevance of the traditional distinction between function-based organization and supporting systems, promoting instead distributed agent-based architectures. Regardless of the agents' nature and capabilities, this raises the question of how to assess organizations in the digital era; the cybernetic answer lies in their ability to manage complexity.

4.4.2.1 Organization and Complexity Management

Complexity is the substrate of business competition, and managing that complexity is the primary factor behind an enterprise's competitive edge. Ideally, complexity would be best managed in relation to scope. In practice, however, due to the lack of reliable complexity metrics, a second-best option is to consider changes instead of the complexity itself. This can be achieved through three organizational tiers aligned with the complexity layers introduced above: operational (experience alignments), tactical (experience and perception alignments), and strategic (experience, perception, and projection alignments) (Figure 4.14).

Managing operational (deterministic) complexity is intended to focus on changes within self-contained situations. This approach allows issues to be addressed locally through problem-solving without affecting overall complexity. It follows that managing operational complexity should not interfere with perception or projection.

Managing tactical (stochastic) complexity aims to address changes in meanings as well as taxonomies necessitated by shifts in business opportunities. While these changes are inherently specific, the facts that trigger them are not, as they are shared across different domains (and therefore ontologies) where they can be interpreted differently. For this reason, taxonomies (experience) and thesauri

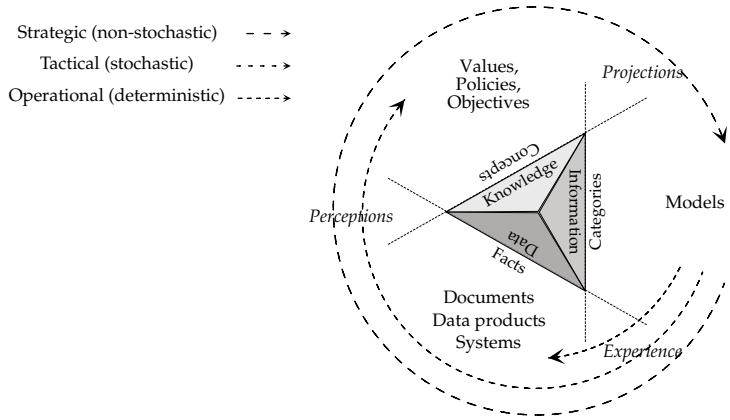


Figure 4.14 Organization and complexity management.

(perception) must serve, respectively, as operational and semantic gateways that separate the commons of facts from domain-specific interpretations. Nevertheless, tactical alignments should not affect intents and objectives, and consequently, projections.

Managing strategic (non-stochastic) complexity aims to address organizational changes necessitated by unexpected disruptions in business environments. The alignments induced by these changes can potentially affect objectives and assets, and consequently, domains' responsibilities and, more generally, projections. Moreover, changes in strategic complexity are likely to cascade into operational and tactical management, thereby impacting experiences and perceptions.

Given the comprehensive digital immersion of enterprises, one might assume that metrics can be developed to assess changes in complexity at the operational, tactical, and strategic levels, thereby enabling the evaluation of organizations' performance.

4.4.2.2 Ontological Complexity and Systems Entropy

As understood in cybernetics, the entropy of a system is the quantum of energy that cannot be converted into mechanical work. Formalized by Shannon in terms of leakage along communication channels, entropy has become a pillar of information theory. From an ontological perspective, entropy represents the amount of data (a reflection of facts, with documents included) that cannot be converted into information (data organized into categories), either directly through taxonomies or indirectly through thesauri and domains. It follows that system entropy can be decomposed and aligned with the sources of ontological complexity.

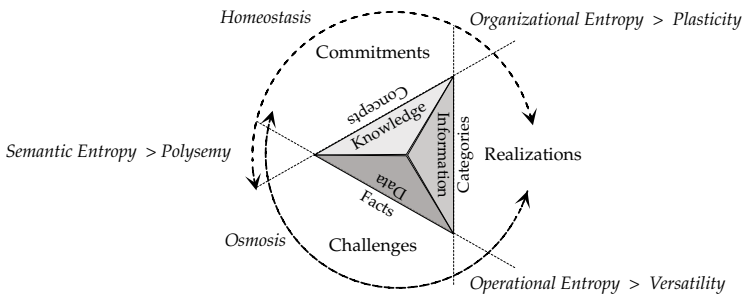


Figure 4.15 Entropy through ontological prisms.

First, ontological prisms are used to extend the concept of systems to organizations and knowledge. From a governance perspective, challenges, commitments, and realizations can be expressed in terms of facts, concepts, and categories, respectively. This parallel enables a detailed analysis of operational, organizational, and semantic entropy (Figure 4.15).

Operational entropy pertains to the discrepancies between the partitioning of challenging facts (as observed through data) and their management (as reflected by information systems). Such entropy can be attributed to emerging challenges or misalignment of solutions. Operational entropy can be redressed through versatility, which is the capacity of a system to change its functions without altering its architecture or increasing its complexity.

Organizational entropy refers to discrepancies in responsibilities between commitments (concepts) and realizations (categories). Such entropy can be attributed to a flawed understanding of intents and objectives, or misguided alignments of domains and responsibilities. Organizational entropy can be redressed through plasticity, which is the capacity of a system to change its architecture without affecting its functions or increasing its complexity.

Semantic entropy refers to the discrepancies in meanings between commitments (as expressed through the concepts used for values, policies, and objectives) and the terms used to describe challenging facts. This type of entropy can be attributed to unnecessary increases in thesaurus complexity or misguided alignments of meanings between observations and intents. Semantic entropy can be redressed through generative interfaces and polysemy, which is the alignment of different meanings for the same word depending on the context of conversations.

From a cybernetics perspective, entropy management can leverage two key benefits of digital immersion, osmosis and homeostasis.

Osmosis is the exchange of operational data between systems and their environment, allowing for unmediated adjustments between challenging facts and supporting systems. Given the inherent porosity of digital boundaries, data mining can provide comprehensive and timely accounts of external events, and process mining enables fine-grained tuning of operations. One step further, enterprises can bolster lean and just-in-time engineering processes through a seamless integration of (digital) business logic into software, thus reinforcing the feedback between engineering and operations. Osmosis can also operate between semantic layers, e.g., through GenLMs.

Homeostasis is the ability of organizations and systems to align their governance with changes in their environments, allowing for continuous adaptation of commitments and realizations. As long as adjustments are limited to extensional complexity – meaning that objectives are not affected – this can be achieved directly through experience and osmosis. However, in cases such as mergers or acquisitions, the whole of taxonomies and experiences, thesauri and perceptions, and domains and projections, must be reviewed and possibly realigned. This kind of holistic overhaul will be much less challenging when supported by ontological prisms.

Framing complexity and entropy through ontological prisms establishes organizations as living organisms whose evolution and sustainability in their ecosystem are determined by how effectively they can process data into information and utilize both as knowledge.

4.4.3 Complexity and Evolution

Evolution means adaptation of organisms to their environments, which involves a balancing act between complexity and entropy. We conclude this chapter by drawing a parallel between the evolutionary model of living organisms within their ecosystems and the evolution of corporate entities as viable systems operating in competitive digital environments.

4.4.3.1 Natural Evolution

The extended evolutionary synthesis proposed by Eva Jablonka and Marion J. Lamb (see Bibliography) defines the evolution of living organisms in four dimensions:

1. Genetics: changes in phenotypes (the actual forms and capabilities of organisms) are carried out through the combination of genotypes (the coded parts of DNA).
2. Epigenetics: changes in genotypes between generations (genetics) can also take place in transmission, induced by the noncoded (or perhaps still undeciphered) parts of the DNA (or epigenetics).

3. Behavioral: changes occur through socially mediated nonsymbolic learning (e.g., imitation).
4. Symbolic representations: changes occur through socially mediated symbolic learning (e.g., political systems).

Assuming that changes in these dimensions can explain the evolution of living species, could there be lessons to be learned from a comparison with enterprises?

4.4.3.2 Evolution of Organizations

Epitomized by Stafford Beer's vision of viable systems, the parallel between corporate entities and living species is underscored by the transformation of organizations into digital entities that combine sensory-motor backbones with neuro-symbolic capabilities. This transformation equips corporate entities with new communication, plasticity, and versatility capacities, which are primary factors behind the evolution of living organisms. The parallel can be illustrated by putting the extended evolutionary synthesis through ontological prisms (Figure 4.16):

- The intrinsic identity (#) of corporate entities is defined by their status, ownership, and collective values and history (genetics).
- Their functional identity (≈) is defined by their business models and processes (behaviors).
- Intrinsic and functional identities are embodied in corporate architectures that combine physical and symbolic assets.

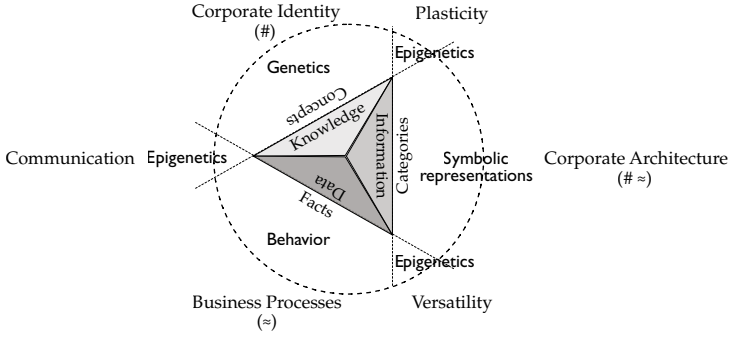


Figure 4.16 Framing human (inward) and corporate (outward) evolutionary models within ontological prisms.

The corporate equivalent of epigenetics operates through versatility, plasticity, and natural language communication:

- Versatility allows for continuous adjustments of business processes (behaviors) in response to changes in the environment. Over time, long-standing adjustments can be integrated into architectures.
- Plasticity allows for the continuous alignment of governance schemes and planned evolutions with emerging factors. Over time, forces arising from business environments can thus be integrated into corporate identity.
- Whether driven by business or social purposes, organizations use natural language to communicate with their ecosystems. Their success thus depends on their conceptual representations being permeated by the collective values and behaviors of their socio-cultural environment.

Assuming that adaptive changes learned through perception, experience, and projection can be woven into the collective knowledge embodied in corporate systems, organizations, and culture, such a learning capability would significantly enhance their sustainability and resilience.

4.4.3.3 Sustainability and Resilience Through Ontological Prisms

Like natural evolution, the evolution of organizations is not linear but rather the combined result of unexpected and planned changes, which may or may not be related. As long as onsets can be established within steady time frames, there is time for governing bodies to guide the path of evolution between changes with adaptive value and extraneous shifts. However, that long game of piecemeal and focused adjustments, while effective for plain-sailing sustainability, cannot unfold with systemic disruptions such as those that could be triggered by climate change.

Systemic disruptions can be characterized by a breakdown of the customary alignments between spatial and temporal dimensions. Given the time scale and scope of these disruptions, the current state of knowledge cannot provide guidance on how to cope with the unfolding events. Conversely, as framed by ontological prisms, sustainability means that courses of events (a) can be aligned with objectives (b) and causal chains (c) (Figure 4.17, left).

Systemic disruptions and the crumbling of alignments between spatial and temporal references render all references to objectives and causal chains irrelevant (Figure 4.17, center). Alternatively, ontological prisms can be leveraged to enhance resilience by using communication (d), versatility (e), and plasticity (f) to renew alignments between the course of events, objectives, and causal chains (Figure 4.17, right).

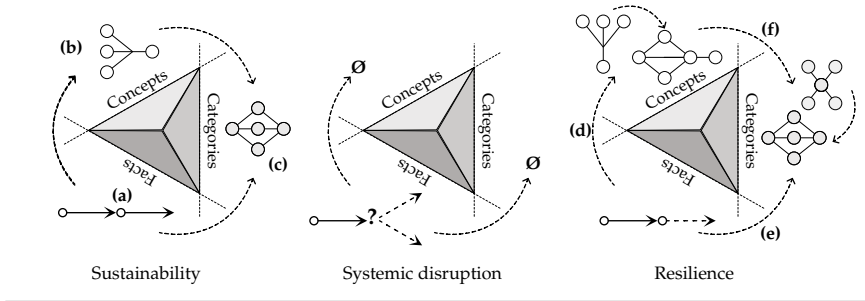


Figure 4.17 Sustainability and resilience.

The following chapters will revisit the issues of communication, causal chains, and objectives in relation to language, reasoning, and decision-making, respectively.

Key Points

Considering the pivotal role of governance for organizations immersed in digital environments, this chapter examines the benefits of ontological prisms in relation to four key dimensions: the institutional status of environments and enterprise architecture layers, value chains, regulatory compliance, and complexity management.

To begin with, exchanges with environments are categorized in relation to status, scope, and modalities of change: institutional, professional, corporate, social, and personal. Conversely, enterprise architectures are put into an ontological perspective with the traditional layers – platform, system, and organization – neatly aligned with the ontological dimensions of data (facts), information (categories), and knowledge (concepts), respectively.

Second, analyzing value chains through ontological prisms offers a dual benefit: it allows for the segmentation of value chains based on added value, and this added value can subsequently be characterized by contributions from the knowledge value chain. Furthermore, governance issues can be articulated in terms of transaction costs, which are used to compare the cost of resources – whether priced internally or by markets – to expected returns. These returns could be obtained directly, for example, through the sales of data products, or indirectly through their integration into marketed products or services.

Third, this chapter examines data protection and intellectual property regulations through ontological lenses. Regarding privacy, the distinction between

data and information emerges as critical, as it establishes a conceptual boundary between private and corporate realms: the former pertains to identified individuals, while the latter relates to categories of managed surrogates. In terms of copyrights, the benefits arise from representing creative works as singular, based on taxonomies, and nominal, based on thesauri. Finally, the representation of patents leverages the three ontological crossroads: thesauri, which differentiate between patents' actual and symbolic scope; domains, which distinguish between intent and patented content; and taxonomies, which separate patented content from its utilization.

Fourth, ontological prisms are used to assess governance in terms of complexity management. A distinction is first made between extensional (2D) and intensional (3D) complexity; the former addresses facts only, while the latter addresses both facts and objectives. Drawing from cybernetics, entropy is then introduced to characterize the objectives of complexity management.

The chapter concludes with a parallel between the sustainability and resilience of enterprises as viable entities in their ecosystem and the evolution of living organisms.

Part II

5

Ontologies and Language

The limits of my language mean the limits of my world

Ludwig Wittgenstein

Overview

Language has always been the underlying subtext of artificial intelligence (AI), whether it's for implementation (computer languages), communication (user interfaces), or truthfulness (knowledge representation). While technologies have long constrained AI advancements into distinct symbolic and nonsymbolic swimlanes, the harnessing of computing power and generative language models (GenLMs) has created a new and powerful momentum that blends communication and representation. The opening of this generative new frontier, accompanied by heightened, if not delusional, expectations, has also led to some confusion between language and knowledge. This confusion is further reinforced by the ubiquity of knowledge graphs (KGs), which are viewed indiscriminately as both content and containers. Ontological prisms provide a tool to sift through and separate language threads from the fabric of knowledge.

5.1 Anatomy of Languages

Until computers could serve as surrogate locutors, our understanding of language was expressed through language itself, which introduced some bias. Computers are logical devices that must be programmed, initially using ad-hoc formal languages. Soon, programming languages evolved to resemble natural languages, while still maintaining specific syntax and semantics. At first, these

advances have put the focus on the formal dimension of languages and on simulating natural language performance. Then the focus shifted from simulation to conversation in natural language, and now to cognizant conversation combining language and knowledge.

5.1.1 Communication vs Representation

From an evolutionary perspective, languages operate at two levels: direct (vocal or signed) and mediated (written) communication. The former relies on things being pointed to, ensuring that meanings can be directly attached to contexts, while the latter uses prior meanings that have been "canned" in shared representations (Figure 5.1).

Besides the unmediated association between conversations and contexts, direct communication can benefit from implicit logic and classifications. However, languages supporting mediated communication must be explicit about supporting logic and classifications.

When supporting mediated communication, languages operate like currencies, first as units of account (names), then as a medium of exchange (meanings), and finally as a store of value (shared representations). Functional integration is achieved through conversations (names and meanings), classifications (names and representations), and reasoning (meanings and representations).

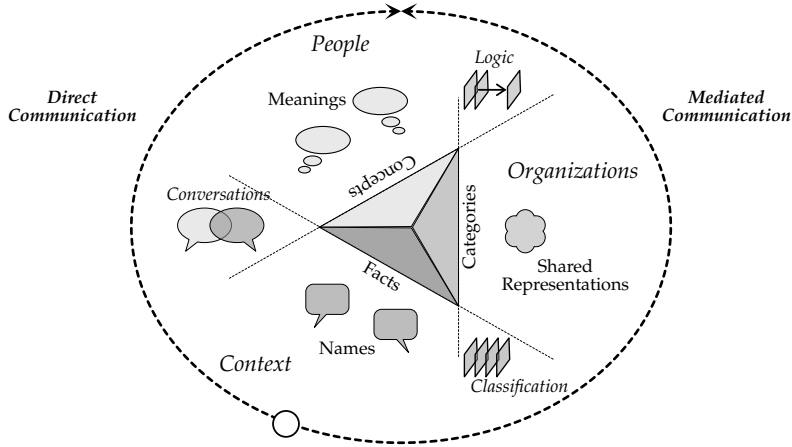


Figure 5.1 Language: communication and representation.

5.1.2 Communication: Signs and Symbols

The parallel between languages and currencies is more than metaphorical, as both serve as universal human mediums of exchange based on signs, symbols, and surrogates. Signs and symbols are used by living agents to communicate meanings, both directly in reference to actual contexts and indirectly through symbolic representations.

As defined by Charles Sanders Peirce, signs are percepts that represent something else in the minds of observers, while symbols are signs that represent common references (to something else) in the minds of observers. Borrowing from William of Ockham, one can further distinguish between first intention, which refers to signs that stand for objects or phenomena, and second intention, which refers to signs that stand for other signs. These distinctions can be neatly aligned with the areas defined by ontological prisms (Figure 5.2):

- First intention signs use actual icons, sounds, or gestures to mimic objects or phenomena; they exist as pure facts, i.e., without a symbolic dimension (a).
- Symbols are shared mental representations (words or pictographs) whose meaning is established independently of the signs representing them (b).
- Second intention signs have a dual nature, encompassing both actual and symbolic dimensions (c).

These tiers determine the functions and technologies of languages: both first and second intention signs enable direct communication through spoken or signed languages; however, representation, and consequently written communication, requires second intention signs. When second intention signs are used to represent things, their dual nature is activated, and they become surrogates, meaning they stand in for some actual counterpart. However, surrogates should

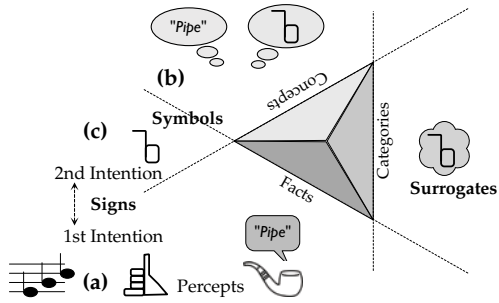


Figure 5.2 Signs and symbols.

not be confused with things or concepts; they are symbolic artifacts fully described by categories. For instance, when scribes in ancient Egypt inscribed the glyphs of cattle on clay tokens, these signs were simultaneously pointing to a category (symbols) and representing actual animals (surrogates).

5.1.3 Representation: Grammars and Semantics

Human language is universal, and yet human languages are arguably the most idiosyncratic of human abilities; on that account, the best approach is the functional distinction between communication and representation.

With regard to communication, the relevant distinction is between natural, symbolic, and digital languages, as these can be aligned with users as well as content: people (everything), information systems (everything symbolic or digital), and devices (everything digital).

With regard to representations, the primary distinction pertains to the nature and role of grammars. While there is little consensus about grammar, aside from the name itself, four tiers are commonly considered:

- Syntactic, for the structural and functional organization of terms independently of their meaning.
- Lexical, for the individual meaning of terms.
- Semantic, for the meaning of the syntactic constructs (aka phrases).
- Pragmatic, for the meaning of the semantic constructs (aka sentences) depending on contexts.

It must be noted that, contrary to communication levels, which pertain to observed linguistic performances, these tiers pertain to the representation of linguistic performances. Nevertheless, they can be mapped with communication levels and from there to prisms' thesauri, models, and ontologies. In addition to natural languages, they can even characterize symbolic languages as modeling, formal, and empirical, depending on their scope: systems, concepts, or facts, respectively. However, difficulties arise in distinguishing between models and ontologies, i.e., between language and knowledge (Figure 5.3).

At first, layered grammars, epitomized by Noam Chomsky's pioneering work, seemed to provide a reference model supporting a distinction between languages, defined by grammatical categories, and knowledge, whether symbolic, represented by KGs, or nonsymbolic, represented by neural networks. However, many linguists disagree with these so-called computational approaches to languages, to name a few:

- Semantic grammars, epitomized by Role and Reference Grammars (RRGs), address semantics through a combination of syntax and lexicon.

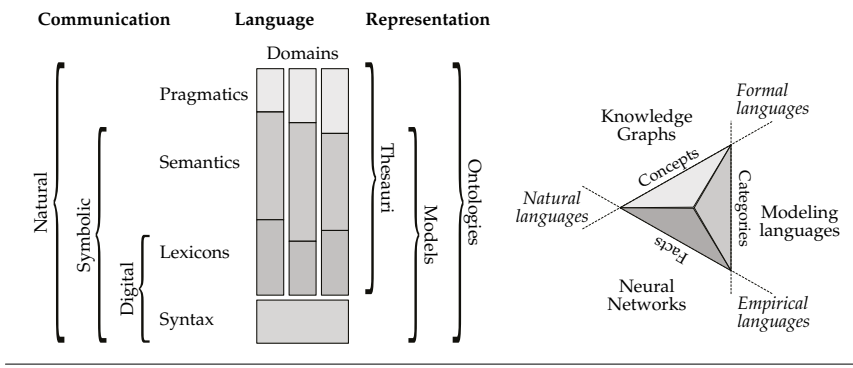


Figure 5.3 Languages and grammars.

- Functional grammars try to align semantic constructs with templates designed to represent states of affairs in possible worlds, such as situations, events, actions, and processes.
- Cognitive grammars, exemplified by George Lakoff, try to ground semantic constructs in deeply ingrained human metaphors.

As would be expected in debates about language, there is no consensus among grammatical approaches, which may explain the overwhelming revival of computational methods driven by machine learning technologies. In a radical departure from the Chomskian layered model, generative models are designed to learn languages without relying on prior linguistic categories.

5.2 Generative Language Models

GenLMs are computational systems that rely on deep learning technologies to combine the capabilities of parsers (which translate signs to symbols) and interpreters (which translate symbols to meanings). On the one hand, GenLMs can learn natural languages from scratch simply by being taught through statistical (as opposed to grammatical) parsing of a vast amount of multimodal (sounds, images, text, etc.) material. On the other hand, they can interpret and process queries and documents, and engage in conversations thereof.

5.2.1 A Functional Perspective

Considering GenLMs' functional architecture and without delving into the wide and rapidly changing array of technologies and algorithms involved, GenLMs rely on two sets of overlapping external and internal options (Figure 5.4).

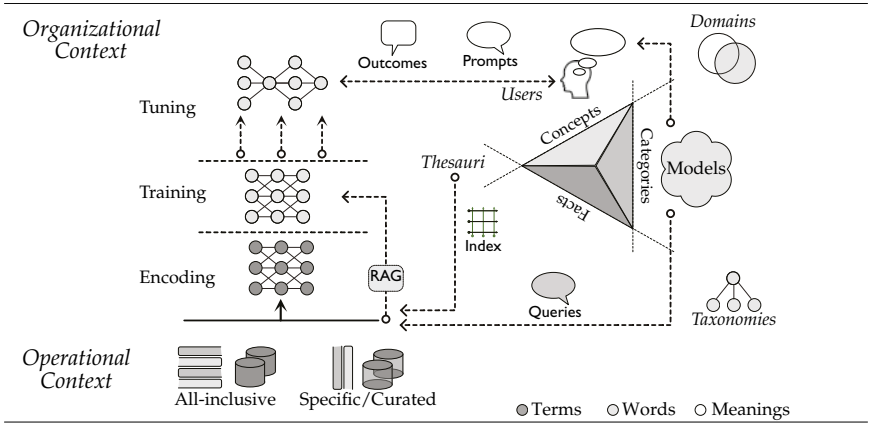


Figure 5.4 A functional view of GenLMs.

The set of external options pertains to training materials. Initially, aptly named large language models (LLMs) were built on the assumption that bigger is better. However, they have encountered the law of diminishing returns, which is manifested by exponential increases in computing power and frustratingly decreasing gains in relevance or accuracy, not to mention the significant input from skilled humans needed to design the queries and prompts. This has led to more diversified approaches that combine wide-ranging, selective, and curated training materials addressing all-inclusive, domain-specific, or tailored content, respectively.

Concomitantly, the set of internal options pertains to training and tuning modalities, including supervised and self-reinforcement learning, as well as encoding/decoding combinations. Beyond a puzzling assortment of deep learning algorithms, GenLMs weave together three kinds of processing:

- Statistics, for morphing sequences of tokens (signs) into sequences of words (symbols).
- Semantics, for the alignment of words with meanings.
- Pragmatics, for the alignment of meanings with users' contexts and knowledge sources.

Embeddings constitute the cornerstones of the models obtained, with each embedding representing the meaning of a word or a concept as a vector set in an n-dimensional space. The goal is to maximize relevance, taking into account the number of dimensions deemed necessary.

How embeddings are combined under the hood of GenLMs depends on the undertakings, typically: operations in bounded contexts (e.g., programming,

editing, help-desks), authoring in open contexts (e.g., writing), or searching in open contexts (e.g., documentation). Nevertheless, overall, exchanges between GenLMs and their environments appear to broadly follow the same configuration. On the side of operational contexts, a variety of commonly called retrieval-augmented generation (RAG) technologies rely on tailored queries to generate different types of training material: all-inclusive, specific, or curated. On the side of organizational (users) contexts, various "chain-of" scripting methods are proposed to define and organize prompts, usually paired with RAGs called dynamically to customize contexts. In between, embeddings provide both static and dynamic semantic gears (more on that below).

Such a functional view of GenLMs can be aligned with the organization of ontological prisms: queries, taxonomies, and models for operational contexts, prompts, and domains for organizational contexts. That alignment could enable the harnessing of GenLMs and ontologies, and thus overcome the limits of GenLMs.

5.2.2 Limits of Generative Language Models

5.2.2.1 Functional Limits of Generative Language Models

Setting aside performance considerations, hallucinations can be viewed as the major issue for LLMs. Set in an ontological perspective, hallucinations can stem from groundless factual references or nonsensical reasoning. Nonfactual hallucinations occur when the intentional modalities of queries (what the user has in mind) are not aligned with the extensional modalities of facts (what the model is considering): observed, asserted, assessed, deduced, or managed. Concomitantly, nonsensical hallucinations happen when the logic of outcomes cannot be supported by categories' aspects and causal chains. Different schemes can be used to reduce GenLMs' hallucinations, from dedicated training and external knowledge sources on the one hand, and fine-tuning with techniques like RAG or chain-of-verification (CoVe) on the other hand. But those approaches remain ad-hoc solutions that mostly rely on supplementing GenLMs with human expertise without addressing their inherent limits.

5.2.2.2 Inherent Limits of Pure Generative Language Models

Generic language models apply statistics to extensive collections of tokens in order to generate the emergence of meanings. To that end, training materials are first divided into chunks, with their contextual affinities encoded as vectors set through a large number of parameters and stored as embeddings. However, there is a catch: emergence phenomena require scale, involving training on trillions of tokens and billions of parameters, which leads to an exponential complexity of embedding lineages. Putting aside trust in the wisdom of crowds, two approaches have

been attempted to ensure some transparency: statistical and scripted. Statistical approaches rely on multilayered regression models to assign weights to causal chains. However, this does not address the original issue of 3D complexity lineages, not to mention the limitations of regression models when applied to nominal independent variables detached from actual environments. Scripted approaches use chains of prompting templates tailored to conversational purposes and grammatical constraints as a way to tutor the reasoning of models. But as they stand, such scripts can only achieve their objective by prompting language models, supposedly already well trained, into reproducing, step by step, human reasoning. While scripted methods may seem more convincing than the regression-based explainability of GenLMs, both are inefficient compared to traditional conversational access to information systems. The former is inefficient in terms of computing power, while the latter is inefficient in terms of human resources.

Differentiated inefficiencies in dealing with GenLMs' opacity may provide some clues to the root causes:

- GenLMs are designed to handle multimodal tokens rather than the facts themselves. When regression models and deep learning algorithms are intended to target facts rather than documents, they can be more effective if they bypass the nominal (textual) layer of language models and are applied directly to the data (see Chapter 7).
- The same rationale applies to symbolic reasoning. GenLM's nominal tokens can only support tutored propositional logic (as opposed to first-order logic and higher-order logics). Applying scripted causal chains built on higher-level logic to nominal arguments can only serve as trompe-l'oeil (see Chapter 6).
- Finally, opacity is not only an issue for causal chains, but it is also (especially?) an issue for accountability. Whatever the expectations about generative artificial intelligence (GenAI), autonomous decision-making is still beyond the reach of GenLMs (see Chapter 7).

Taking these inherent limits into account, attempts are being made to offload knowledge-based activities to cognizant agents, using embeddings and RAG to support collaboration.

5.2.3 Agents and Collaborative Architectures

Initial wild expectations about GenLMs' capabilities have been progressively tempered by two main factors: effectiveness and reliability. These factors, in turn, have highlighted the benefits of outsourcing knowledge-intensive tasks to external agents. The wide range of undertakings aimed at agentic collaboration raises a number of issues regarding outsourced tasks and agents' abilities, and answers are still fragmentary. The main issue is that allegedly cognizant agents remain as

opaque as the GenLMs they are meant to support, because – aside from their reliance on KGs – not much is said about agents' cognitive abilities.

5.2.3.1 Graph-enhanced Retrieval-augmented Generations

Behind the marketing buzz, agentic AI approaches refer to a set of nonspecific collaborative architectures capable of hosting agents with neuro-symbolic capabilities. Agentic graphs (AGs) can be viewed as representative of the agentic AI concept. Simply put, AGs double down on the static use of queries during training (Figure 5.5a) with a dynamic use during tuning, taking into account users' context (Figure 5.5b). By leveraging KGs built from existing models, domain-specific contexts can be structured and updated according to queries.

By harnessing KGs with RAG, AGs achieve a twofold objective: taking responsibility for incorporating explicit knowledge into graphs, and enabling dynamic management of domain-specific structured contexts. Beyond clear operational benefits, AGs have broader significance regarding the collaboration between language models and knowledge sources. First and foremost, introducing KGs as GenLMs' significant other debunks the delusional belief of knowledge emerging from a primal soup of tokens. Second, the benefits of structured dynamic contexts highlight, in contrast, the wasted time and human resources spent on scripting elaborate chains of prompts.

While graph-enhanced RAGs can significantly improve the relevance of data used by GenLMs, the contribution of KGs is limited by the resources used to build them: vector-based retrieval from (unstructured) datasets and documents, or searches from (structured) domain-specific models. Neither of these methods fully addresses the root causes of hallucinations, namely, unfounded facts and nonsensical reasoning.

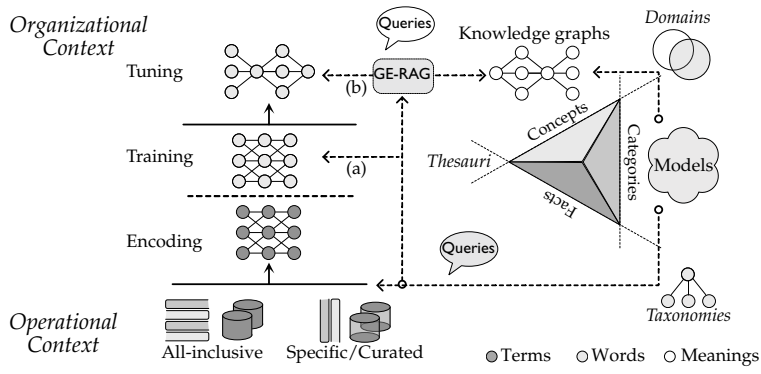


Figure 5.5 Graph-enhanced RAG: queries are used to train the language model (a) and build contexts (b).

5.2.3.2 Language Models and Ontological Prisms

Graph-enhanced RAGs' inability to deal properly with groundless facts and nonsensical reasoning is a direct consequence of their purpose: they are simply not supposed to address these issues. Nevertheless, graph-enhanced RAGs point in the right directions: regarding GenLMs, embeddings could do more than store pieces of knowledge; regarding KGs, embeddings could be built from ontologies instead of conceptual or relational models.

To begin with, standard GenLM methods could be used to embed ontological templates targeting syntactic constructs such as parthood and semantic contexts. Then, using ontological prisms as reference models, functional embedding would be tasked with collaboration with cognizant agents based on cognitive capabilities, typically observations, logic and reasoning, and judgment and accountability (Figure 5.6). These capabilities will be detailed in the next chapters.

Under GenLMs' hood, dedicated embeddings would operate on prompts like enzymes on substrates, first by applying syntactic embeddings to identify conceptual ones, and then using cognitive embeddings to outsource tailored knowledge-based tasks to ontological prisms.

5.2.3.3 Knowledge-driven Architectures

Beyond the benefits of pairing GenLMs with databases or more KGs systems, technology providers at large put their effort on distributed architectures that could support collaborations between specialized AI digital agents, architectures that are commonly known as agentic AI. In that context, the importance given to ontologies illustrates the pendulum swings between nonsymbolic and symbolic

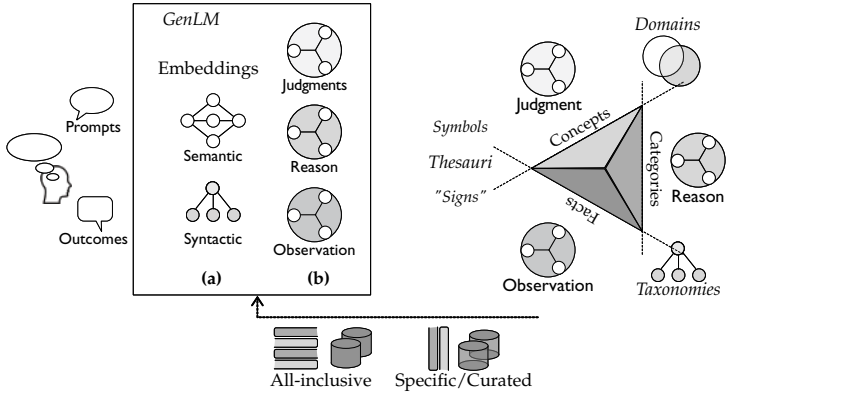


Figure 5.6 GenLMs with linguistic (a) and cognitive (b) embeddings.

approaches to AI, the former rooted in neural networks and the latter in KGs; the question is how to harmonize the swings. At ground level, it's how to offload knowledge-intensive activities to KGs, for instance through the Model Context Protocol (MCP). But KGs are containers, not agents, and assuming they are parts and parcels of agents, both will need a new kind of operating system. More generally, outsourcing knowledge-based activities points to a knowledge-driven collaborative architecture harnessing GenLMs for user experience and ontologies for knowledge sources. The question is, could this functional distribution be aligned with a broader one between language and knowledge? Ontological prisms put the issue through two dimensions, one for communication: conversational or contextual; the other for cognitive capabilities: observation, reasoning, and judgment.

5.3 Languages and Knowledge

As a discipline, GenAI can be compared to mythological chimeras – hybrid creatures made of organs and limbs transplanted from various animal species; in this case, from computer and cognitive sciences. The cultural and scientific backgrounds of GenAI experts often reflect this intrinsic duality, leading them to introduce symmetric biases toward either a symbolic or nonsymbolic approach. Interestingly, both biases share the same blind spot: the conceptual Gordian Knot that ties language and knowledge together, leading to a fruitless controversy over whether the cart or the ox should be in front, and of course, we know the answer: languages have been born to trade knowledge.

5.3.1 Direct and Mediated Communication

Language is all about communication, and communication arises from the need to share information, whether it's about a threat, an opportunity, or simply to tell a good story. Written languages have been developed as a means to liberate spoken languages from the constraints of immediate contextual references, thereby supporting the continuity and consistency of human organizations and endeavors – in other words, collective knowledge. For that purpose, conversational tokens – whether words or otherwise – must be associated with signs, either at the sound or sense level. This evolution can be illustrated by the paths taken by Occidental and Oriental language groups. Occidental languages that originated in the Middle East, such as ancient Egyptian hieroglyphs, began with figurative schemes but soon shifted to phonetic systems, breaking phrases into phonemes and using 22 basic alphabetical glyphs to represent

them. In contrast, Oriental languages have consistently followed a figurative path, embodying whole meanings in logographic glyphs and graphemes.

By not being based on phonemes, logographic representations are not tied to spoken expressions. As a corollary, as exemplified by Kanji, a single written logographic language can support different spoken languages. The significance of this capability can be illustrated by the postal service of the ancient Persian Empire, which provided a transcription service that used the Aramaic official cuneiform written language to represent the contents of exchanged messages. This enabled communication across the empire between senders and receivers who spoke different languages. The Persian Empire's transcription service stands as an operational real-life illustration of the ontological prism principle: meanings in one language can be transcribed into a second language and rewritten using a third language. However, it also illustrates the intractable distinction between language and knowledge: that distinction is either conflated with the distinction between direct (language only) and mediated (language and representation) communication or, as with semantic layers and the Language of Thought, the distinction is simply ignored. It may now take on a new relevance with the development of agentic architectures, under the guise of conversational semantics.

5.3.2 Language of Thought and the Computation Paradigm

Jerry Fodor's The Language of Thought (1975) can be viewed as representative of approaches weaving language and knowledge into the same fabric. In summary (our objective is not to present the Language of Thought models), Fodor considers mental representations as types that can be instantiated as cognitive events (typically intentions or beliefs), and which support propositional attitudes, which constitute the building blocks of logic. Thinking can thus be described as chains of mental events that instantiate mental representations. Language of thought models are further characterized by a dual reference to the computational theory of mind and the representational theory of thought, establishing a clear conceptual lineage between language of thought and GenLMs, namely the direct representation of cognitive events (symbolic instances of mental representations) by neural networks (the brain's nonsymbolic states) (Figure 5.7).

That direct association of symbolic and nonsymbolic tokens makes no room for a conceptual distinction between language and meanings: on the one hand (Figure 5.7, left), it's a matter of computation (GenLMs' training and tuning), on the other hand (Figure 5.7, right), the issue is ignored (the language of thought hypothesis doesn't take into account a shared symbolic representations of meanings).

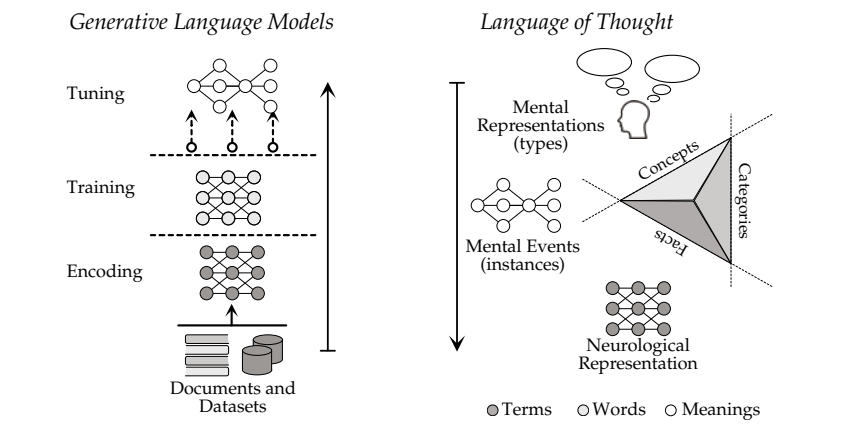


Figure 5.7 From neurons to thought: (left) generative language models, (right) language of thought.

5.3.3 Conversational vs Contextual Semantics

The unruly surge of GenLMs, on one hand, coupled with a more principled development of KGs, on the other hand, could suggest a specialization of agents between language and knowledge. Along that rationale, agents responsible for communication and user experience would rely on semantics and pragmatics to ensure the consistency of meanings, while agents in charge of shared representations would focus on knowledge-based activities. This assumes that semantics do not affect knowledge, which could occur at both ends of conversations, among users in actual (extensional) realms, and across sources in conceptual (intensional) realms. That assumption would be of little relevance for direct communications if they were limited to the actual here and now, which they usually are not. Hence, the question: how can we distinguish between extensional and intensional semantics in conversations?

5.3.3.1 Extensional Semantics

Initially, the so-called semantic layers were intended to function as an ad-hoc combination of thesauri and indexes, ensuring consistency in the meanings of user experiences across corporate resources. However, in the absence of an ontological perspective framing conceptual models, information systems, and master data management, semantic layers have become jack-of-all-trades, undertaking all ancillary activities associated with the variety of documents and data sources (databases, data warehouses, data lakes, etc.) (Figure 5.8).

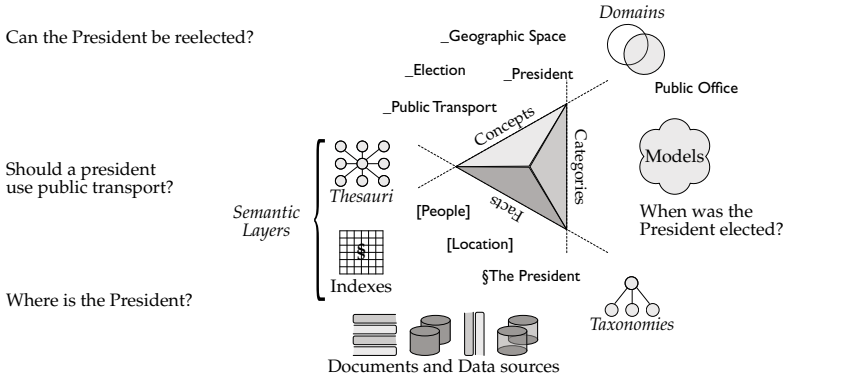


Figure 5.8 Extensional and intensional semantic realms.

Not surprisingly, most address semantic layers in terms of vocabulary, thesauri integration, and conceptual domains interoperability, with leading tool providers using KGs to extend the semantic capabilities of their user interfaces. Still, a number of startup companies go further, harnessing KGs and GenLMs in order to build knowledgeable frontends for corporate information systems. While advances are being made in both directions, developments are driven by the needs of business users accessing shared information assets from specific perspectives. When knowledge engineering issues are addressed, it is most often as an afterthought.

5.3.3.2 Intensional Semantics

Intensional semantics (see von Fintel and Heim, 2011) addresses the aspect of semantics dealing with the displacement of contexts, allowing conversations detached from actual (extensional) references. As such, intensional semantics can be viewed as a kind of metasemantics pertaining to grammar, as illustrated by the wide range of embodiments across languages. From an ontological perspective, intensional semantics may also characterize marches areas of known unknowns. For instance:

- Where is the president? When was the president elected?
- How many times can a president be reelected?
- Should a president use public transport?

These concerns could be addressed through RRGs by languages that combine syntactic and semantic constructs. However, aside from the specificity of such approaches, they would entrench the confusion between language and knowledge. Alternatively, intensional modalities could be addressed dynamically through conversations between agents, provided that extensional specifics

(what is known) are not entangled with intensional universals (who knows, how, why, when). This can be achieved by using ontological prisms to distinguish between contextual and conversational components.

5.3.3.3 Semantic Layers Revisited

For all intents and purposes, developments in agentic AI are being driven by the inherent limitations of GenLMs and the need for AI architectures that allow for the outsourcing of knowledge-intensive activities. However, such architectures must support collaboration between agents with different intelligent capabilities, utilizing different semantics to address heterogeneous contexts. Ontological prisms can be employed to frame this collaboration, provided that the exchanged messages can be characterized by the type of context (direct or mediated) and the type of cognitive capability (observation, reasoning, and judgment) involved.

As explained in Chapter 3, ontological prisms distinguish four semantic layers: two related to communication (empirical and nominal) and two related to knowledge representation (formal and deontic). Intensional semantics can thus be reframed in broader contextual terms:

- Empirical (extensional) when meanings can be directly observed from context (e.g., where is the president?).
- Nominal (circumstantial) when meanings are obtained from documents that can be directly accessed in the context (e.g., when was the president elected?, should a president use public transport?).
- Formal (logical) when meanings are obtained from shared representations detached from the current context (e.g., when was the president elected?).
- Deontic (intentional) when rules are involved (e.g., how many times can a president be reelected?).

These revisited semantic layers could thus be used to characterize collaboration between generative agents: conversational (nominal, empirical), contextual (formal, deontic), or both (Figure 5.9).

Concomitantly, agents could also be defined by their cognitive capabilities: observation, reasoning, and judgment. Cognitive and communication capabilities could then be combined to characterize access to knowledge:

- Conversational (direct communication) pertaining to extensional and intensional realms using natural and empirical languages.
- Contextual (mediated communication) pertaining to intensional and logical realms using natural and formal languages.

That framing would allow for the subcontracting of GenLMs for knowledge-intensive activities, outlining the upper layers of an agentic operating system. It could also help to untangle the Gordian Knot that binds language and knowledge.

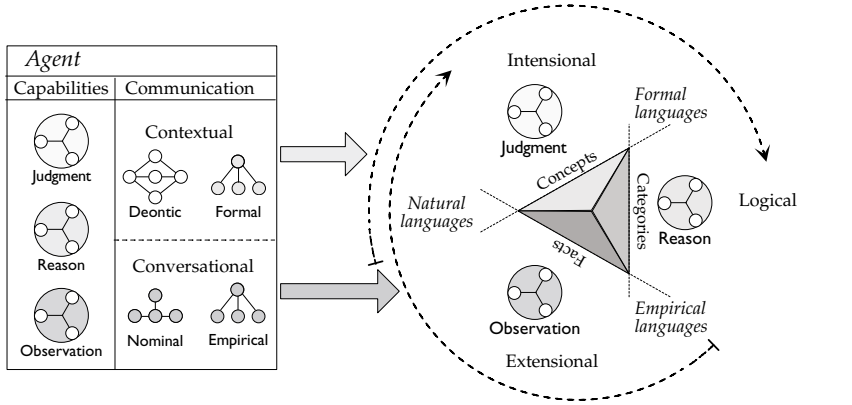


Figure 5.9 Framing agentic AI collaboration.

5.3.4 Language at the Edges of Knowledge

5.3.4.1 Meanings and Semantic Entropy

From philosophers to linguists, and from cognitive psychologists to computer scientists, advances in generative intelligence have put new light on the role of language in expanding the galaxies of knowledge: sciences progress on the wings of new concepts. On the language side, the unfolding stems from the overlap of conversational and contextual semantics, and develops in the semantic span between what is said and what is meant. On the knowledge side, that span is reflected by semantic entropy (see Chapter 4).

Semantic entropy reflects the discrepancies between the meanings of concepts used for values, policies, and objectives on one hand, and the meanings of terms used in documents, applications, or datasets. In terms of knowledge, these discrepancies can be viewed as the semantic imprints of flawed taxonomies of interest or misguided domains of concern (Figure 5.10).

Taxonomies constitute a primary source of meaning by establishing mappings between words and environments. To that end, each taxonomy introduces a semantic space (e.g., Diet[2]), which is used to frame the definition of dependent terms (e.g., religious, culinary, medical, etc.) (more about this issue in Chapter 9). Taken on their own, taxonomies should be effective: when applied to targeted individuals, they must yield clear and unambiguous answers (or mutually exclusive subsets). When set within ontologies, they must also serve domain-specific purposes.

In contrast to taxonomies, which operate bottom-up from terms, domains constitute a secondary source of meaning that operates top-down, defining semantic spaces from concepts. Taken on their own, domains must be consistent: when

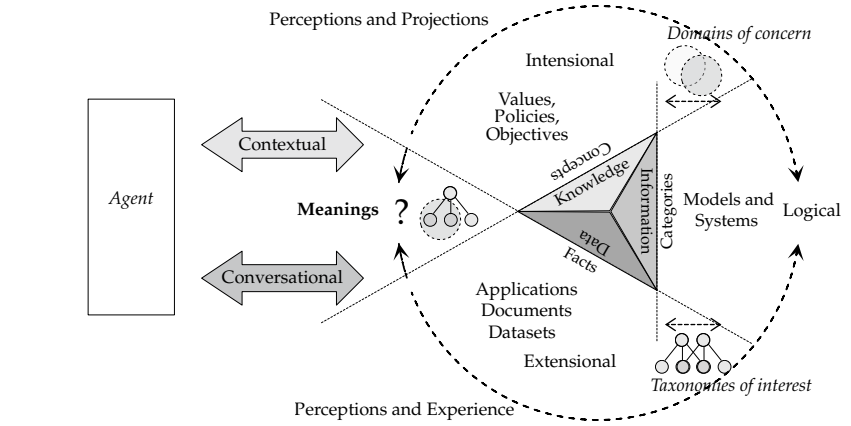


Figure 5.10 Meanings at the cross of semantic spaces.

they define semantic spaces for words, they must ensure clear and unambiguous meanings. When set within ontologies, they must also ensure the interoperability of their respective namespaces; their effectiveness, or lack thereof, is represented by conceptual entropy, specifically the gaps left by the intersection of meanings emerging from taxonomies and those defined by domains. Assuming that semantics are used to shift knowledge boundaries, one should expect new concepts to emerge from attempts to fill these gaps.

5.3.4.2 Languages and Emerging Concepts

So far, we have considered languages in relation to cognitive capabilities (observation, reasoning, judgment) and communication (contextual or conversational). A third perspective must be added to account for the role of language in knowledge extension. In this regard, the concepts of "machine learning" and "generative intelligence" seem particularly relevant, regardless of intent, because natural languages are, by their very nature, generative of concepts. The question is how emerging concepts can be related to machines.

While paths of discovery are, by definition, countless and infinite, worthwhile ones must eventually be anchored to concepts, whether expressed in sentences or represented by pictograms. Regardless of the limitations of GenLMs detailed above, multimodal language models (MLMs) and large concept models (LCMs) illustrate the pivotal role of concepts between language and knowledge.

By applying GenLM technologies uniformly to texts, sounds, and images, MLMs aim to blur the distinction between communication and representation, and more generally, to bypass languages altogether. Ultimately, what remains are memes and concepts embedded as vectors.

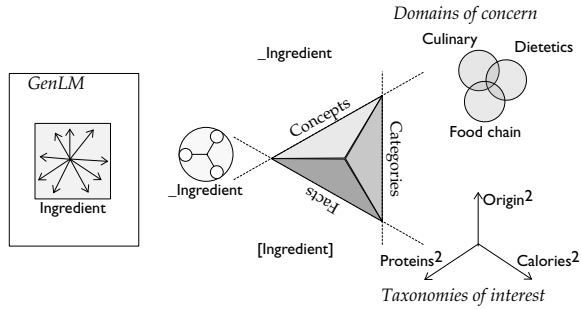


Figure 5.11 The matter of concepts.

In contrast to the all-inclusive approach of MLMs, LCMs remain focused on textual tokens but operate on sentence embeddings instead of individual terms. The aim of LCMs is to capture meanings at a higher (conceptual) level in order to support language-agnostic reasoning. In this regard, LCM sentences can be compared to the logographic glyphs of oriental languages such as Kanji, or to Aramaic cuneiform writing used by the postal service of the ancient Persian Empire mentioned above.

Our point here is that, one way or another, generative approaches to language converge in their attempts to capture concepts that could be embedded as vectors and used to access knowledge. At the same time, most have also concluded that language models alone will never achieve this kind of literacy and will require the assistance of cognizant agents and ontologies. What is missing is the agentic AI middleware supporting the collaborative architecture. Enter ontological prisms.

When knowledge is managed through ontological prisms, concepts are defined by taxonomies and domain-specific semantic spaces, and can thus also be represented as semantic vectors, which can be mapped to corresponding embeddings (Figure 5.11).

Concepts can thus provide the critical hinge between GenLMs and knowledge sources, provided these sources are managed through ontological prisms. The following chapters will shift the focus to the benefits of reasoning and decision-making.

Key Points

Part one has described the structure of ontological prisms, while part two focuses on their functions, starting with language.

Our primary objective has been to clarify the fundamental traits of language: signs versus symbols, communication versus representation, direct versus mediated communication, and grammar versus semantics. Ontological prisms have been used to organize these traits and their relationships.

Next, we have employed that conceptual grid to introduce a functional approach to GenLMs, which are arguably a defining advance in AI technologies. Our focus has been on two main issues: the inherent limitations of GenLMs, which are designed to process predefined words (tokens) detached from reality and cannot be expected to go beyond nominal reasoning; and the need for GenLMs to collaborate with cogent and cognizant agents to whom they can delegate knowledge-intensive tasks. We concluded that both issues highlight the benefits of a knowledge-based architecture with ontological prisms at its core.

The last segment has addressed the role of language in knowledge-based architectures. Returning to the distinction between direct and mediated communication, we have analyzed semantics through ontological prisms, redefining layers in terms of conversational and contextual semantics.

6

Reasoning with Ontological Prisms

> *There's no sense in being precise when you don't even know what you're talking about*
>
> John von Neumann

Reason can be viewed from two perspectives: one pertains to the suspension of time, allowing for decisions to be made (which will be discussed in Chapter 7), while the other involves preventing every truth from being stated all at once. But how could that be?

Philosophers suggest that there are multiple truths depending on various contexts, such as conceptual, pragmatic, linguistic, semantic, normative, and so on. Ontological prisms are more pragmatic, as they only consider truths that can be rooted in definite realms: intensional (mental representations), extensional (empirical observations), or logical (collective or shared representations). However, since truths can only be expressed through language (see Chapter 5), the interoperability of observations and representations (see Chapter 3) must be revisited to support reasoning across heterogeneous realms. First, assuming that reasoning is meant to be explicit, we will explore the ways reasoning can be expressed through ontological prisms. Based on that, we will address the corresponding reasoning options: formal (concepts/categories), nominal (facts/concepts), and empirical (facts/categories). Finally, we will consider reasoning from a broader knowledge perspective.

6.1 Ontologies and Reason

6.1.1 The Scope of Reason

When viewed through ontological prisms, truths are grounded in facts, concepts, and categories. "Grounded" means that corresponding truths relate to these realms; it does not imply that facts, concepts, or categories can be classified as

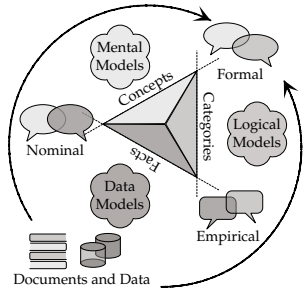


Figure 6.1 Nominal (clockwise) and empirical (counterclockwise) reasoning paths.

true or false on their own. The most that can be said is that they are consistently represented. Ontological prisms go further by enabling transverse reasoning across different understandings of truth: empirical (facts), deontic (concepts), and logical (categories).

To ensure the interoperability of reasoning across heterogeneous realms, ontological prisms rely on three mechanisms: thesauri for aligning meanings between data (facts) and mental (concepts) models, taxonomies for aligning partitions in data and logical models, and domains for aligning responsibilities between mental (concepts) and logical (categories) models. Moreover, to maintain the consistency of reasoning across realms, ontological prisms utilize aspects and syntactic and semantic connectors. Aspects and syntactic connectors are semantically neutral and can be used uniformly across realms, while semantic connectors are specific to individual realms (see Chapter 2) (Figure 6.1).

Reasoning with ontological prisms can thus be carried out in both clockwise and counterclockwise directions, crossing concepts and documents (nominal reasoning), concepts and categories (formal reasoning), as well as categories and datasets (empirical reasoning).

6.1.2 Reasoning Paths

As reasoning paths traverse the realms of ontological prisms, they can rely on three kinds of logic: nominal logic, which operates on terms; formal logic, which operates on symbolic representations; and empirical logic, which operates on data (Figure 6.2).

To that end, reasoning can make use of resources from extensional realms:

- Documents, taken as nominal default inputs, are made of elementary meaningful units (EMUs), textual or otherwise.
- Databases (managed facts) and datasets (taken as empirical default inputs), referenced through directories, dictionaries, or catalogs.

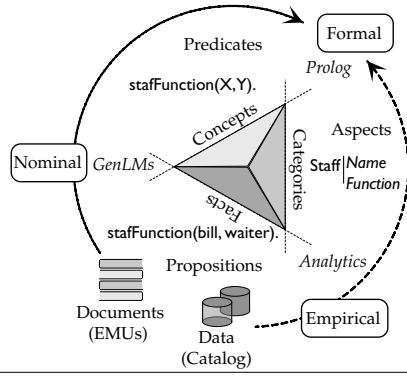


Figure 6.2 Overview of reasoning paths across ontological prisms.

Besides nominal logic, which operates solely on literals, reasoning with ontological prisms relies on propositions and predicates:

- Pure propositional logic, which operates on terms (no variables) and makes no difference between EMUs and logical operators, and its classic variant, which uses thesauri to organize terms around concepts.
- Predicate (or first degree) logic operates on concepts. Along with operators and syntactic constructs, predicates bind concept semantics to variables and may be combined with categories (logical models).

Both propositions and predicates apply to aspects, allowing the alignment of empirical reasoning (data analytics) and formal (predicates) reasoning. Generative language models (GenLMs), data analytics, and the Prolog language are used to illustrate nominal, empirical, and formal reasoning, respectively. The Common Logic Interchange Format (CLIF) is meant to serve as a reference model.

6.2 Formal Reasoning

6.2.1 Reason and Logic

For all intents and purposes, reason is the way humans organize symbolic representations, using logic to that end. While there can be as many logics as there are symbolic languages, ontological prisms aim for a pragmatic approach focused on nominal, propositional, and predicate (or first-order) logic.

6.2.1.1 Nominal Logic

Nominal logic operates on terms: if "A" is true and "A" implies "B," then "B" is true. Since terms are signs, their intrinsic truth value is irrelevant. For instance,

if it is stated that "Blue is a color," "The sky is blue," and "This hat is blue," then "This hat has the color of the sky" is true regardless of the meaning of "blue." From that point, assuming that sentences are reduced to EMUs, two approaches can be considered: propositional (or sentential) logic operates directly on EMUs to form logical relationships and properties; alternatively, formal logic introduces syntactic operators and constructs to serve as logical glue between EMUs. Mixed approaches like truth-functional propositional logic are bridging the gap.

The relevance of pure nominal reasoning rests on two pillars: one theoretical and the other technical. On one hand, semantic and functional grammars blur the line between syntax and semantics, providing a theoretical space for nominal reasoning where the truth value of terms can vary depending on their role in sentences. On the other hand, deep learning technologies can be used to build GenLMs that take advantage of that space to exhibit nominal reasoning capabilities (see Chapter 5).

6.2.1.2 Propositions and Predicates

In contrast to pure propositional logic, which makes no distinction between meaningful elements (sentences or statements) and syntactic glue (logical constructs), predicate (or first-order) logic relies on explicit grammars to organize EMUs into Truth Maintenance Systems (TMS). Nevertheless, both propositional and predicate logic depend on logical connectives such as "and" and "or," as well as the rules that determine the truth values of the propositions they are used to join. Prolog, an inherently declarative language with a long-established track record as a logic engine, can be used to illustrate the continuum between propositional and predicate logic.

Using propositions, the simplistic example above would be expressed with one proposition ("isTrue") and three assertions (Prolog terms start with lower-case letters, variables with capital ones):

- isTrue("Blue is a color").
- isTrue("The sky is blue").
- isTrue("This hat is blue").

and one rule (:-):

- isTrue("Blue is a color"), isTrue("The sky is blue"), isTrue("This hat is blue"):- isTrue("This hat has the color of the sky").

Predicates would allow for key distinctions:

- For syntactic (isa) and semantic (color) constructs.
- For type and instance: *isa(color, "blue")*.
- For feature and value: *color(sky, "blue")*.
- For variables in rules: *color(hat, X):- color(sky,X)*.

That polyvalence would enable an ontological continuity between reasoning with facts (propositions) and concepts (predicates).

6.2.1.3 Predicates and Categories

As noted in Chapter 3, the ontological alignment of heterogeneous representations transcends lexical, syntactic, and semantic distinctions. In the simple Prolog example above, while "isa" is unambiguously introduced as a user-defined logical (syntactic) operator, "color" occupies a syntactic position but could also be represented as an aspect (lexical) or as a concept (semantic). Similarly, "sky" and "hat" are represented as aspects (lexical) but could also be represented as concepts (semantic). That same issue of singling out concepts from sentences also arises with GenLMs, as noted at the end of Chapter 5.

When addressed within ontological prisms, singling out concepts from sentences using categories is straightforward if predicates can mirror types and aspects. Assuming a category representing staff members (see Chapter 2), facts corresponding to category instances (surrogates) could be represented as propositions (or assertions) (Figure 6.3):

- englishLevel(bill, 'Poor').
- englishLevel(sara, 'Good').
- stafFunction(bill, waiter).
- stafFunction(sara, headWaiter).

Rules regarding the required English level for staff functions would then be represented by predicates, for example:

- stafFunction(X, headWaiter): englishLevel(X, 'Good').

A query like *stafFunction(bill, headWaiter)* will return *false*, and one like *stafFunction(X, headWaiter)* will return *X=sara*.

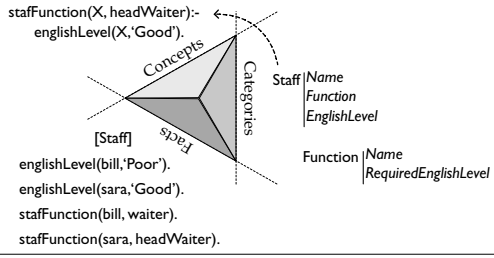


Figure 6.3 Building a truth maintenance system from categories (Prolog terms start with lower-case letters, variables with capital ones).

It should be noted that the mapping of categories to concepts cannot be reversed because terms in predicates can represent individuals, e.g., English Level(sara, 'Good'), as well as types, e.g., languageLevel(English, 'Good'). More importantly, the issue is not a programming one but an ontological one, e.g., depending on context, "English" can represent an item (term) in a list of courses, or a course (type) with multiple occurrences (instances). This brings us back to the process of singling out concepts from sentences, and more generally, to the Aristotelian distinction between concepts and nominal things, i.e., things without substance (see Chapter 2). We will revisit this issue below in the context of nominal reasoning.

6.2.2 Explanation and Comprehension

The systemic opacity of generative artificial intelligence (GenAI) algorithms is undeniably a significant barrier to their widespread use, whether for regulatory compliance, security concerns, due diligence, or all of the above. This issue can be addressed on various levels. With regard to decision-making, the focus is on transparency and traceability (see Chapter 7). With regard to reasoning, the focus is on explainability.

Transparency is the answer to a priori opacity, as it relates to what can be known in advance about contexts. Conversely, traceability addresses a posteriori opacity, as it pertains to the sources (facts/data), motives (concepts/knowledge), and arguments (categories/information) that have contributed to the outcome. Explainability encompasses the entire process, integrating correlations (extensional realm), causations (logical realm), and interpretations (intensional realm) (Figure 6.4).

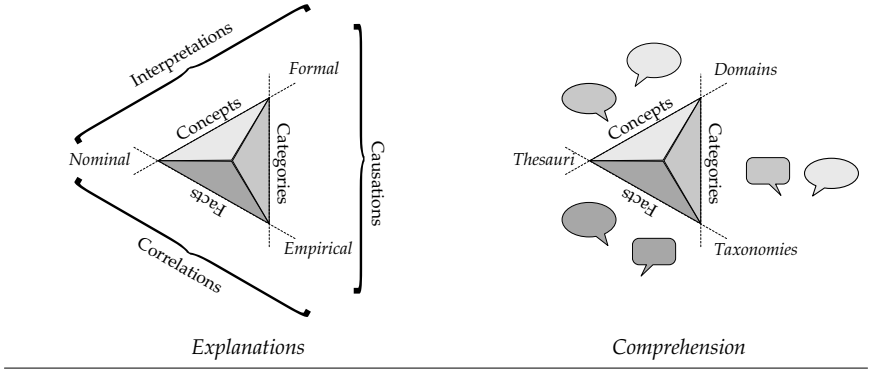


Figure 6.4 Explanations and comprehension.

Comprehension can thus be defined as an agent's holistic experience of explanations.

6.2.2.1 Correlations

Correlations consider past outcomes through the rearview mirror of datasets, using thesauri (naming) and taxonomies (partitions) to make sense of the reasoning process. However, correlations by themselves can be misleading because they rely solely on observations, independent of causes and effects. The contribution of correlations to understanding lies in a balancing act between statistical and nominal inferences: the former is used for aligning reasoning steps with prior knowledge (a kind of Bayesian depiction of a reasoning situation), while the latter is used for aligning steps and subsequent prompts and queries (Figure 6.5).

6.2.2.2 Causations

Causations rely on empirical and formal reasoning to build explanatory representations (or models) of facts according to modalities: observed, assessed, projected, etc. Empirical reasoning applies descriptive and inferential statistics to data to identify possible causes and effects and to draw hypothetical causal chains (Figure 6.5c). Inferential statistics are also used to build generative models that combine factual (Figure 6.5b) and counterfactual (Figure 6.5d) reasoning. Formal reasoning then uses predicates (Figure 6.5e) and clauses (Figure 6.5f) to devise sound pathways between the present state of affairs and conjectured horizons.

6.2.2.3 Interpretations

Interpretations belong in the minds of the beholders and can operate across three tiers: a single mind harboring its own objectives and interpretations, cooperating minds sharing objectives and interpretations, and adversarial minds

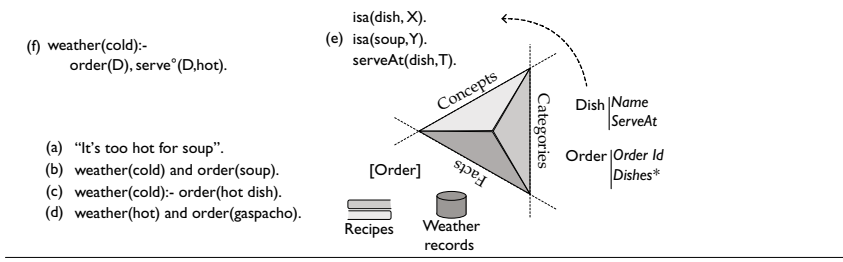


Figure 6.5 (a) Sentence, (b) correlation, (c) propositional logic, (d) counterfactual, (e) predicates, and (f) clause.

with concealed objectives and interpretations. With ontological prisms, these tiers can be concurrently represented and, therefore, operated as mental representations (see Chapter 2). Moreover, they can be reasoned about differently depending on the tiers: on the nominal side, footprints are used to match words according to contexts (interpretation); on the formal side, interpretations are aligned with causations through predicate logic (one-sided interpretations), fuzzy logic (collaborative interpretations), or game theory (adversarial interpretations). For example, in a marketing research study for a food packaging company, segmentation could take into account cultural experiences with spices, as well as the interpretation of "spicy," in order to design products and tune campaigns accordingly.

6.2.2.4 Explanation vs Comprehension

Explanation is selective as it weaves together the threads of correlation, causation, and interpretation into explicit designs. In contrast, comprehension is holistic, blending designs into persuasion. While explanations can be generated through artificial intelligence, comprehension is the human cognitive ability to unify empirical, nominal, and formal reasoning into an all-encompassing and consistent understanding. As such, comprehension spans two critical divides: first, between explicit and intuitive reasoning; and second, between individual and collective intelligence. Given that the aim of ontological prisms is to expose the fabric of automated (artificial?) reasoning, the approach appears to have a limited relevance for human cognition, in which comprehension comes first and may or may not be followed by purported explanations. While this may place comprehension beyond the scope of that book, the issue holds some relevance concerning the links between language and logic.

6.3 Language Models and Logic

6.3.1 Semantic Fault Lines and Fuzzy Reasoning

Chapter 3 introduced the dual issue of semantic layers and explainability in terms of the interoperability of heterogeneous ontologies, defining semantic layers accordingly: nominal, empirical, formal, and deontic. Alternatively, conceptual ontologies (i.e., those without explicit representation of facts and categories) can focus on the semantic issue and rely on extended thesauri or lightweight ontologies, such as Simple Knowledge Organization System (SKOS), to manage shared vocabularies. However, SKOS on its own does not have built-in reasoning capabilities. Language concept models (LCMs) represent a different approach, blending language and logic, at least at the nominal level.

6.3.1.1 Lightweight Ontologies

Conceptual ontologies are extended conceptual models implemented by knowledge graphs (KGs), making room for a generation of predicates or any other kind of logical constructs. Typical approaches rely on SKOS to build lightweight ontologies from data models that combine the functionalities of thesauri and taxonomies, allowing for a KG representation of vocabularies shared between domains. Since SKOS, like Web Ontology Language (OWL), is a declarative language implemented with Resource Description Framework (RDF), constraints and rules can be added to support reasoning; however, with two major caveats. On a practical level, SKOS remains a programming language, which means that everything must be programmed, especially the reasoning. On the ontological level, SKOS makes no semantic distinction between thesauri and taxonomies, i.e., between the semantics of conceptual and data models, respectively. Taking, for instance, market research about customer satisfaction, SKOS will make no difference between the business concept and the analytical variables used to ground it in facts. Interestingly, that semantic fault line is explicitly addressed by LCMs.

6.3.1.2 Language Concept Models and Fuzzy Reasoning

By rote or parrot, GenLMs are great learners that can be trained to extract meanings from wide-ranging, tailored, or curated materials (see Chapter 5). As long as nominal reasoning is concerned, meanings are not an issue because logic applies only to literals. In order to advance on the reasoning ladder, GenLMs must address the alignment of semantic anchors, specifically the role of concepts in sentences. To that end, the aim of what is commonly known as LCMs is to progress beyond tokens up the implicit syntactic category ladder, using sentences to produce embedded concepts (Figure 6.6).

A practical benefit of LCMs is that they can serve as bridges to modeling methods such as Entity–Relationship (Figure 6.6b) or Unified Modeling Language. When the corresponding models are implemented as KGs or RDF directly, ontological prisms can be used to attach predicates to concepts, paving the way to further reasoning (Figure 6.6d). However, the reliability of such reasoning is dependent on the relevance of the concepts defined by the sentences.

As GenLMs do in general for terms, LCMs use embeddings to represent sentences as vectors in semantic spaces. The soundness of embedded sentences, and consequently the relevance of associated concepts, depends on the dimensions that characterize the embedding spaces used to train LCMs. These dimensions are defined through billions of parameters, which are themselves defined at the token (sign) level. Augmenting these dimensions introduces conflicting training and tuning policies. For training, increasing the dimensions of semantic spaces will smooth out domain-specific semantics. For tuning, extending semantic spaces will undermine the benefits of semantic contexts (see Chapter 5). Either way, it

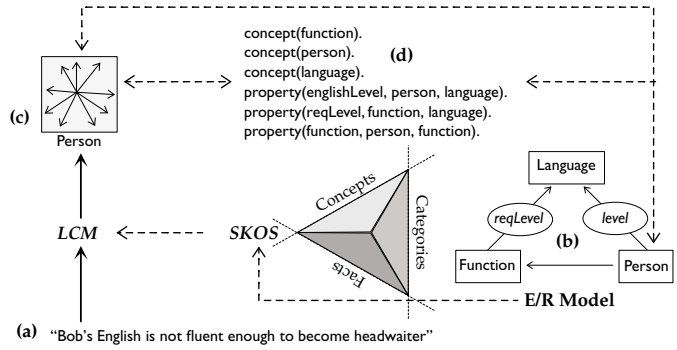


Figure 6.6 Models and predicates: (a) sentence, (b) model, (c) embedded concept, and (d) predicates.

means that the benefits that can be achieved from the dimension of sentences' semantic spaces are limited; it follows that the concepts derived from these sentences are inherently fuzzy, as will be the reasoning they would support.

6.3.2 Hallucinations and Nominal Reasoning

6.3.2.1 Hallucinations

The swift and widespread generalization of GenLMs (see Chapter 5) in natural language processing has highlighted the functional divide between communication and reasoning, and more generally, the foundational AI issue of language versus knowledge. On the one hand, GenLMs' impressive performance in processing documents has led to wild expectations, massive investments, and continuous if circumscribed improvements. On the other hand, attempts to extend their reasoning capabilities appear to stumble on intrinsic limits materialized by what is commonly described as hallucinations.

Hallucinations, broadly defined as factually incorrect or nonsensical responses, can be seen as inherent to GenLMs because such models operate on tokens that are disengaged from their original contexts. As seen in Chapter 5, various schemes have been developed to reduce hallucinations, in particular with dynamic contexts (e.g., graph-extended retrieval-augmented generation) and collaborative approaches between GenLMs and more cognizant agents tasked with knowledge-intensive activities. However, these approaches essentially shift the issues rather than solve them. First, they describe contexts and AI agents' behavior without delving into their realization, except for the customary mention of KGs. Second, the specificity of agentic AI communication protocols is simply overlooked. Nevertheless, as seen before, these issues can be effectively addressed

through a two-pronged distinction between contextual and conversational communication on one hand, and agents' cognitive capabilities (observation, reasoning, judgment) on the other. Based on these distinctions, an agentic AI middleware could set apart a conversational layer supporting nominal reasoning detached from explicit references to facts, concepts, or categories. Nonfactual and nonsensical hallucination issues would be addressed separately through a contextual layer by dedicated agents (Figure 6.7).

Using ontological prisms to check factual references and, consequently, to correct the nonfactual hallucinations of GenLMs can be straightforward. This is because the representation of facts is explicit regarding epistemic (concrete, abstract, virtual, nominal) and logical (observed, assessed, asserted, deduced, managed) modalities, as well as for ontological references across heterogeneous realms of concepts and categories.

Addressing GenLMs' nonsensical hallucinations is more challenging because, unlike fact-checking, determining what makes sense and what doesn't typically cannot be accomplished at the endpoint. Meanings are more like notes in a piece of music; they can only be perceived when performed as a whole, which is not possible due to the inherent opacity of GenLMs.

6.3.2.2 Nominal Embeddings and Vector Semantics

From a semiotics perspective, hallucinations can be explained by the confusion between signs (e.g., "beverage") and symbols (e.g., beverage), as mentioned earlier with the example of philosophy101, which can represent an item in a list of courses (sign) or a course (symbol) to be instantiated. In terms of language models, the issue is expressed through vector semantics.

Language models represent meanings as vectors in high-dimensional numerical spaces, assuming that the meaning of words and phrases is brought about by their

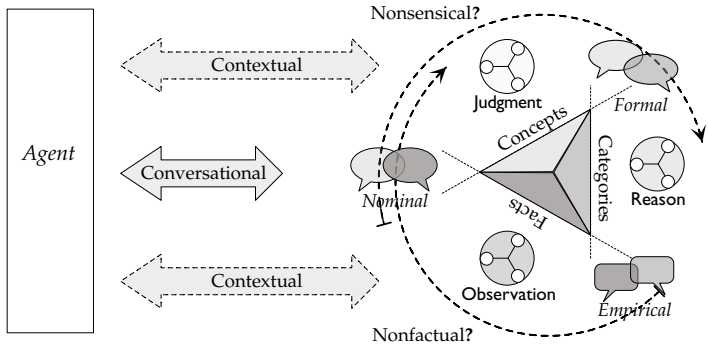


Figure 6.7 Addressing hallucinations through ontological prisms.

relationships with other words and phrases, with semantics acting like gravity between galaxies, stars, and planets. That assumption has two inherent caveats. On one hand, vector semantics are prone to local optimization; on the other hand, they completely ignore the attachment of meanings to reality. Both caveats are evident when vector translation is considered.

Simply put, a translation is a geometric transformation that moves every point of a figure, shape, or space by the same distance in a given direction, wherein the image is congruent to its preimage. Applied to meanings, congruency could pertain to:

- Nominal spaces, within which signs (tokens) keep their meaning regardless of attachments to environments.
- Formal spaces, within which symbols of categories keep their meaning regardless of attachment to environments.
- Empiric spaces, within which symbols of categories defined in bounded semantic contexts keep their meaning across selected environments.
- Ontological spaces, within which symbols of categories defined in unbounded contexts keep their meaning regardless of environments.

Given growing evidence of language models being limited to nominal translation, a second-best objective could be to represent nominal footprints as embeddings that could serve as patches or causeways to make up for lapses in the semantic layer (see Chapter 5). As nominal substitutes of concepts, they could:

- Circumscribe semantic ambiguities to small sets of clear-cut definitions meant to be either accepted or rejected as they are.
- Serve as firebreaks in dictionaries and thesauri, preventing circular definitions and reasoning.
- Facilitate translations between nominal expressions.

Nominal footprints would use conceptual connectors between nominal nodes; taking the example of beverages, combined with thesauri for synonyms, they could help to reconcile the meanings of labels on facts with the meanings of concepts (Figure 6.8).

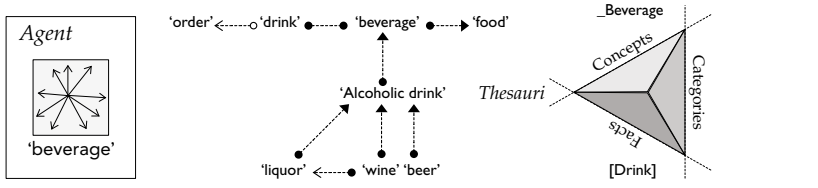


Figure 6.8 An example of nominal footprint (notations in Annex).

It should be noted that such nominal footprints are labels, not qualifiers, and that their scope (or the scope of any other set built on the same principle) is by nature limited to thesauri.

6.4 Empirical Reasoning

First, we have taken for granted that reasoning is symbolic and explicit. Then come general language models, which introduce a different kind of reasoning that operates on signs rather than symbols and remains mostly Delphic. Empirical reasoning takes a broader view, as it uses data analytics to establish symbolic representations of facts.

6.4.1 Data Analytics

Back to Aristotle, the fabric of reason is made of two threads, substance and accidents. Substance threads use thesauri to define the meanings of concepts, while accident threads use partitions to sort out distinguishing aspects, i.e., the variables that would bear out causes and effects. However, there is a catch: empirical reasoning begins with the tagging of partitioned data, rather than with established dictionaries. While data analytics methods and tools, a subset of data science, are intended to bridge that gap, they only go halfway, stopping at data catalogs that address documents and datasets. The question is how to proceed with empirical reasoning without the conceptual guardrails provided by Thesauri.

6.4.1.1 Reality and Causal Chains

Problems cannot be solved without untangling causal chains; for complex issues, this requires a comprehensive, consistent, and effective representation of causes, effects, and options. Analytic methods and tools, whether or not they incorporate artificial intelligence technologies, operate on data observations to sort causes and effects, fostering an understanding of current situations and defining valuable interventions.

Taking our food marketing example, data analytics would try to explain observed variables such as orders (outcome) in terms of weather (explanatory, exogenous), and menus and dishes (explanatory, endogenous) (Figure 6.9).

Typically, analytics will use directed acyclic graphs (DAGs) to represent the causal relationships between explanatory variables (such as weather, menus, and dishes) and outcome variables (such as orders). Framed through ontological prisms, the objective is to untangle three types of causal chains: extensional, nominal, and intentional. Extensional (or empirical) causal chains pertain to past events; they are the ones meant to be uncovered. Nominal chains are the

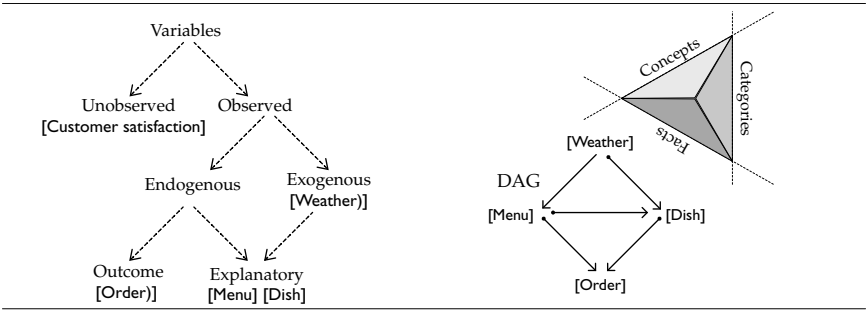


Figure 6.9 Directed acyclic graphs.

hypothetical ones currently expressed verbally, before semantic alignments. Intentional chains frame hypothetical ones with regard to aimed at (future) alignments; they are the ones meant to be explored. By leveraging this three-pronged perspective, ontological prisms can help untangle these chains, allowing for a more specific approach.

First, categories can be used to distinguish managed variables, such as menus and dishes' calories and prices, from outcome and exogenous ones, orders and weather, respectively. Additionally, nominal and conceptual placeholders can be added for unobserved explanatory variables, such as customer satisfaction (a virtual objective) and labeling regulations (a nominal constraint on menu and dish categories). DAGs can thus be revisited and interpreted according to the reasoning modalities introduced above (Figure 6.10):

- Extensional (or empirical) reasoning relies on qualitative approaches, such as regression analysis, to align facts with categories. For example, it examines the impact of weather on dishes, as well as the influence of dish prices and quality on sales (a).
- Nominal reasoning relies on qualitative approaches to assess nominal (unobserved) facts, such as customer satisfaction (b), based on observed facts, such as orders.
- Intentional reasoning relies on projective approaches to devise paths between present and future configurations, such as compliance with regulations (c).

That three-pronged understanding is key to overcoming the main challenge of GenAI algorithms, namely their explainability.

6.4.1.2 Shared Vocabularies and Aspects

The aim of data analytics is to make sense of unstructured data, which, in ontological (or Aristotelian) terms, corresponds to orphan accidents and, in marketing terms, to observations still unattached to meaningful business

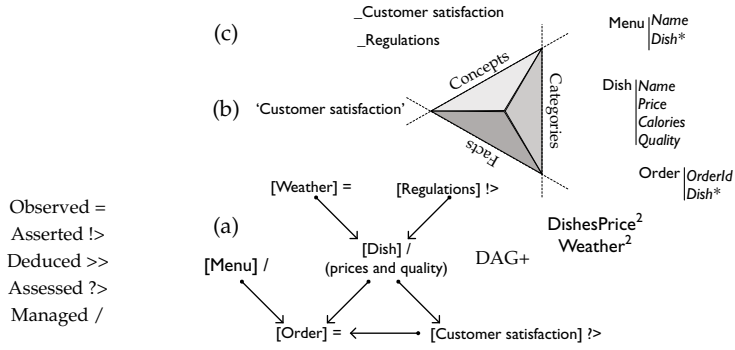


Figure 6.10 Untangling empirical (a), nominal (b), and intentional (c) causalities.

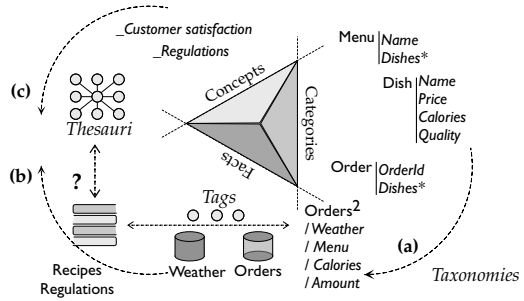


Figure 6.11 How to align vocabularies shared between (a) categories, (b) data, and (c) concepts.

entities, floating in semantic limbo. Given a flat data model without the support of a conceptual thesaurus, empirical reasoning can only rely on data catalogs. For instance, a marketing team is trying to understand the factors that impact restaurant orders. To that end, the team relies on a sample of orders with four observed variables (weather, menu, calories, and amounts) and two unobserved variables (customer satisfaction and regulations). Without delving into analytical details, a straightforward approach would be to tag observations according to partitions that could explain orders, such as cold weather or meal calories. Whether or not the tags (or labels) coincide with terms already used in documents, reasoning will remain limited in the absence of semantic consistency between databases, catalogs (e.g., orders), new datasets (e.g., weather), and conceptual thesauri (Figure 6.11).

It follows that, in the absence of conceptual anchors, counterclockwise reasoning from domain-specific dictionaries could make room for semantic discrepancies. Conversely, clockwise reasoning from datasets relies on meanings assigned to taxonomies that may or may not align with thesauri's prior understandings. These issues, commonly referred to as shared vocabularies or semantic layers, are generally addressed through various combinations of semantic graphs and language models.

6.4.2 Arguing with Facts

Empirical reasoning is inherently prone to unreliable data, divergent information, and subjective knowledge. Since the associated challenges are often addressed from different perspectives, typically business analysis and data science, reasoning can be enhanced when guided by argumentation profiles (Figure 6.12).

6.4.2.1 Empirical Challenges

Empirical reasoning is intended to explore new territories and is consequently driven by conflicting understandings of data, information, and knowledge.

6.4.2.1.1 Unreliable Data

Reasoning is, to some extent, supposed to be rooted in facts; however, facts are not given – they can only be observed through data, which usually provide a partial, biased, and/or temporal reflection of what occurs in actual environments. This issue becomes even more critical when data turns out to be the main resource for pervasive GenLMs that lack the human common sense needed to separate the wheat from the chaff. Ontological modalities of facts, along with metadata, can help address these issues by identifying data sources and their caveats.

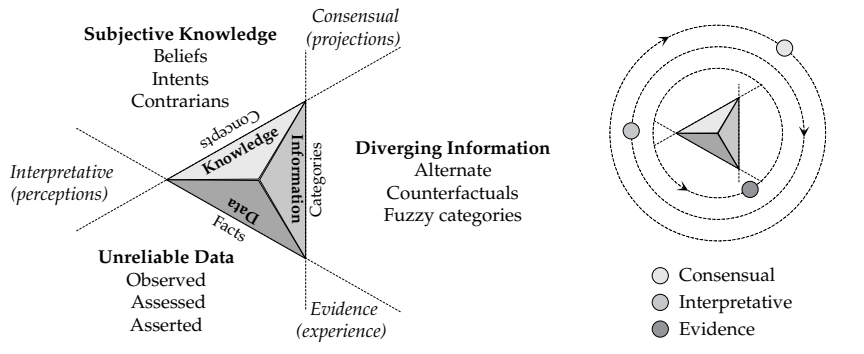


Figure 6.12 Reasoning challenge (left) and argumentation profiles (right).

6.4.2.1.2 Misleading Information

While reason may be intuitive, reasoning is meant to be explicit and, consequently, to rely on categorized data and shared representations. When empirical reasoning is driven by renewed business purposes, chances are that such representations are not fully aligned with newly relevant data. That's when the back-and-forth moves to and from shared representations can help to align the reasoning guardrails set by domains and taxonomies.

6.4.2.1.3 Subjective Beliefs and Intents

Finally, individual reasoning is driven by mental representations, which are a combination of manifold beliefs, intentions, and knowledge. In that case, the back-and-forth adjustments will address the reasoning guardrails set by shared vocabulary (thesauri) and conceptual responsibilities (domains).

6.4.2.2 Argumentation Profiles

The aim of argumentation profiles is to compensate for reasoning hurdles, establish common ground, and, when attainable, promote consensus. To that end, reasoning profiles are suggested based on the comprehension pivots introduced in Chapter 3: evidence (experience), interpretative (perceptions), and consensual (projections).

6.4.2.2.1 Evidence Reasoning

Mutual comprehension begins with an agreement on the facts. Assuming that data sources and their temporality are properly identified, the aim of evidence reasoning is to use learning curves to build consensus regarding the mapping of data into information. With ontological prisms, data itself is managed uniformly, enabling agreed-upon sources, relevance, and reliability. This paves the way for the second step of evidence reasoning: using empirical evidence (e.g., Bayesian or counterfactual inference) to ensure the practical convergence of reasoning paths conducted from different perspectives.

6.4.2.2.2 Interpretative Reasoning

Parties to arguments may agree on the facts but diverge in their perceptions. Interpretative reasoning focuses on achieving agreements about meanings regardless of factual evidence or goals. With ontological prisms, interpretative reasoning is explicitly supported by the distinction between terms and meanings. Semantic inference (e.g., metonymies and metaphors) can then be used to prop up conceptual commons.

6.4.2.2.3 Controversial Reasoning

As a symbolic tool, reason is usually called upon to serve purposes other than the pursuit of truth. And when conflicting perspectives are involved, reasoning

paths are driven by different intents, based on different categories, and refer to different facts. In such cases, ontological prisms allow reasonings to be conducted conjointly across alternative mental representations, as well as agreed-upon facts and categories. The issue will be further addressed in Chapter 7 through the pairing of ontological prisms with the observation–orientation–decision–action decision-making model.

6.5 Reason and Knowledge

We began this chapter by relating reason to truth; we conclude by relating it to knowledge, and more precisely, as the engine of actionable knowledge.

6.5.1 Actionable Knowledge

As stated by Davis, Shrobe, and Szolovits in their seminal paper (introduced in our preamble), knowledge is of little use without a fragmentary theory of intelligent reasoning. Thus, after considering reasons in terms of formal constructs and cognitive capabilities, we now address it in terms of knowledge representation.

6.5.1.1 Declarative Knowledge Representation

In order for knowledge representations to be actionable and readily renewable, they must be declarative. Otherwise, they would epitomize Zeno's Achilles paradox. With knowledge (Achilles) continuously racing against new facts (the tortoise), procedural representations would either be lagging behind or ahead of reality, but never in sync. This means that some dual latency (hindsight or foresight) would be unavoidable.

While KGs, which serve as a de facto implementation standard, embody declarative approaches, they do not scale well and can become cumbersome to maintain or evolve. This drawback appears to be inherent to KGs; they can be very effective in representing a wide range of dynamic systems, yet they are essentially static when left on their own. It follows that, when designing KGs, one needs to distinguish between constraints, which can be directly and effectively managed through KGs, and rules, which are inherently dynamic and, when implemented in scaling KGs, induce exponential complexity.

These general guidelines for KG become pivotal for heterogeneous ontologies due to the encroachment of rule footprints. Rules are defined by their scope: triggering conditions (rule's domain) on the left hand, affected footprint (rule's co-domain) on the right hand. In heterogeneous ontologies, rules are homogeneous when they can be evaluated and executed within the same realm – whether

intensional, extensional, or logical. They are heterogeneous when they are evaluated and executed in different realms. It follows (see Chapter 4) that heterogeneous rules induce a higher level of complexity and should therefore be refactored as homogeneous ones whenever possible.

6.5.1.2 Heterogeneous Reasoning

Reasoning cannot be reduced to logic, and logic cannot be reduced to mathematical logic. These issues, generally addressed by pluralist theories of truth (in philosophy) or modal logic (in mathematics), gain practical relevance when framed in ontological terms. Ontologists typically distinguish between deontic and alethic logic: the former pertains to what can be clearly ascribed to social conventions, while the latter relates to everything else. Ontological prisms are more specific regarding these issues and, consequently, about the types of reasoning that can be appropriately applied.

Extensional reasoning pertains to the possibility (or impossibility) and necessity (or contingency) of things being or phenomena occurring. As a corollary, the facts considered are nonsymbolic and are not meant to be ascribed to willing agents.

Intensional reasoning pertains to the mandatory (or elective) and permissive (or prohibitive) status or behavior of socially defined entities. As a corollary, the facts considered are symbolic and are meant to be ascribed to willing agents.

Logical reasoning pertains to symbolic representations regardless of their extensional or intensional counterparts.

Ontological reasoning will thus rely on three kinds of rules pertaining to existential, intensional, and logical realms. Endogenous reasoning is meant to be carried out within realms, while heterogeneous reasoning relies on inferences that combine different kinds of rules: logical and empirical (facts and categories), semantic (facts and concepts), and deontic (concepts and categories) (Figure 6.13).

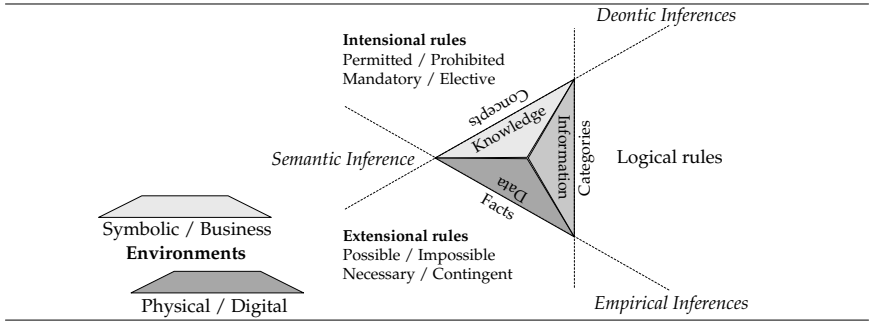


Figure 6.13 Rules footprint and ontological reasoning.

It must be stressed that ontologies are purely symbolic systems: the rules they encompass solely pertain to knowledge and are not supposed to operate on environments; otherwise, operational rules are defined with agents at the modeling level (see Chapter 2).

6.5.1.3 Rules Representation

Whatever logic paths and modalities ontological prisms may employ, they should function as TMS, ensuring consistent rule-based reasoning across facts, concepts, and categories. In principle, this objective could be achieved with ontological languages like OWL and SKOS. Ontologists could thus take advantage of the RDF, the standard implementation of both OWL and SKOS, to attach rules and constraints to properties. In practice, using annotation systems like SKOS to implement rules substitutes procedural complexity for structural complexity, undermining the declarative benefits of KGs. This makes the maintenance of heterogeneous ontologies cumbersome, and as they scale, the systems become overwhelmed by exponential complexity.

To avoid scaling constraints and in line with the principle of separation of concerns, ontological prisms rely on dual architecture. On one hand, annotation systems such as SKOS are ignored and reserved for external business-specific developments. On the other hand, ontological prisms allow for rules to be expressed as OWL entities (see Annex), with CLIF and the Prolog logic programming language serving as references.

6.5.2 Ontological Inference

For all intents and purposes, knowledge is heterogeneous, which makes for heterogeneous reasoning when rules are set across realms.

6.5.2.1 Inference Patterns

Ontological prisms draw inspiration from Peirce's principles to define reasoning patterns in terms of intrinsic and functional connectors (see Chapter 2). These patterns are illustrated in relation to facts but can also pertain to concepts and categories (Figure 6.14):

Deductions combine intrinsic knowledge of facts and categories to extend intrinsic knowledge of facts. For instance, if [spinach] is a subset of [vegetable] (both facts) and Vegetable is a subtype of Plant (both categories), then [spinach] can be represented by Plant.

Inductions are hypothetical deductions in that they use functional knowledge of facts and categories to extend functional knowledge of facts. For instance, if

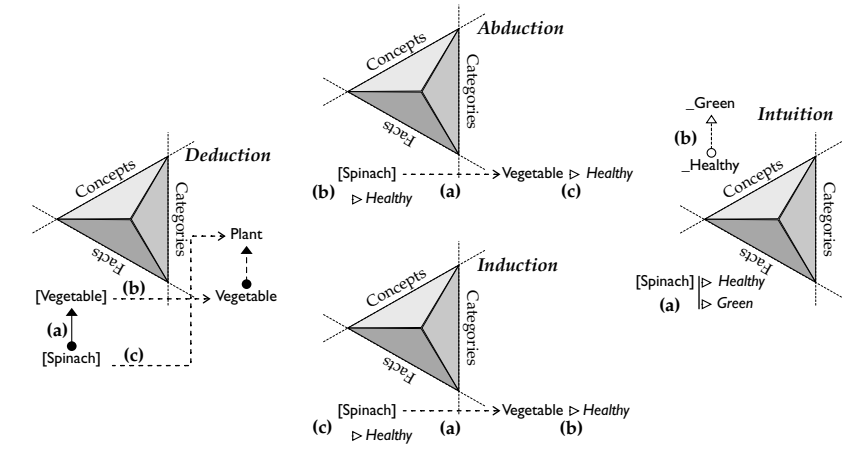


Figure 6.14 Logical inference and intuition (notations in Annex).

[spinach] (a fact) is represented by Vegetable (a category) and Vegetable has a *healthy* (an aspect) property, then [spinach] has a *healthy* property.

Abductions are inductions about categories instead of facts. For instance, if [spinach] (a fact) is represented by Vegetable (a category) and has a *healthy* (an aspect) property, then Vegetable could have a *healthy* property.

Intuitions (or insights) are nonsymbolic inferences that operate independently of categories. For instance, if [spinach] (a fact) has *green* and *healthy* properties, then _Green and _Healthy concepts may be functionally associated. More generally, intuition appears as the basis of creative thinking operating across all ontological realms.

6.5.2.2 Empirical Inference

As illustrated with the examples above, empirical inference – and therefore reasoning – is driven by observations. Broadly known as conditional or Bayesian logic (from Bayes' theorem), empirical inference uses statistical methods to weigh the probability of hypotheses given prior evidence. Taking advantage of factual modalities such as observed, assessed, expected, etc. (see Chapter 2), Bayesian principles (see Pearl, J., Mackenzie, D., (2018) and Clayton, Aubrey (2022)) can be applied dynamically to guide the course of reasoning. For instance, the probability of a vegetarian dish being chosen on a rainy day is given by the ratio of the probability of simultaneous occurrence to the probability of a rainy day. Empirical inference can push things even further by applying Bayesian logic negatively, introducing counterfactuals (facts that didn't happen) to support a contrario reasoning.

6.5.2.3 Semantic Inference

Semantic inference, or figures of speech, does with words what logical inference does with facts. Figures of speech, for long a key topic of semiotics, may now get a new relevance as an antidote to the confusion introduced by GenLMs between semiotics (signs) and meanings (symbols). Without delving into semiotic terminology, two standard figures of speech epitomize cases of sentences being substituted for something other than their literal meaning: metonymies for extensions, metaphors for intensions (Figure 6.15).

6.5.2.3.1 Metonymies

With metonymies, individual (§) parts of salient referents are used to represent the referents themselves. Since this representation occurs at the semantic level, the nature of individual contexts is irrelevant: it can be structural (components), functional (activities), or temporal (events). With ontological prisms, metonymies are expressed through associations. For instance, 'Washington' could stand for the White House, and 'The White House' could stand for the US President.

6.5.2.3.2 Metaphors

Metonymies operate on individuals (e.g., §White House), while metaphors operate on types (e.g., [Pointing Device]) detached from literal meanings. For example, consider "throwing in the towel" and "throwing down the gauntlet": both can be used as metaphors for conceding defeat and challenging an opponent, respectively, because they represent a break from literal meanings. The comparison also illustrates the role of understanding the original contexts where actual deeds combine with meanings, while simultaneously maintaining a necessary distance: today's challenges no longer involve actual gauntlets. With ontological prisms, metaphors are expressed through functional (nonintrinsic) conceptual subsumption, e.g., the respective form of a mouse and a pointing device.

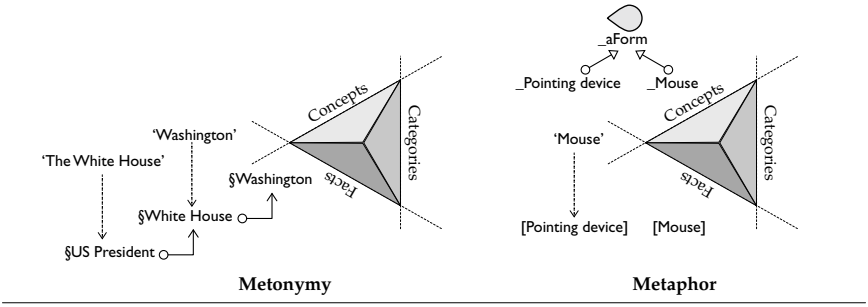


Figure 6.15 Metonymies use physical or functional proximity between individuals (§) to match extensions, Metaphors use similarity between types (e.g., [Pointing device]) to abstract concepts (e.g., _aForm) (notations in Annex).

Metonymies and metaphors correspond to the third and fourth layers of semantic vector translation mentioned above with regard to empiric and ontological spaces, respectively.

6.5.2.4 Deontic Inference

Deontic inference is at the core of transparency, traceability, and accountability in enterprise governance. In this context, ontological prisms can be used to characterize the deontic dimension of reasoning steps in relation to the status of resources and the nature of rationality involved: institutional, professional, corporate, social, or personal (see Chapter 4). These issues will be discussed in relation to decision-making processes in Chapter 7.

6.5.3 Reason and Collective Learning

Great leadership clearly helps, but organizations primarily thrive through collective learning. It is the only way to transform individual tacit and explicit capabilities into shared ones shaped by the culture and objectives of the organization, as well as by specific functional or operational contexts. In this regard, collective reasoning is doubly critical: from an organizational perspective, it allows for a fusion of individual and collective behaviors; from a systems perspective, it ensures the integration of computational (systems) and cognitive (people) levels.

6.5.3.1 Communication and Organization

Attempts to build collaborative architectures (known as agentic AI) that combine GenLMs and AI agents are hindered by two critical blind spots. The first pertains to conversation. While the limited reasoning capabilities of GenLMs can be viewed as a key factor in the development of cognizant AI agents, the need for specific agentic middleware supporting the collaboration between them and GenLMs has been largely ignored. It is also safe to assume that replicating with AI agents the model of human prompts is not a viable option. The second blind spot pertains to organizations, which are implicitly subsumed into agentic architecture, leaving isolated users facing distributed nebulae that compound complexity and opacity.

With regard to collective reasoning, these blind spots have major consequences. First and foremost, agentic architectures effectively make no room for direct collaboration between people, definitely a key factor behind thriving organizations. Second, such architectures stamp out any functional distinction between the respective contributions of systems and people, undermining effective accountability and governance.

Attempts to address these issues can be observed at two levels. With regard to conversations, advances are focused on the management of dynamic contexts; with respect to organizations, the focus is on using ontologies to manage

shared knowledge. Contexts, as well as ontologies, rely on KG representations. While these developments highlight the potential benefits of a functional distinction between language processing and knowledge management, they fall short in addressing the core issue, namely, how to use reasoning to enhance collective learning capabilities.

6.5.3.2 Learning Organizations

As noted above, the distinction between explanation and comprehension can be aligned with the divide between personal and collective reasoning, which further suggests that explainability plays a critical role in the two main organizational dimensions. The first dimension pertains to the integration of individual and collective tiers: directly, explainability ensures the consistency and sustainability of organizations' objectives, incentives, and accountability; indirectly, it acts as a means of integrating individual motivations and capabilities with collective culture and experience. The second dimension pertains to the functional integration of systems and people: while both can contribute to explainability, only people, individually or collectively, can ensure comprehension and take responsibility for judgment and decision-making. Framed by ontological prisms, reasoning capabilities can be used to leverage learning capabilities through people and systems organizational dimensions.

Key Points

We began by framing the scope and paths of reasoning in terms of facts, concepts, and categories, and then provided a quick overview of nominal, propositional, and predicate logic.

Next, we engaged with the aims of reason. We approached this at a functional level, typically distinguishing between correlation and causation, and at a cognitive level, typically distinguishing between interpretation, explanation, and comprehension. Following that, we used this foundation to assess the reasoning capabilities of GenLMs and to explain their hallucinations.

The fourth segment addressed empirical and analytical reasoning, focusing on the distinction between evidence, interpretative, and controversial reasoning. Finally, the last segment examined the interplay of reason and knowledge from two key perspectives. First, reasoning through heterogeneous ontologies has been defined in terms of empirical, semantic, and deontic inferences. Second, we discussed collective reasoning in terms of collaboration, systems, and learning organizations. Chapter 7 considers the benefits of knowledge-driven decision-making.

7

Decision-making with Ontological Prisms

A good decision is based on knowledge and not on numbers

Plato

7.1 Decision-making Dimensions

Decision-making is best defined in relation to reasoning and problem-solving. Unlike reasoning, which pertains solely to symbolic realms, decision-making is intended to affect environments, whether or not it is supported by reasoning. Unlike problem-solving, which can be confined to atemporal and bounded spaces, decision-making operates within open contexts, carries ambivalent semantics, and involves interested parties who are committed to and accountable for actual operations and time frames. This chapter considers the benefits of harnessing ontological prisms in enterprises' decision-making processes, taking as a reference the observation–orientation–decision–action (OODA) blueprint.

7.1.1 Intrinsic Assessment of Decision-making Processes

The distinction between problem-solving and decision-making highlights what characterizes decisions: they involve agents, stakes, and commitments. This intrinsic assessment can be set forth along the usual five points of What, Who, How, Where, and When (Figure 7.1).

7.1.1.1 What: The Matter of Decisions

Contrary to problems that can be defined unequivocally and solved independently, decisions are shaped by semantics, time, and uncertainties. This leads to a dual constraint that is specific to decisions: exogenous complexity (boundless spaces) and risks (boundless time frames).

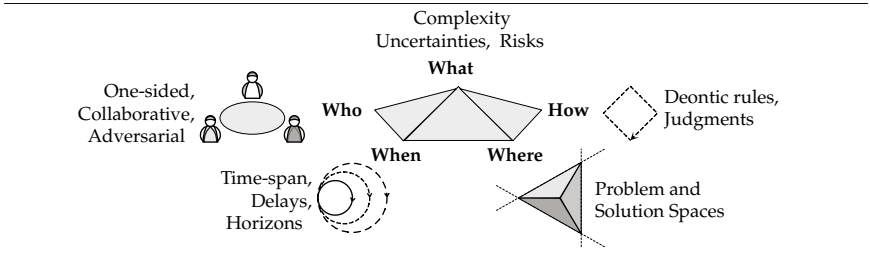


Figure 7.1 Intrinsic assessment of decision-making processes.

7.1.1.2 Who: Decision-makers

Compared to problem-solvers who don't have to consider interested parties and are not supposed to be directly concerned by outcomes, decision-makers have skin in the game, whether that game is one-sided, collaborative, or adversarial.

7.1.1.3 How: Decision-making Processes

Problem-solving can fully rely on reasoning, algorithms, and heuristics; by contrast, decision-making processes are primarily governed by deontic values and objectives, and judgment.

7.1.1.4 Where: Problem and Solution Spaces

Problems and their solutions are meant to be set within single and well-defined bounded spaces; by contrast, decisions entail separate and partially defined problem and solution spaces framed by intents which may or may not be aligned.

7.1.1.5 When: External Time Frames and Events

While problems can be wrapped in time capsules, decisions are conditioned by external time frames and subject to external events.

This intrinsic assessment of decision-making processes is generally overlooked by supporting systems whose functional architecture is defined by their origin.

7.1.2 Decision-making Environments

Decision support systems (DSSs) can be characterized by the way they integrate two key technologies with their legacy architecture. First, what is ambiguously referred to as knowledge graphs (KGs) has, for some time, served as a modeling Swiss Army knife for a wide range of issues. However, the surge in knowledge-driven applications has placed them in every corner of information systems. The integration of KGs into DSSs has been facilitated by KGs' continuity with more established methods, such as conceptual models. In contrast to the replacement of conceptual models by, or their rebranding as, KGs, the integration

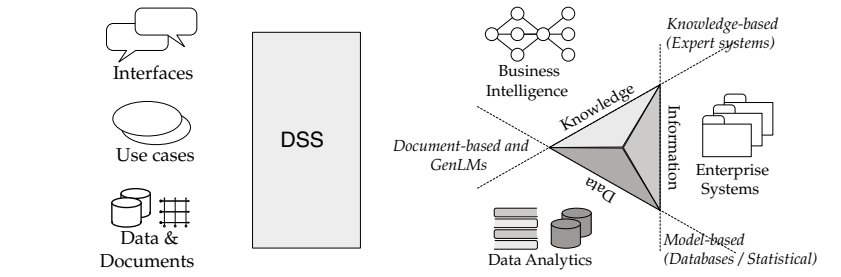


Figure 7.2 An ontological view of DSSs.

of generative language models (GenLMs) is being more abrupt and disruptive, as this technology entails radical changes to the legacy architecture of DSS. Assuming customary resources, use cases, and user interfaces, these architectures can be grouped into three categories (Figure 7.2).

7.1.2.1 Document-based Decision Support Systems

From a functional perspective, document management systems (DMSs) are more about search than rescue, making them much closer to ontologies than to decision-making. On the other hand, DMSs are also the first to benefit from KGs and GenLMs, which present a challenge along with an opportunity: on the one hand, using KGs to transform documents into a treasure trove of knowledge, on the other hand, using GenLMs to explore that knowledge. Still, these undertakings may fall short with regard to decision-making.

7.1.2.2 Model-based Decision Support Systems

What is commonly known as model-based DSSs are extensions of two different types of tools: enterprise-oriented tools and analytical tools. The former, typically database management systems and enterprise resource planning systems, rely on endogenous resources and associated schemas (information). Additional analytical tools are applied to these resources, schemas are used to build KGs, and GenLMs are introduced for user access.

Conversely, the latter rely on proprietary platforms that utilize sophisticated mathematical and statistical tools, which are applied to both exogenous and endogenous resources (data). Compared to those focused on enterprises' endogenous resources, the decision-making functions of analytical platforms are built-in capabilities, ensuring a smooth integration of KGs. This is also true for the integration of GenLMs, as analytics-based DSSs are designed to process documents and datasets. Along with major providers like IBM, Oracle, and SAP, contenders in specific segments of enterprise software have had to follow suit and add KGs and GenLM packages with decision-making capabilities.

7.1.2.3 Knowledge-based Decision Support Systems

While the distinction between model-based and knowledge-based DSSs is progressively blurred by the generalization of KGs and GenLMs, it remains relevant for pure players built specifically for this purpose, without being hindered by legacy architectures. Developed as integral components of large multidomain ontologies, such as those in healthcare (see Chapter 1), these systems appear to renew the symbolic approach of expert systems from the 1970s and 1980s, with two critical differences: their access to an almost unlimited corpus of data and the advantages of machine learning technologies.

Concomitant with these substantial and principled developments, attempts are being made to develop distributed networks of AI agents that could support decision-making as microservices.

7.1.2.4 Decision-making and Ontologies

All in all, whether they are limited by the reasoning capabilities of GenLMs or driven by the representation capabilities of KGs, most decision-making systems are currently engaged in some form of ontological development. From a general perspective, such developments should lead to a differentiated understanding of problem and solution spaces, more transparent processes and outcomes, a principled approach to time and uncertainties, and, most importantly, a seamless integration of decision-making and knowledge engineering processes.

7.2 Problem and Solution Spaces

The clearer demarcation line between problem-solving and decision-making is the distinction between problem and solution spaces. This line can be overlooked when addressing problems, but it cannot be ignored when decisions need to be made.

7.2.1 Decision-making Spaces

Divide and conquer strategies have a long-established track record for tackling complexity, and they can be especially effective in addressing heterogeneous issues.

7.2.1.1 Microstates and Macrostates

In cybernetics parlance, a microstate reflects the positions and momenta of all individual particles or components that make up a system at a given moment. In contrast, a macrostate represents the positions and momenta of the same individuals as determined by categories built from selected parameters such as temperature, pressure, volume, or density.

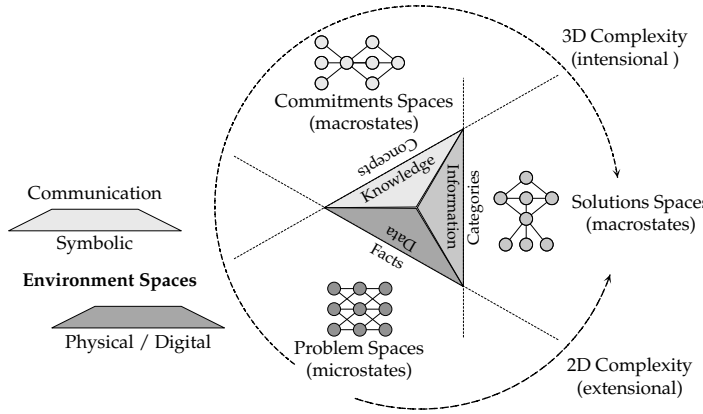


Figure 7.3 Problems and solution spaces.

When applied to ontological prisms viewed as truth maintenance systems, a microstate represents the set of all individual facts deemed relevant in the considered environment, while a macrostate represents the set of ontological commitments (concepts) or representations (categories) pertaining to those facts. When decision-making is considered, microstates and macrostates can be used to represent problem, commitment, and solution spaces, respectively (Figure 7.3).

7.2.1.2 Decision-making Spaces

Problems can be neatly encapsulated and solved in the same space, independent of context; that's not the case with decisions, as they have consequences. This distinction is best defined in terms of problem and solution spaces. The former, represented by microstates, refers to the terms of the problems, while the latter, represented by macrostates, refers to the categories that represent the solutions. In between, the commitment space, also represented by macrostates, refers to the concepts used to represent interpretations, intents, goals, policies, and so on.

That understanding partially aligns with the traditional 2D data modeling approach, where the problem space pertains to observed data, and the solution space pertains to the categories (or models) meant to represent these data. Ontological prisms introduce perspectives. To begin with, self-contained problems can be rooted in conceptual or logical realms as well as in facts. To take a few examples: how to boil eggs (factual), how many angels can dance on the head of a pin (conceptual), and how to find the surface area of a circle (logical). Each of these examples can be addressed within its original space because they are problems to solve, not decisions to make: their respective micro and macro states can

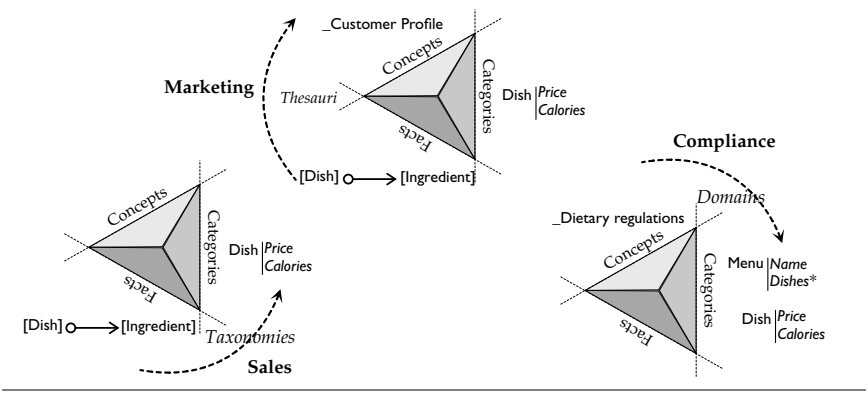


Figure 7.4 Decisions are set across realms.

be mapped directly, whether physically (facts) or symbolically (concepts or categories). By contrast, most decisions at the enterprise level are set across different realms, calling for explicit decision-making processes, for example, (Figure 7.4):

- Sales: using taxonomies to price dishes according to ingredients (facts/categories).
- Marketing: using thesauri to market dishes according to customer profiles (facts/concepts).
- Compliance: collaboration between domains to design dietary menus according to labeling regulations (concepts/categories).

Commitment spaces have broader ontological significance when they are used to characterize not only the solutions, such as through policies or priorities, but also the organization of the solution spaces. This includes representation methods, such as functional or object-oriented approaches, as well as the level of the solution, typically whether it is technical or conceptual.

7.2.2 Decision-making Templates

7.2.2.1 Decisions Intrinsic Footprint

The primary benefit of ontological prisms is the ability to decompose complex decisions into elementary ones along the reasoning steps identified in Chapter 6. Decisions can then be characterized by their intrinsic footprint and their composition. Regarding footprints, the distinction is between operational decisions, which pertain to the mapping of facts into categories (e.g., sales), and symbolic decisions, which have no direct consequences on facts (e.g., marketing and compliance). In terms of composition, the distinction is between elementary

and structured decisions; the former involves sequences of self-contained issues (e.g., setting prices), while the latter consists of a combination of intertwined issues with potential overlapping decisions and feedback (e.g., setting prices in accordance with regulatory compliance policies).

7.2.2.2 Decision-making Templates

Footprint and composition can be used to define decision-making templates depending on the mapping of problem spaces into commitment and solution spaces: empirical, interpretative, and organizational (Figure 7.5):

- *Empirical* decisions are based on experience, when the alignment of actual situations with categories can be directly assessed and operations tuned accordingly, e.g., managing inventories according to levels.
- *Interpretative* decisions involve perceptions, i.e., the alignment of meanings with intents without reconsidering intents or projections; e.g., managing inventories in relation to market prices.
- *Organizational* decisions involve projections, i.e., the way intents and deontic rules are applied to perceptions; e.g., managing inventories in relation to compliance with new labeling regulations.

These templates coincide with the reasoning modalities set forth in Chapter 6, and can be further characterized with regard to process execution and time frame horizons (Figure 7.6).

- Process execution is determined by the nature of the issues at stake: one-sided (aligned issues), collaborative (nonaligned yet nonconflicting), and adversarial (conflicting issues).
- Time frame horizons are established based on the predictability of environments and, consequently, on the type of risk management: deterministic (or operational), stochastic, or strategic (i.e., non-stochastic).

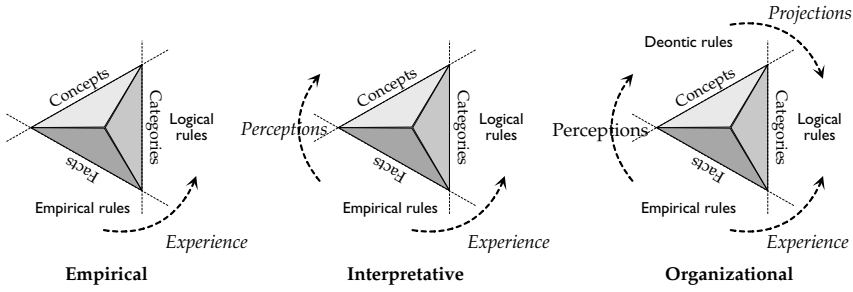


Figure 7.5 Decision footprints.

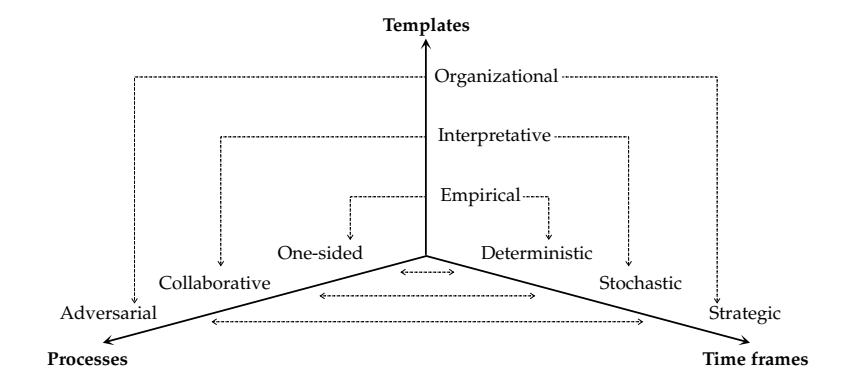


Figure 7.6 Decision-making templates are characterized by processes and time frames.

Both dimensions relate to data, information, and knowledge: the processes through which these elements are intended to support reasoning and judgment, whether for collaboration or competition, and the time frames that affect the availability and reliability of relevant data, and consequently, the supporting knowledge chain.

7.3 Visibility and Time Frames

Decision-making is arguably a key challenge of enterprise governance, struggling to find a way between the Charybdis of obfuscation and the Scylla of dilution.

7.3.1 Transparency and Traceability

In contrast to problem-solving, decision-making is inherently a matter of judgment, and assessments typically occur long after decisions have been enacted. Until then, the primary objective of governance should be to ensure that all relevant elements are considered and utilized by contributing agents to the best of their abilities. With ontological prisms, that can be achieved through transparency and traceability, the former relating to the objectives, resources, and conditions determining the decisions, the latter relating to the ways decisions have been reached. Assuming that everything is being done to ensure access to data, information, and knowledge, improvements are considered under four angles: what is to be decided and why; who will decide, contribute, and take responsibility; and when and where options should be considered. The organization of processes will be discussed in Section 7.5.

7.3.1.1 What Is to Be Decided, and Why

The primary objective is to clearly identify issues and stakes. Using ontological prisms, this is achieved by mapping territories in relation to relevant facts, existing legacies (categories), and governance principles, processes, and methods. Issues should then be organized in a manner that ensures consistency in references and semantics. Equally important is the need to build arguments in support of the decisions to be made, which is achieved by documenting intents, conditions, and objectives at a conceptual level. The importance of defining the context of decisions in terms of commitments (concepts) and legacy realizations (categories) cannot be overstated, along with the benefits of heterogeneous ontologies.

7.3.1.2 Who Is to Contribute and Be Accountable

Decisions result from various types of contributions (typically observation, reasoning, or judgment) made by different kinds of agents (typically individuals, collectives, or systems) with different types of entitlements (expert, interested party, or manager). Except for explicit policies, there is no reason to assume nonoverlapping contributions or entitlements, nor exclusive associations between the respective partitions. Not to mention the blurring impact of digital architectures and generative artificial intelligence (GenAI) technologies on the boundaries between people and systems contributions. On that account, the primary objective should be to ensure transparency regarding agents' entitlements.

7.3.1.2.1 Entitlement and Contributions

While systems can be expected to support observations and reasoning, they are not meant to extend their reach by making judgments, let alone commitments. These, and therefore decisions, can only be entitled to people at a personal, functional, or organizational level. At the personal level, entitlements are attached to named individuals; at the functional level, they are attached to named individuals in relation to their statutory position within a structure (e.g., a committee) or activity (e.g., a project team); and at the organizational level, entitlements are associated with statutory units. Regardless of the role of supporting systems, processes should ensure, and transparency should demonstrate, the distinction between reasoning (by people or systems) and judgment and commitments (by people exclusively). Furthermore, transparency should allow for checking the consistency between contributors' entitlements and contributions throughout the processes.

7.3.1.2.2 Explainability and Accountability

Decisions, whether executed directly or through commitments, introduce an inflection point between explainability for the day before and accountability

for the day after. This inflection point becomes critical when individual contributions must be transformed into collective commitments. As seen in Chapter 6, explainability is the only way to ensure a shared understanding of the stakes and solutions. It should, therefore, be fully supported by systems, particularly because the inflection points between explainability (before the decisions) and accountability (after the decisions) may be elusive when timing is an integral part of decision-making. Adding to its significance for decision-making processes, explainability operates differently within and outside organizations, which becomes a critical issue when decisions are adversarial. These points are addressed below, along with time frames.

7.3.1.3 Where Options Should Be Considered

Organizational decisions are, by nature, transverse, leading to matrices of bounded arbitrations. Since it is safe to assume that not all arbitrations can be settled upfront, a number of intermediate decisions will need to be made during processes concerning schedules, sequencing, or representations. By leveraging transparency and traceability across ontological prisms, these branching points can be managed in terms of perceptions, projections, and experience.

Perceptions relate to how decisions can be supported by available data (facts) or whether improvements are necessary. Projections refer to the intents and/or expectations of the parties involved (as represented through concepts), which may need to be reassessed. Experience pertains to the follow-up and learning curve of processes themselves.

How branching options can be effectively managed is considered in Section 7.5.

7.3.1.4 When Decisions Should Be Made

Time is what occurs between events, and events reflect changes in environments. It follows that time scales can be individual when events are perceived by living organisms or collective when events are perceived by organizations. In this context, we will only consider collective time scales. With regard to decision-making, relevant events are facts characterized by their native environment, time stamps are set by that environment, and events' shelf lives are determined in both their native environment and the context of decision-making processes. Time-stamped events can be used to define overall time frames, such as calendar events for the publication of corporate records; as triggers for intermediate decisions within orchestrated decision-making processes, such as departmental results for consolidated sales records; or to define local time spans, such as the shelf life of estimated monthly statistics.

Decisions at the corporate level often rely on alternative scenarios built from intersecting paths set across overlapping time spans. Keeping track of the intricacies of these time scales throughout the decision-making process can be best

achieved with heterogeneous ontologies that combine the representation of virtual facts (assessed, expected, etc.) with agents' expectations (concepts).

7.3.2 Decision-making Horizons

The larger or more complex organizations are, the longer the time spans and the farther the horizons of commitments. Given the dynamics between changes in facts, information, and knowledge, it's safe to assume that commitments set on different and/or overlapping time scales are bound to operate on heterogeneous or misaligned data, categories, and/or concepts. While the terms may vary, this issue is commonly addressed as operational, tactical, and strategic decision-making. Ontological prisms allow for a more principled understanding of these concepts (Figure 7.7).

Operational horizons are set by observed data that can be directly mapped to information and put to use as knowledge, thereby allowing for routine decision-making. For example, a food packaging company working with restaurants would choose suppliers of fresh fruits and vegetables weekly based on their prices. Operational horizons correspond to empirical decisions mentioned above.

Tactical horizons are set by the alignment of observations (data) and causal chains (categories) within a given time frame. Partial or imperfect information (i.e., data addressed through categories) can be improved with additional data until reliable causal chains can be established. Accordingly, our food packaging company will make monthly plans based on observed taste trends, weather forecasts, and contracts with suppliers. Tactical horizons correspond to interpretative decisions mentioned above.

Strategic horizons are introduced when commitments must be undertaken without enough supporting information and knowledge. When uncertainties about facts, unproven causal chains, and/or unpredictable factors prevent reliable assessments within specific time frames, risk management schemes must be introduced to mitigate the impact of adverse developments. In our example, a strategic horizon could be set to address the risk of supplies being affected by

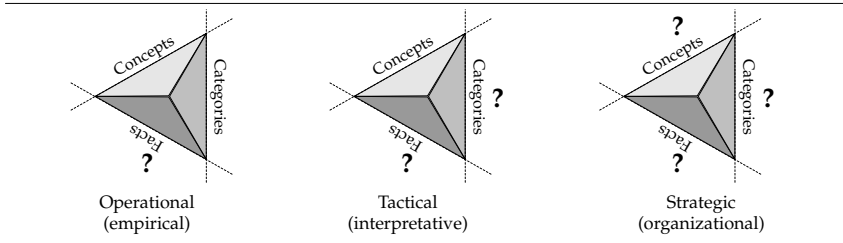


Figure 7.7 Decision-making horizons.

natural phenomena and/or currency fluctuations. Strategic horizons correspond to organizational decisions mentioned above.

In theory, decisions should be clearly categorized as empirical, interpretative, or organizational, and designated for operational, tactical, or strategic horizons, respectively. In practice, however, this is more the exception than the rule. First, because horizons set at different tiers are bound to overlap; and second, because horizons are established based on the availability and reliability of information, which may change while decisions are pending.

7.4 Parties and Structure

Taking inspiration from Albert Einstein's famous quote, "The only reason for time is so that everything doesn't happen at once," we could say that, when organizations are concerned, the purpose of decision-making processes is to freeze time while agreements are reached and commitments can be made. It follows that the structure and control of these processes must be defined according to the types of issues at stake, and consequently, the parties involved.

7.4.1 Sided Decision-making

To begin with, the distinction between one-sided and multi-sided decision-making processes can be added to the templates introduced above (Figure 7.8):

- One-sided decisions rely on shared observations, are entitled to a single agent, individual or collective, driven by its own actual judgment based on established reasoning references.

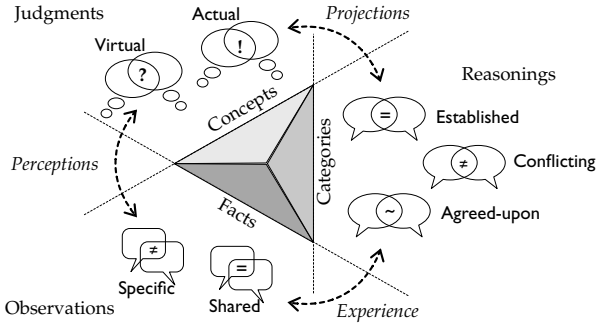


Figure 7.8 One-sided, collaborative, and adversarial footprints.

- Collaborative decisions rely on shared and specific observations, are entitled to multiple agents, individual or collective, each agent driven by its own actual judgment, all based on agreed-upon reasoning references.
- Adversarial decisions stem from shared and specific observations, are entitled to multiple agents, individual or collective, each agent driven by its own actual judgment as well as virtual ones for the judgments allegedly made by other agents, each possibly based on conflictive reasoning references.

7.4.1.1 One-sided Processes

One-sided decision-making processes can be viewed as the organizational equivalent of problem-solving, with the difference that external environments and time frames must be taken into account. All contributors share the same experience and perceptions and use the same reasoning references (empirical footprint) concerning a deterministic environment (operational horizons).

7.4.1.2 Collaborative Processes

Collaborative decision-making processes involve parties with specific experience and perceptions of a stochastic environment (tactical horizons), with different but congruent reasoning references (interpretative footprint).

Collaborative issues create opportunities for win-win outcomes and mutual contributions between parties. Collaborations can be explicit, established through rules or negotiation, or latent, arising from independently made decisions. When collaborative decision-making processes are supported by ontological prisms, they can be aligned with reasoning processes, with knowledge modalities used to represent the actual or hypothetical intents and objectives of different parties. Furthermore, the dynamics of their respective behaviors can be made explicit through the distinction between reasoning and judgments provided by the ontological prisms. This will induce a convergence of rationales and the emergence of mutually favorable decisions.

7.4.1.3 Adversarial Processes

Adversarial decision-making processes involve parties with specific experience and perceptions of a non-stochastic environment (strategic horizons), with parties driven by possibly conflicting intents and reasoning references (organizational footprint).

Adversarial decision-making encompasses multisided competitive issues that can be situated in bounded environments, where gains are balanced by losses (zero-sum issues), or in boundless environments, where gains and losses cannot be determined upfront (nonzero-sum issues). The former entails fully assessable outcomes, allowing for improvements based on comparisons, while the latter results in multidimensional outcomes that cannot be unequivocally assessed upfront.

The distinction between zero-sum and nonzero-sum issues, a cornerstone of business intelligence and game theory, becomes even more significant with the ubiquity of GenAI capabilities, along with the blurring lines between human and AI agents.

On that account, decision-making templates can help establish boundaries for human and system contributions. First and foremost, AI agents should not be entitled to make judgments or commitments. Nevertheless, intelligent systems have a proven track record of applying formal and empirical reasoning with data, and GenLMs follow suit with words and nominal reasoning. By leveraging their representation through ontological prisms, these systems could be allowed to contribute to empirical (experience) and interpretative (perceptions) templates.

7.4.2 Phased vs Iterative Processes

Decision-making processes can be broadly categorized into phased and iterative approaches, the former typically represented by UMass Dartmouth's seven-step process, while the latter is epitomized by the four-step OODA loop.

7.4.2.1 Phased Decision-making Processes

Beyond lexical variations, phased decision-making processes identify 6-8 steps, as exemplified by UMass Dartmouth's 7 steps: identify the decision, gather the relevant information, identify the alternatives, weigh the evidence, choose among alternative options, take action, review the decision and its consequences (Figure 7.9).

As with any sequential approach, the UMass scheme implies a time capsule free of environmental uncertainties. Some caveats can be addressed by introducing loops that allow for (a) updated information, (b) changes in

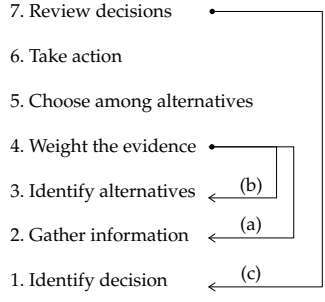


Figure 7.9 UMass decision-making process.

alternatives, or (c) adjustments and extensions of decisions. However, even so, such schemes fall short in addressing the multisided decisions of large organizations immersed in digital environments.

7.4.2.2 Iterative Decision-making Processes

Iterative decision-making models can be best represented by the OODA loop (Sharp, 2020), developed in the 60s by Colonel John Boyd, based on his experience as a fighter pilot, military strategist, and business consultant. The renewed relevance of the OODA loop comes from converging operational and technological perspectives. First, the multisided decision-making processes of enterprises immersed in competitive digital environments mirror the command and control processes of jet pilots in dogfights. Second, new technologies can be leveraged to support the critical role of data processing capabilities and their seamless, real-time integration with information systems and decision-making processes. It follows that the apparent simplicity of the OODA loop becomes a decisive advantage compared to more detailed sequential approaches, as it allows for wide-ranged, knowledge-based extensions without undermining its foundations.

The OODA loop provides a basic template for iterative decision-making processes:

- Observation: changes in business and digital environments are monitored (and their reliability assessed) at the digital (data mining) or business (business intelligence) level.
- Orientation: models and KGs are used to assess changes with regard to enterprise objectives, organization, and systems.
- Decision: policies are defined or updated, and commitments made regarding organization, systems, or platforms.
- Action: commitments are carried out, and decisions are implemented in terms of processes, functions, or applications.

At first, sequential schemes such as UMass Dartmouth's steps appear to align with Boyd's observations (steps 2, 3, and 7), orientation (steps 3 and 4), and action (steps 1 and 6) (Figure 7.10).

However, this apparent alignment overlooks key dimensions of decision-making processes, namely time, uncertainties, and multisided issues.

Phased decision-making processes are designed to freeze time and replace it with structured arrangements, ensuring that events can only occur when called for. This approach may be effective when issues are set within bounded spaces and time frames, allowing decisions to be enacted fully and without delay. However, when the arrow of time is taken into account, information follows suit, reshaping the gathering and weighting of alternatives. The consequences become compounded when multiple parties are involved, even without adversarial intent.

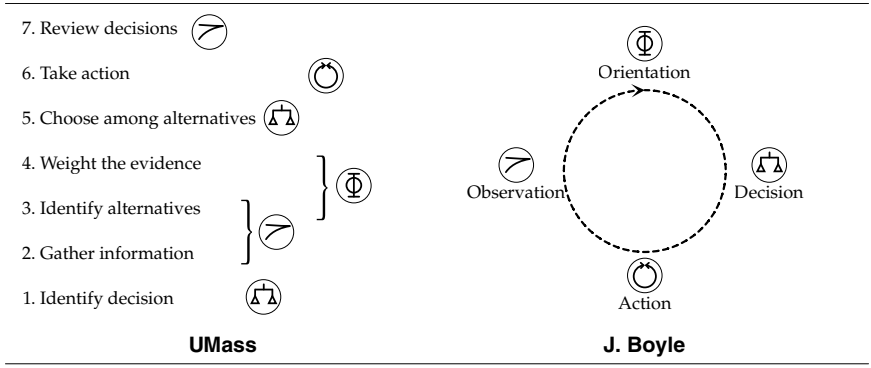


Figure 7.10 A mapping of typical phased (left) and iterative (right) decision-making processes.

In contrast, iterative schemes like OODA enable the integration of data capture and elicitation with command and control – an integration that can now be enhanced by AI and machine learning technologies. Still, this integration remains limited when operated with flat 2D data models that ignore the dynamics of changes affecting data, information, and knowledge. These limitations can be overcome if OODA steps are paired with ontological prisms.

7.4.2.3 Decisions, Time, Knowledge

For all intents and purposes, business decisions are bets against time and uncertainties. On that account, self-contained decisions are of limited use because they make no room for the continuous and combined impacts on the arguments and opportunities governing decision-making. The arguments for decisions, i.e., data, information, and knowledge, are weighted by uncertainties; the opportunity of decisions is weighted by time, namely the last responsible moment, at which point further delays reduce the value of the alternatives. While arguments and last responsible moments are specific to contexts and concerns, pairing decision-making with ontologies could set forth and make explicit the ways changes can affect decision-making (Figure 7.11).

A semblance of an iterative process could thus be obtained, and time-induced changes could be taken into account by repeating sequences, with steps being executed wholly, partially, or not at all, depending on the status of ontological resources. However, such a scheme would not allow for the feedback between facts, concepts, and categories, and consequently for a fine-grained tuning of overlapping commitments.

Compared to the attachment of phased steps to ontological realms, the pairing of OODA loops allows for a refinement of decision-making processes that

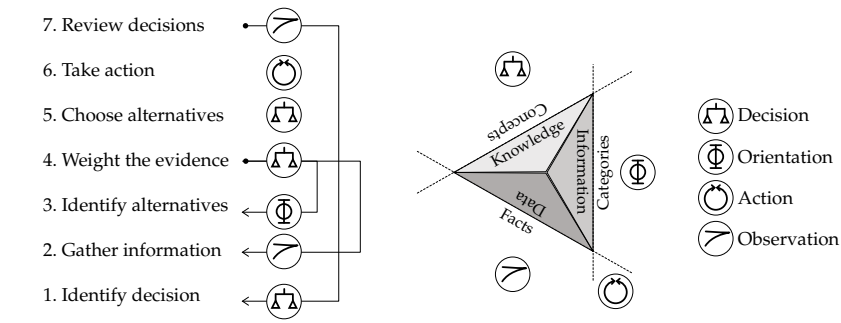


Figure 7.11 Pairing decision-making steps with ontologies.

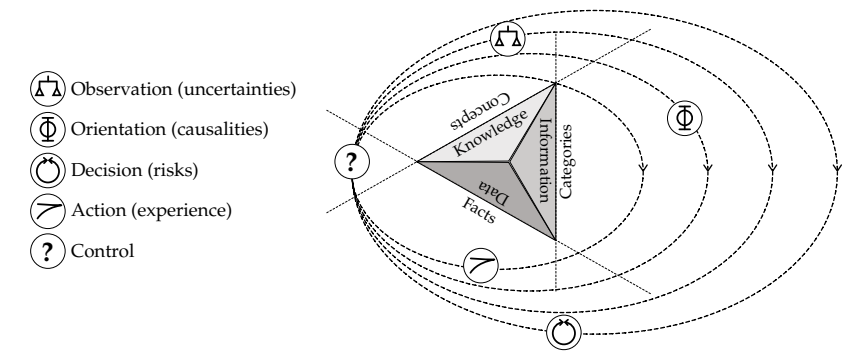


Figure 7.12 Knowledge-based OODA loops.

takes into account the nature and purpose of cognitive operations: observation, to assess uncertainties (data); orientation, to identify causalities (information); decision, to weigh risks (knowledge); and action, to learn from experience (data again) (Figure 7.12).

This paves the way to the integration of decision-making and knowledge-driven architectures.

7.5 Agentic Decision-making

This concluding section uses ontological prisms to revisit the visionary approaches of John Boyd's OODA and Stafford Beer's viable system model, demonstrating their new relevance for knowledge-driven enterprise governance and more generally agentic AI architectures.

7.5.1 Knowledge-based Decision-making

Pairing decision-making processes with both exogenous and endogenous changes in supporting data, information, and knowledge can be a game-changer for enterprises competing in digital environments. On the one end, such processes can take advantage of machine learning technologies and KGs to ensure a seamless integration with raw data resources and information assets, respectively. On the other end, fine-grained decisions can be tuned in relation to perceptions (observations/orientations), projections (orientations/decisions), commitments and accountability (decisions/actions), and experience (actions/observations) (Figure 7.13).

Assuming well-defined problem and solution spaces, contributing parties, and time frames, iterations will progress through the OODA stages based on their purposes and evolving conditions.

7.5.2 Observation

7.5.2.1 Purpose

The aim of OODA observation is to gather data about relevant facts and analyze it in light of managed categories (information) and business objectives (knowledge). Perceptions can be descriptive, focusing on environments, or prescriptive, concentrating on alternative options. In any case, the main challenge is to address massive and unstructured flows of data, which requires trade-offs between excessive complexity and the overfitting of explanatory variables and datasets.

To that end, perception can rely on four main categories of resources:

- Datasets collected from environments (data mining) or operations (process mining); these datasets can be exhaustive (populations) or selective (samples).
- Aggregate datasets and/or measurements obtained from external sources.
- Managed datasets (information) obtained through applications.
- Direct observations of external events reflecting changes in environments.

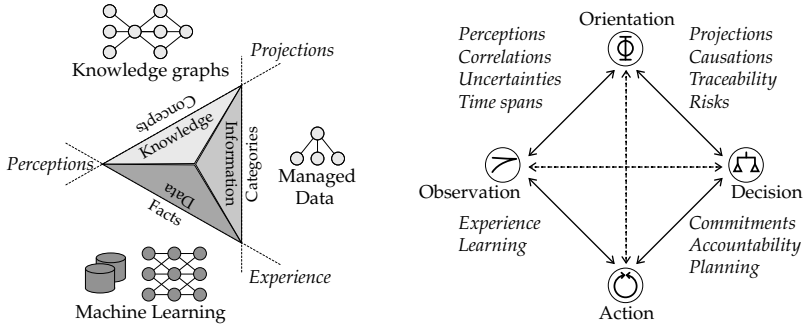


Figure 7.13 Overview of knowledge-based decision-making.

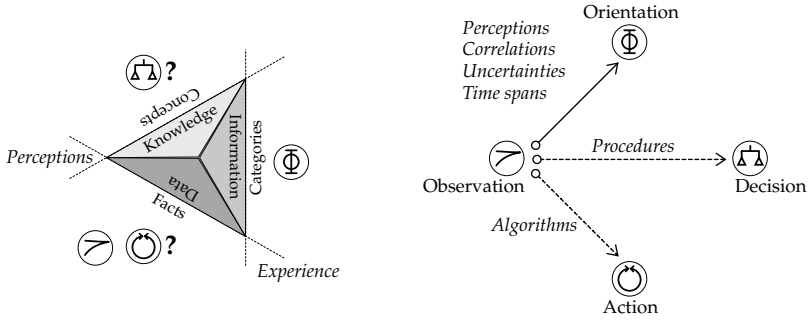


Figure 7.14 OODA: observation.

In OODA loops, observation can be conducted on both resources (perception) and outcomes (experience), which means that data can be improved on the fly, enhancing their relevance with regard to current models (information) and objectives (knowledge). In contrast, metadata that documents the provenance of resources (where, when, and how they were obtained, and who the provider is) is intended to be managed independently of the OODA loops (Figure 7.14).

7.5.2.2 Transitions

Regular advances from observation to orientation are conditioned by perceptions, in particular:

- A comprehensive assessment of resources available with regard to relevance and reliability.
- The discovery of correlations between alternative options.
- The establishment of timelines that take into account business constraints as well as data uncertainties and temporality.

In addition to regular advances in orientation, the integration of machine learning (implicit knowledge) and KG (explicit knowledge) technologies enables event-triggered shortcuts that allow for planned transitions from observation to decisions based on "canned" orientation, or for algorithmic transitions to automatized actions.

7.5.3 Orientation

7.5.3.1 Purpose

In contrast to observation, which is focused outward on data from environments, orientation looks inward, comparing perceptions, intents, and symbolic representations, the aim being to establish the rationales for projections. To that end, given

relevant and reliable observations, orientation will map targeted environments, set time frames, and build causal chains.

7.5.3.1.1 Horizons

Taking cues from travelers, maps can serve two purposes: sightseeing (perception) or planning (projection). While only the latter is directly involved in decision-making, in both cases, orientation begins with identifying territories and the relevant maps (managed information). Then, as seen above, decisions can be set in different time frames: operational, tactical, and strategic, and causal chains explored accordingly:

- Settled orientation (constant horizon): causal chains built on reliable observations (data), stable models (information), and well-defined business options (knowledge).
- Shifting orientation (change to broader horizons): causal chains built on less reliable observations or changes in models and/or business options.
- Subsumed orientation (change to shorter horizons): causal chains built on more reliable observations and/or a well-defined subset of business options.

The objective is then to ensure the interoperability of orientations (and consequently decisions) across time frames.

7.5.3.1.2 Modalities

Conditional (or Bayes) probabilities are the nuts and bolts of orientation as they are used to build causal chains from observations, assumptions, and decisions. Orientation can also take into account epistemic modalities, e.g., when decisions involve alternative options or competitors' decisions. Conditional probabilities and epistemic modalities are often mixed with business rules and uniformly represented by semantic networks or conceptual graphs. To take a typical case of business and epistemic confusion: an actual vehicle, a car design, a concept car, a digital car in a video, a car character in a story. Assuming that instances of each of these can exist in parallel yet connected realms, their undifferentiated (or flattened) representation through nodes and edges will produce an exponential complexity that prevents effective orientation; this is not the case with ontological prisms (Figure 7.15).

7.5.3.2 Transitions

Regular advances from orientation to decision are conditioned by projections, in particular:

- A mapping of causal chains that could ensure the explainability of the commitments under consideration.

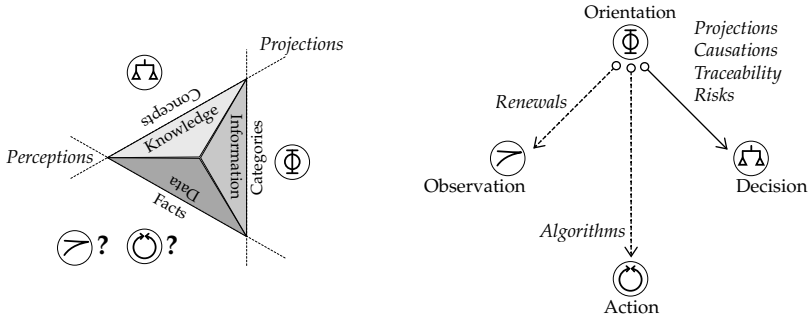


Figure 7.15 OODA: orientation.

- The identification of agents meant to be accountable for the commitments under consideration.
- The definition of swimlanes and constraints on the commitments under consideration.

Besides regular advances to decisions, renewed observations can be needed in support of changes in causal chains and/or timelines. Alternatively, paused decisions could be directly activated through algorithms.

7.5.4 Decision

7.5.4.1 Purpose

The aim of OODA decision is to define the courses of action associated with the decisions under consideration, namely, commitments and accountability, timelines and risks, and planning and modus operandi.

As emphasized earlier, our focus is on collective decision-making, which, in addition to collective reasoning and comprehension (see Chapter 6), entails explainability and collective commitments. The first objective is to draw a distinction between administrative (or pseudo) decisions and actual decisions. The former can be fully explained by observations, rules, and reasoning, while the latter combines reasoning, rules, and judgment. This distinction is particularly important for organizations immersed in digital environments permeated by AI and machine learning technologies, as it marks the boundary between options that, assuming explainability, can be left to systems and actual decisions that require commitment and accountability. Furthermore, taking into account unavoidable cross-dependencies and misaligned time spans of decisions in large organizations, individual commitments will need to be reset at a collective level, and accountabilities redefined accordingly.

Once observation and orientation have done the groundwork, sorting out options and timescales, decision-making is driven by uncertainties, the aim being to maximize returns and minimize risks. Assuming that returns are addressed separately by the business parties concerned, the question is how to take risks into account. A basic approach will be to align risks with horizons:

- Operational risks stem from implementation challenges; they can be reduced through more support training (action).
- Tactical risks stem from insufficient information; improvements can be achieved through additional observations, assuming decisions could be postponed.
- Strategic risks stem from a lack of visibility, when commitments must be made without reliable causal chains for the time frame considered; reducing time frames (shorter horizons) can improve visibility, along with risk-management schemes covering for ill-fated turns of events.

Finally, given cross-dependencies and misaligned time spans of commitments, detailed planning and modus operandi must take into account both exogenous and endogenous events. Regarding exogenous events, planned actions can be aligned with horizons and risks (Figure 7.16).

Deterministic planning (operational risks) is used for situations where unambiguous exogenous events can be directly attributed to relevant information and utilized as knowledge. In such cases, planning can be formalized into established procedures.

Stochastic planning (tactical risks) applies to situations where exogenous events involve random variables, and their potential range can be accounted for by existing symbolic representations. This type of planning can continue as long as the reliability of observations can be improved or until the "last responsible

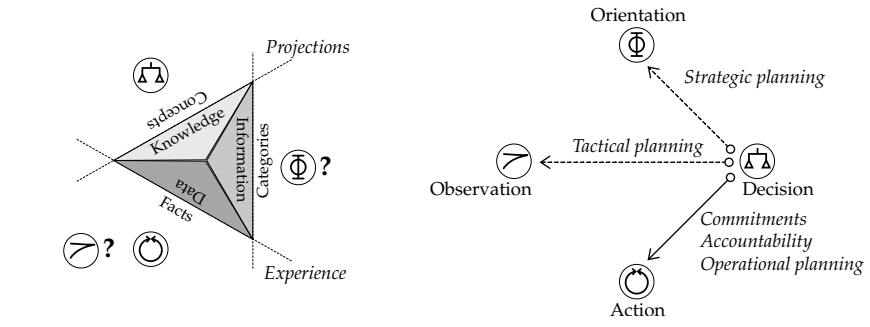


Figure 7.16 OODA: decision.

moment," when delaying commitments would reduce the range of options or expected benefits.

Strategic planning (strategic risks) pertains to situations where exogenous events may involve unbounded developments that may or may not be accounted for by existing symbolic representations. This type of planning can persist as long as risk management schemes can cover ill-fated turns of events.

Assuming that routine resource planning is being addressed separately, the generalization of GenAI capabilities is creating a new type of endogenous risk directly associated with decision-making. In this context, the European Union's regulatory initiative categorizes the risks associated with the use of AI on a four-level scale (minimal, limited, high, unacceptable) depending on the nature of the decisions involved. However, this approach may be doubly misguided: first, because it relies on administrative distinctions of activities that are becoming redundant due to the digital integration of organizations and processes, as well as the ubiquity of AI technologies; and second, because even the terms employed (e.g., "adequate," "high" quality or level, "detailed," "necessary," "clear," or "appropriate") are ambiguous.

With ontological prisms, instead of relying on arbitrary or vague qualifications, the risks arising from poor decision designs could be framed by the nature of the cognitive capacities involved and the reliability of the corresponding resources: observation and experience, reasoning and information, judgment and knowledge.

7.5.4.2 Transitions

In principle, advances from decision to action should mark definite commitments; in practice, this is a challenging ambition given the complexity of cross-dependencies and overlapping time spans. Hence, the benefit of using decision-making templates is that they help frame actions according to spaces, agents, and time frames. Not by chance, this principle coincides with Aristotelian rules regarding the unity of action, unity of place, and unity of time. In our case, actions should be designed to ensure that commitments are assigned to one agent, can be carried out within the same solution space, and are set along the same timeline.

Until conditions for regular advance to action can be fully met, decision-making processes can go backward to observation and/or orientation in order to reassess perceptions and projections.

7.5.5 Action

7.5.5.1 Purpose

As part and parcel of the OODA decision-making process, action pertains to what should be done when plans meet reality, i.e., when commitments are carried out, and experience is gained.

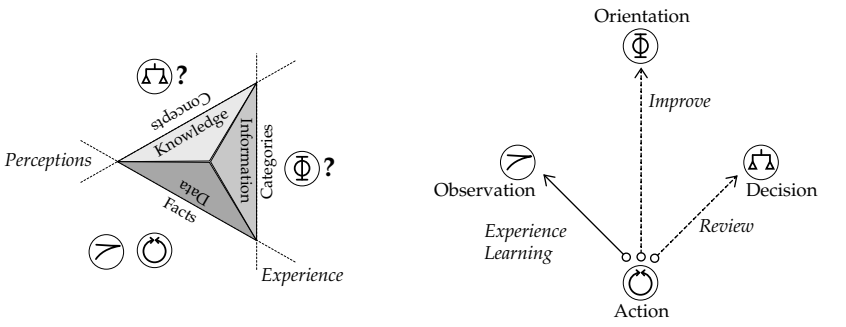


Figure 7.17 OODA: action and follow up (osmosis and homeostasis).

The first objective is therefore to monitor execution, which, with ontological prisms, can best be achieved through osmosis and homeostasis (see Chapter 4).

Osmosis is the exchange of data between systems, for data managed as information, and their operational environment. Given the inherent porosity of digital boundaries, applying machine learning technologies, such as data and process mining, to the mapping of individual facts (as reflected by data) to categories can enable fine-grained tuning of operations.

Homeostasis is the ability of organizations and systems to align their governance with changes in their business or digital environments. With ontological prisms, homeostasis can be achieved through domains that ensure the interoperability of conceptual knowledge and managed categories. Combined with osmosis, homeostasis allows for the alignment of planned (endogenous) actions with emerging (exogenous) events.

Assuming a digital integration of operations, organization, and communication, experience can be turned into a key vector through which practical (implicit) know-how can be transformed into explicit (symbolic) knowledge, learning curves can be developed through the dual interoperability of individual and collective knowledge, as well as of organization and system (Figure 7.17).

7.5.5.2 Transitions

Regular transitions from action to observation are triggered by expected completion or the need for data updates. While still in action, returns to orientation or decision may improve perceptions or projections, respectively. Full reverses would represent exceptions, justified by exogenous changes affecting Aristotelian unity rules regarding agents, spaces, or time frames.

7.5.6 Assessment: Learning Curves and Complexity Management

7.5.6.1 The Brain of the Firm

As already mentioned, Stafford Beer's vision of the "Brain of the Firm" is coming to fruition for enterprises competing in digital ecosystems. This significance is amplified when decision-making processes must combine emerging factors with planned pursuits. This duality of corporate decision-making can be neatly aligned with the levels established by Nobel Prize winners A. Tversky and D. Kahneman:

1. The first level "operates automatically and quickly, with little or no effort and no sense of voluntary control." It's put into use when actual situations must be assessed and decisions taken immediately.
2. The second level "allocates attention to the effortful mental activities that demand it, including complex computations." It's put into use when situations can be assessed with regard to past experience in order to support informed decision-making.

That two-pronged cognitive view of decision-making can be materialized through the pairing of ontological prisms and the OODA loop. Set within the broader cybernetics perspective introduced in Chapter 4, immediate decisions rely on the osmosis between operations and supporting systems, while mediated (or thoughtful) decisions are supported by the homeostasis that weaves together organization and knowledge (Figure 7.18).

Balancing emerging options with planned ones relies on layered communication combining terms used in the environment (problem) and concepts used in knowledge spaces.

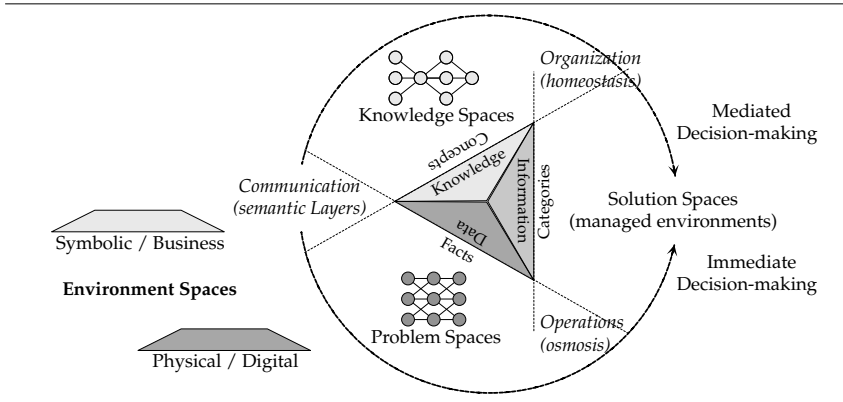


Figure 7.18 Assessment.

7.5.6.2 Learning Curves

Weaving decision-making and knowledge processing can clearly reinforce collective learning capabilities, and traceability will facilitate the development of smooth learning curves. Moreover, mapping decision-making steps with relevant cognitive capabilities will enhance the transparency of human and system contributions to observation, reasoning, judgments, and commitments.

7.5.6.3 Spaces and Complexity

As explained in Chapter 4, while managing complexity can be seen as the epitome of enterprise governance, measuring complexity remains a significant challenge. Nonetheless, ontological prisms may enable a pragmatic assessment of changes in complexity, providing a yardstick for decision-making processes.

Decision-making processes can be defined by paths from problem to solution spaces, the former set in symbolic or nonsymbolic (physical or digital) environments, the latter in a managed environment. From problem spaces, decision-making processes can go directly (counterclockwise) to solution spaces set in managed environments, or indirectly (clockwise) through knowledge spaces. As seen in Chapter 4, direct paths are determined by the extensional complexity of problems, and their effectiveness can be assessed by the induced changes in complexity. These changes can be measured using established metrics on database schemas. Additionally, the assessment of indirect paths will also take into account the induced changes in responsibilities and semantics, which could be measured using established metrics on networks (semantics) and graphs (knowledge).

7.5.6.4 Complexity and Entropy

Entropy is the blind spot of complexity, the part of observations that cannot be categorized by taxonomies or borne out by intent or purpose. As such, limiting entropy can serve as a guiding rod to decision-making profiles:

- The first objective should be to contain, as far as possible, complexity to an extensional dimension by segmenting problem spaces in order to define subspaces that could be directly mapped to solution spaces.
- Then, the design of decision-making processes should prioritize adjustments in semantic layers as defined in Chapter 5, e.g., through the tuning of GenLMs interfaces.
- Finally, changes in responsibilities, and, consequently, organizational complexity should aim to differentiate and organize decisions according to their statutory level (personal, functional, or organizational) and nature (observation, reasoning, judgment).

Broader assessments could be further achieved by harnessing ontological modalities (applied to problem spaces) and decision-making templates (applied to solution spaces), paving the way to decision-making patterns.

Key Points

This chapter concludes Part Two, which focuses on the functions of ontological prisms, with decision-making following the chapter on reasoning. The preamble has outlined the intrinsic dimensions of decision-making (issues, parties, processes, problem and solution spaces, and time frames), as well as the supporting environments.

Core issues have been introduced through the categorization of decisions in terms of problem and solution spaces framed by ontological prisms. This has led to decision-making templates characterized by processes (one-sided, collaborative, adversarial) and the predictability of environments (deterministic, stochastic, strategic).

Next, we introduced the OODA loop and explained how it can be used to integrate iterative decision-making processes with knowledge management, a key component of the agentic AI collaborative architecture.

We conclude with the assessment of the proposed approach with regard to learning curves and complexity management.

Part III

8

Knowledge-based System Engineering

I am not interested in constructing a building, so much as in having a perspicuous view of the foundations of possible buildings

Ludwig Wittgenstein

This chapter explores the engineering benefits of ontological prisms (OPs) for enterprises embedded in digital ecosystems. The first part examines the erosion of traditional architectural boundaries and the blurring of distinctions between business processes (BPs), organizations, and supporting systems. This issue is addressed by combining use cases (UCs), a well-established requirements approach, with OPs. Next, we consider the consequences of pervasive artificial and generative intelligence technologies on the implementation of BPs, particularly regarding the distinction between algorithms and decision-making. We then demonstrate how the outcomes presented in the previous chapters concerning language, reasoning, and decision-making can be utilized to build knowledge-based engineering processes that ensure absolute transparency regarding the respective contributions of systems and people to BPs. We conclude by framing quality issues through OPs.

8.1 Systems and Organizations

8.1.1 ABC of Enterprise Architecture

So much has been said about machine learning (ML), but what about systems and organizations? Taking cues from Spinoza, organizations can learn in three ways: through the senses (including common sense, experience, and beliefs), reason, and judgment. That understanding aligns neatly with the perspectives of OPs, which combine facts (for the senses), categories (for reason), and concepts (for judgment). As for the machines themselves, we can draw from

Stafford Beer's pioneering attempt to bring cybernetics to management with his viable system model. Today, Beer's vision, freed from the technological impediments of his time, is gaining a new relevance for enterprises characterized by actionable architectures, brainy capabilities, and collaborative as well as competitive behaviors.

8.1.1.1 Actionable

For enterprises competing in digital environments, change is the rule of the game, and their architectures must undertake emerging as well as planned evolutions, driven, respectively, by unforeseen and/or forced changes, and designed ones reflecting the objectives and visions of their stakeholders. To that end, enterprises can use OPs to turn their architectures into digital twins supporting a comprehensive transparency and interoperability of all physical, digital, and symbolic components.

8.1.1.2 Brainy

Viable systems must ensure their sustainability in changing environments, which entails balancing extensional assessments (which improve with time until becoming redundant) and intensional commitments (which bear the costs of missed opportunities if kept waiting too long). Such abilities, as described in the previous chapters, can be best achieved using OPs to integrate system and knowledge engineering.

8.1.1.3 Collaborative and Competitive

As part of ecosystems, enterprises are expected to communicate with other agents, whether to collaborate or compete. Moreover, the lines between friends and foes in business are often more dotted than solid, and these dots may change swiftly. This entails different semantic layers depending on the parties involved and the communication channels used: executable languages (digital devices), actionable languages (symbolic systems), or natural languages (people).

8.1.2 Engineering and Digital Immersion

8.1.2.1 Change Is the New Normal

In principle, enterprise architectures (EAs) combine purely symbolic systems (organizations) with embodied ones (computers), the latter being a mix of symbolic and physical artifacts. In practice, the digital immersion of EAs has rendered these distinctions largely irrelevant, entailing holistic and continuous changes in both the symbolic and physical realms. However, while systems can be partially and/or temporarily paused for maintenance, refactoring,

or extension, it's not the case for EAs as a whole, which cannot be discontinued, let alone dismantled and reconstructed. Hence the conundrum.

At first glance, changes to systems, such as engineering projects, can be compared to database transactions that are subject to constraints regarding atomicity (the all-or-nothing rule), consistency (with respect to context), isolation (in relation to other transactions), and durability (of changes once committed). However, there is a critical difference: isolation and durability cannot be applied to projects in the same way they are applied to transactions due to their context: systems for the former, business environment for the latter.

8.1.2.2 Isolation and Interoperability

In a 2D flat representation model, transaction isolation is intended to secure the consistency of managed data (i.e., surrogates). Applying the same principle to ensure the consistency of applications would require a corresponding representation of applications at the architectural level; otherwise, it would only be feasible for standalone applications with a limited scope and/or time span. This limitation would significantly hinder the changeability of architectures.

3D OPs allow for a distinction between representation levels, typically databases or EA, and consequently enable local isolation and global interoperability (see Chapter 3). On one hand, by taking advantage of the intrinsic ontological and functional distinctions, the isolation of intrinsic components can be secured while fine-grained functional changes are implemented. On the other hand, ontological modalities can be used to build heterogeneous representations of actual, expected, or planned configurations.

8.1.2.3 Sustainability and Time Frames

Compared to database transactions designed to be carried out within time capsules, the engineering of systems is governed by two different time frames: one set by the business environment and the other by the organization itself. It follows that the durability constraint does not pertain to committed changes (set at the system level) but to committed requirements (set at the organizational level). This conundrum is epitomized by alternative engineering processes:

- Iterative processes, characterized by time capsules, changing requirements, and frozen business context and organization.
- Phased processes, characterized by time scales, frozen requirements, and planned changes in business context and organization.

Once stand-alone objectives are established and addressed through iterative processes, model-based system engineering (MBSE) may, to some extent, overcome the issue of having requirements fixed upfront. However, this is only to a

certain extent if the models considered are related to system realms and engineering time frames, while overlooking business considerations. This limitation can be addressed by anchoring models to ontological realms with an explicit distinction of time frames.

8.2 Engineering Processes and Ontological Prisms

From an ontological perspective, engineering processes can be organized into three phases: requirements (natural language, direct communication), models (symbolic representation, mediated communication), and code (signs, computation). Given the growing impact of generative artificial intelligence (GenAI) technologies applied to coding, this chapter focuses on the requirements and models phases.

8.2.1 Scope: The Pagoda Blueprint

Compared to traditional 2D modeling approaches that only address the representations at the system level, ontological representations must encompass environments as well as all kinds of artifacts, physical or symbolic. Taking a cue from Stanford University's Symbolic Systems Program, such systems are: "... agents that use meaningful symbols to represent the world around them so as to communicate and generally act in the world."

This seemingly plain definition implies:

- A unified understanding of people, organizations, and systems as agents with the ability to process symbolic representations.
- A threefold modeling paradigm: for physical environments (objects and phenomena), for symbolic environments, and for the symbolic representations of environments within systems.
- A conceptual distinction between representations at enterprise and systems levels: the former, unbounded and pragmatic; the latter, finite and normalized.

On that basis, EAs can be best defined through the Pagoda blueprint (see Fannader R. (2021)), an extension of the well-known Zachman Framework (see Bibliography), redesigned to align with the dimensions of OPs:

- Enterprise level, for the symbolic representation of business concepts and organization.
- System level, for the logical representation of information and functions.
- Platform level, for the implementation of resources and operations.

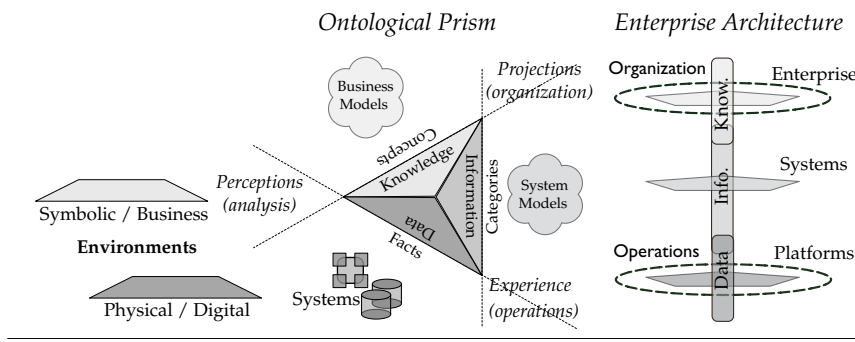


Figure 8.1 EA engineering through OPs.

That paradigm shift redefines EA along ontological tiers: facts/data, categories/information, concepts/knowledge; and language semantics: executable, actionable, and natural, respectively (Figure 8.1).

The aim of EA engineering can thus be expressed as ensuring continuous and consistent changes in perceptions (data and business analysis), projections (organization and systems), and experience (requirements and operations). As far as engineering processes are concerned, these changes can be addressed through user stories (USs) and UCs on the one hand, and models on the other hand.

8.2.2 Objectives: User Stories and Use Cases

Assuming that engineering processes pertain to systems, they are supposed to be rooted in natural language requirements, typically USs, UCs, or functional and system cases (SCs). These inception points can be neatly aligned with OPs (Figure 8.2).

It should be noted that OPs are, by design, holistic representations aimed at supporting various iterative and/or incremental processes targeting all types of architectures and systems. In line with that ecumenism, requirements should be categorized in ontological terms.

8.2.2.1 Stories

USs (Cohn 2004) are literal descriptions of actual users' operations, expressed in natural language, independent of supporting systems and development models. These stories are meant to be communicated through direct conversations, without the mediation of shared representations, and that's their underlying rationale: expressing requirements as narratives that weave together circumstances, roles, events, operations, and outcomes. Assuming shared ownership and continuous delivery, USs can be associated with iterative development

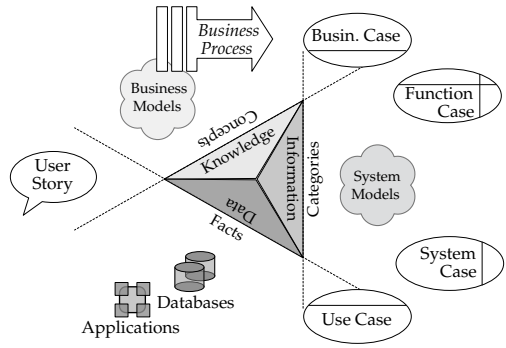


Figure 8.2 UCs and OPs.

models (broadly known as Agile) that target stand-alone applications without organizational or technical dependencies. From a broader perspective, when linked to OPs, USs can serve as semantic bridges between specific needs (facts) and business objectives (concepts).

8.2.2.2 Cases

In contrast to USs, cases pertain to designated boundaries: between environments and organizations (business cases [BCs]), between users and systems (UCs), between systems (function cases [FCs]), or between system components (SCs).

8.2.2.2.1 Use Cases

As originally defined by Ivar Jacobson [Jacobson], UCs can be seen as a de facto standard for the description of what users expect from systems. In principle, UCs consider systems as black boxes addressed from a bounded context (facts) by identified agents (actors) referencing managed instances of categories (surrogates). Yet, enterprise digital immersion and advances in generative language models (GenLMs) are reshuffling the designs of user interfaces, undermining the distinction between agents (in the environment) and applications (in systems), and more critically, the distinction between language and knowledge. These issues can be clarified using OPs to make an explicit distinction between UCs (set between facts and categories), BCs (set between concepts and categories), and FCs and SCs (set between categories).

8.2.2.2.2 Business Cases

Attempts to extend UCs with additional business concerns [Cockburn] have been hindered by the traditional 2D modeling paradigm, which does not accommodate the conceptual distinction between business and system realms, and consequently, their respective modeling semantics. Furthermore, the apparent

continuity between these realms obscures strategic issues related to the realization of business models within EAs. Compared to UCs, which are situated between users and systems, BCs with ontological perspectives are positioned between enterprises and their environments. As such, they present business stakeholders and enterprise architects with clear alternatives regarding the realization of objectives in terms of information systems as well as organizational responsibilities.

8.2.2.2.3 Function and System Cases

FCs are symbolic descriptions of exchanges between systems independently of business or technical contexts. SCs do the same within systems and are therefore dependent on technical contexts.

8.2.2.3 Models

In line with the distinction introduced between direct and mediated communication, the raison d'être of models is to support mediated communication based on shared symbolic representations. When applied to EA, models can be characterized by tiers: platforms (technical models [TMs]), systems (logical models [LMs]), or enterprise (conceptual models [CMs]) (Figure 8.3).

CMs define categories of business objects, agents, events, or activities independently of their logical representation by systems. LMs define the categories that are meant to realize CMs independently of the technical platforms used to implement them. TMs define the categories used to implement logical categories. It must be emphasized that the categories used to build these models belong to the same symbolic realm and differ only with regard to the instances represented: symbolic objects identified within the organization (conceptual), surrogates of objects identified in the environment (logical), and digital objects identified within the system (technical). In terms of model-based engineering, CMs are interpretable, LMs are actionable, and TMs are executable.

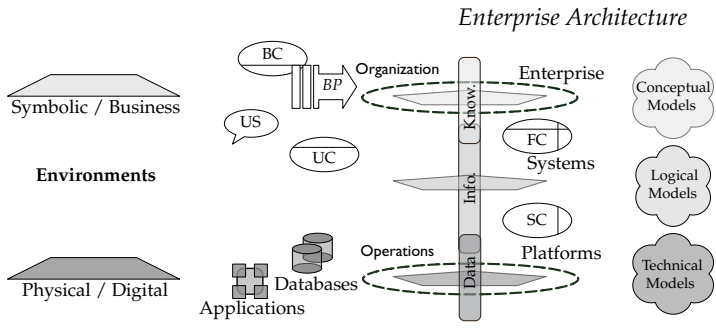


Figure 8.3 UCs and models.

8.2.3 Processes: Workflows and Remits

8.2.3.1 Workflows

EA engineering workflows are anchored to cases and models, except for stand-alone applications developed directly from user requirements, e.g., with USs (Figure 8.4).

- BCs translate business objectives into CMs.
- UCs, after taking account of CMs, merge user needs with LMs; UCs are then realized by SCs, already present or newly created.
- FCs differ from UCs as they are meant to support BPs regardless of user-specific circumstances; otherwise, they operate like UCs with regard to CMs and LMs on the one hand, SCs on the other hand.
- SCs, as FCs, do not directly reflect users' needs; they are meant to enable the effective realization of both use and FCs.

Crossing business-specific needs and shared models determines integration junctures:

- Organizational integration, for the continuous alignment of business and system models.
- Functional integration of CMs, FMs, and TMs.
- Operational integration, for the dynamic alignment of TMs with changes in physical environments

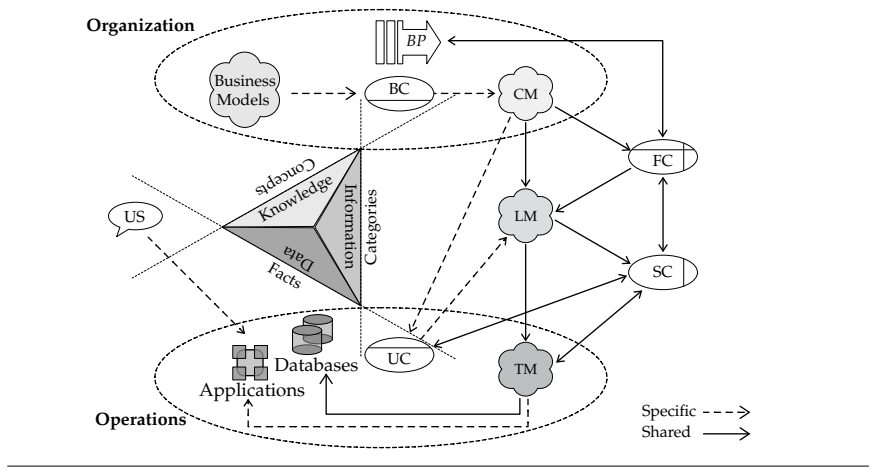


Figure 8.4 Application-specific (dotted line) and shared (solid line) workflows.

By combining the Pagoda blueprint with ontologies, EA workflows can ensure the consistency of architectural changes over time and across business, organizational, and technical tiers.

8.2.3.2 Remits and Workshops

As noted previously in Chapter 4, introducing OPs should not involve additional organizational layers, except for those specific to knowledge management when none exist. When OPs are used to support engineering, the aim should be to adopt preexisting remits and responsibilities, typically:

- Business domain managers, taking responsibility for BCs and CMs.
- Business analysts, taking responsibility for USs and UCs, taking account of BCs.
- Enterprise architects, taking responsibility for CMs, LMs, and TMs.
- System architects and quality managers, taking responsibility for the realization (through models) of FCs and SCs.
- Process managers, taking responsibility for deployment and operations.

Taking a leaf from lean manufacturing, workflow units could be handled across four engineering workshops (Figure 8.5):

- Enterprise: Where enterprise architects, system architects, and business analysts identify business objects and anchors as well as their semantics (BCs).
- Domains: Where business analysts and system architects define the corresponding representations to be managed by systems (FCs).

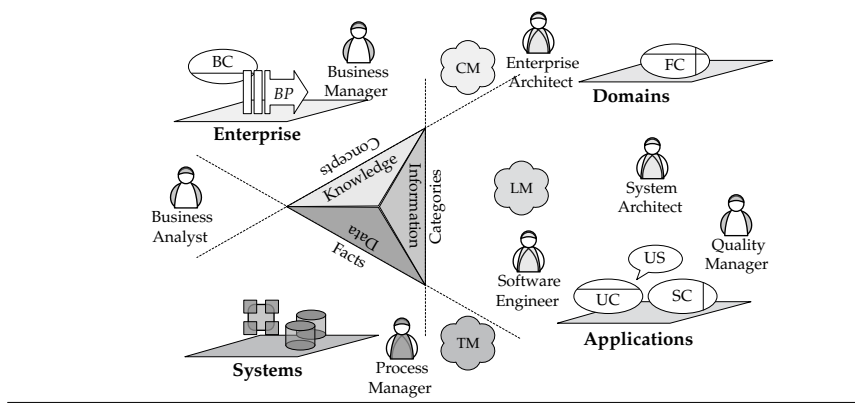


Figure 8.5 Engineering remits and workshops.

- Applications: Where business analysts and software engineers define and develop the corresponding implementations by systems (UCs and SCs).
- Systems workshop: Where software engineers and process managers define and manage deployed configurations.

Indicative remits can be added, but the specifics are best left to enterprises' organization.

8.3 Anchoring Engineering to Ontologies

Knowledge-based engineering processes can be described as a DNA-like double helix that pairs UCs with models. Anchoring both threads to OPs will ensure effective, continuous, and consistent integration of the business, organizational, and systems components of EAs.

8.3.1 Anchoring Business, Function, and Use Cases

Engineering processes start with requirements, which may or may not distinguish between concerns (typically business, functional, or technical) and/or parties concerned. Hence, the benefits of setting functional boundaries.

8.3.1.1 Use Cases

UCs are primarily identified by locations, primary actors, and triggering events (Figure 8.6a). Goals and business objects (ontological concepts) may be referenced directly or through domains and/or BPs (Figure 8.6b).

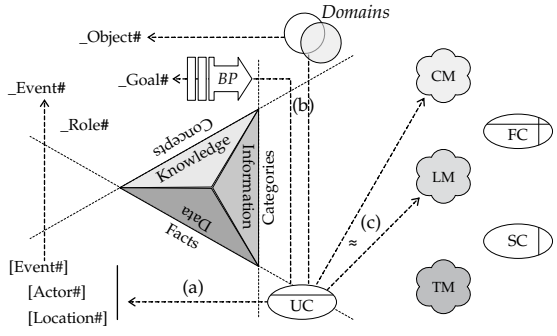


Figure 8.6 Anchoring UCs to ontological facts (a) and concepts (b).

Actors, events, and locations are meant to be identified as facts, possibly with references to intrinsic concepts for roles and business events. By contrast, UCs' view of business objects is functional, i.e., UCs know targeted domains and aspects in conceptual (CM) and/or logical (LM) models (c) but not intrinsic identities. The distinction between UCs' intrinsic and functional references enables their iterative and incremental development, first with regard to requirements and then with regard to their realization through LMs and SCs.

8.3.1.2 Business and Function Cases

Contrary to UCs, BCs and FCs must be anchored to intrinsic concepts, the former through BPs, the latter through business domains (Figure 8.7).

As UCs address what should happen between enterprises and supporting systems, BCs do likewise between enterprises and their environment. On that account, internal (enterprise-defined) events are replaced by the ones signaling changes in environments, and actors, defined in relation to BPs, are replaced by roles, defined in relation to organizations. Nevertheless, BCs are not identified by business events or organizational roles but by intents and goals. Moreover, assuming that BCs are directed at further and fuzzier horizons than UCs or domains, they cannot be tied by intrinsic references to corresponding categories. By comparison, while FCs are also set for long life spans, such spans are meant to be well defined, along with both referenced domains and LMs. Then, ontological modalities can be used to manage virtual, planned, or actual conceptual and LMs for business and FCs, respectively.

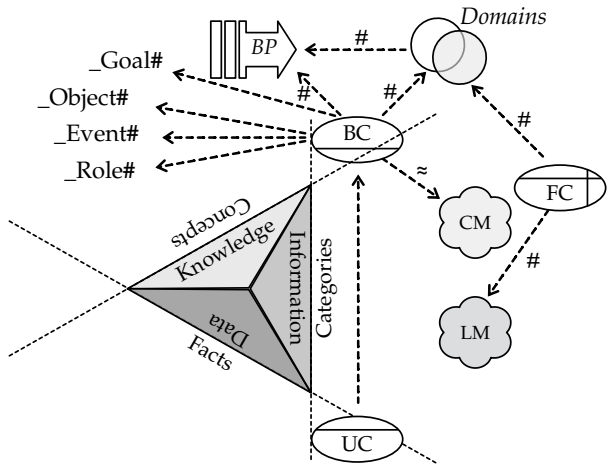


Figure 8.7 Anchoring business and FCs.

8.3.2 Anchoring Models

Models can be likened to currency, serving as a means of exchange, an instrument of measure, or a repository. Setting apart direct exchanges (barter), models can serve as a medium supporting exchanges between organizational units with different business or technical concerns, or be kept and maintained for subsequent use or reuse. In any case, models can be used to assess contents with regard to size, complexity, or quality. The distinction of roles (communication or reuse) induces a distinction with regard to engineering processes: communication puts the focus on language, reuse on contents. Regarding the organization of engineering processes, the primary distinction pertains to realms: business models for ontological concepts, and CMs, LMs, and TMs for ontological categories. In order to avoid confusion with regard to conceptual and data models, which straddle ontological realms (with data models viewed as part of TMs, both should be represented by categories (Figure 8.8).

8.3.2.1 Conceptual Models

Also known as computation-independent models (CIMs), the raison d'être of CMs is to bring under a common roof business-specific domains. As such, their main purpose is twofold: first, to enhance communication between stakeholders sharing information assets but not necessarily semantics; second, to ensure the consistency and interoperability of specific contents. With OPs, a conceptual shared architecture is built on intrinsic concepts and semantic variants derived from their functional relations.

8.3.2.2 Logical Models

Also known as platform-independent models (PIMs), the purpose of LMs is to support the implementation of the same functional and logical capabilities across different technical platforms. As such, changes in content must be subject

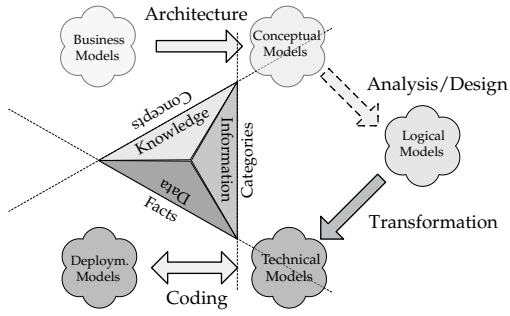


Figure 8.8 Models anchors.

to strict versioning, and the focus is placed on the trustworthiness of semantics, allowing for reliable, possibly automated, transformation from LMs to TMs.

Setting apart functional aspects, which can be obtained through data analysis, the main challenge lies in the origin of LMs, whose intrinsic content is inherently different from that of CMs. Compared to the mapping of business domains into CMs, which results from collaborative processes between organizational units, the mapping of CMs into logical ones involves intrinsically different content and, consequently, explicit analysis and design decisions. That issue, which is at the core of MBSE, can be resolved by combining OPs with the domain-driven design (DDD) approach (Section 8.5.2.1).

8.3.2.3 Technical Models

Also known as platform-specific models (PSMs), the purpose of TMs is to support the operational deployment of systems across different platforms. When integrated into model-based systems engineering (MBSE) frameworks, most activities (management or usage) related to TMs can be at least partially automated, and using ontologies would provide limited benefits to engineering frameworks.

8.3.2.4 Anchors and Semantic Layers

Capturing requirements through explicit entry points offers two major benefits. First, it distinguishes homogeneous requirements in functional and SCs, which pertain only to categories, from heterogeneous ones in business and UCs, which are situated between concepts and categories for the former, facts and categories for the latter. Then, since requirements are meant to be expressed in natural language, and based on the semantic layers introduced in Chapter 3, certain assumptions can be made upfront depending on the entry points (Figure 8.9):

- Nothing can be assumed upfront about the semantics of USs, except nominal and empirical meanings.
- Since function and SCs are fully integrated into systems, with no interfaces with the environment, one can assume that the corresponding requirements can be formally expressed.
- Conversely, since BCs are set at business and organizational levels, empiric (for domain-specific) and ontological (for projections) meanings should be the default options for corresponding requirements, possibly with formal segments.
- Requirements expressed through UCs are the most challenging because they can combine empiric and formal semantics, the former reflecting business concerns, the latter managed categories.

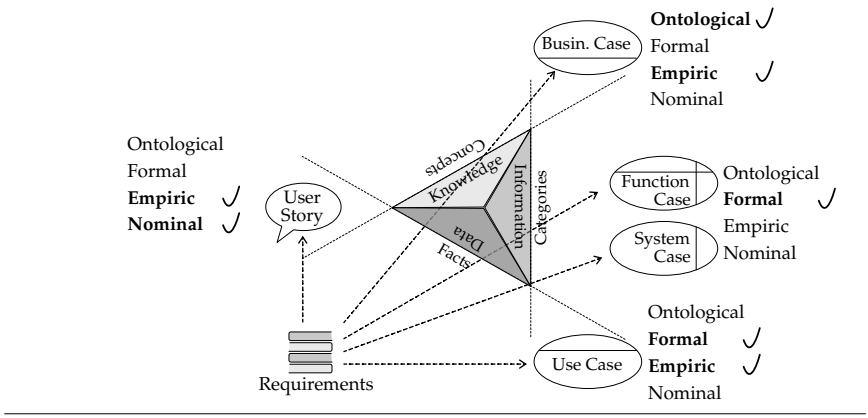


Figure 8.9 Filtering requirements according to cases and semantic layers.

Filtering requirements based on entry points, models, and semantic layers is inherently shallow, as it does not address the content itself. As a result, the requirement units obtained remain presumptive and fragmentary. Nevertheless, they lay the groundwork for in-depth and comprehensive processing of the content.

8.3.3 Engineering Loop

Knowledge-based requirements processes can be best implemented through the observation–orientation–decision–action (OODA) decision-making loop introduced in Chapter 7 (Figure 8.10):

- Observation, for the capture and elicitation of any kind of requirements, verbal or textual, structured or otherwise. It can be done through some kind of parsing (syntax, facts, data), with or without language models' assistance.
- Orientation (analysis), for the mapping of requirements to EAs. That calls for reasoning capabilities (semantics, categories, information).
- Decision, for the assignment of requirements to work units. That would rely on judgment and entail transparency and accountability.
- Action, for the development of software components.

The proposed approach distinguishes between requirements processing, which relies on the whole of ontological realms, and model-based engineering, which pertains to logical representations, i.e., categories.

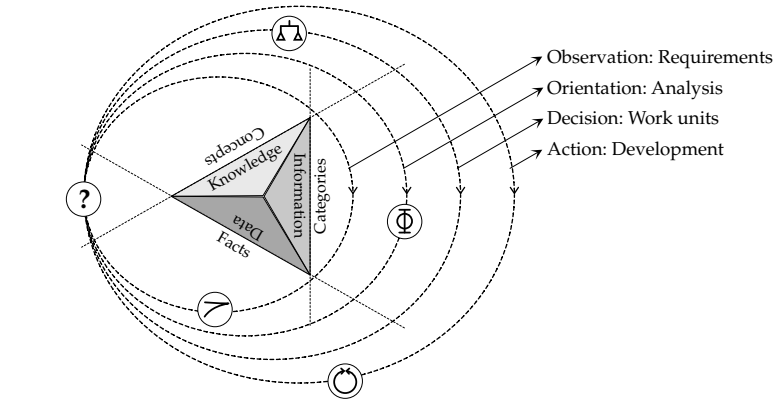


Figure 8.10 Engineering OODA loop.

8.4 Requirements with Ontological Prisms

Business, use, and functional cases have been defined as the entry points of engineering processes through which requirements are captured and elicited, while models are the symbolic representations of systems – past, present, or future. First, we consider the benefits of OPs in relation to requirements.

8.4.1 Requirements Capture and Elicitation

On the one hand, requirements mark the origin of engineering processes, and as such, nothing can be assumed upfront about their form or content: whatever happens to requirements down the road, nothing can be decided a priori. On the other hand, requirements must provide a clear, consistent, and nonambiguous basis to system analysts. That challenge can be mitigated when requirements can be earmarked upfront to business, use, functional, and SCs, because only UCs are set astride business and system realms; the business, function, and system options are focused on one side or the other of the divide. That provides the basis for an initial refining.

8.4.1.1 Capture

Given a set of unstructured and heterogeneous requirements, the primary objective is to edit them so as to allocate as many units as possible to single entry points:

- USs are the option of last resort for requirements units that pertain to facts without explicit reference to supporting systems or models (categories).

- FCs are for units that pertain to LMs (categories) without references to concrete users (facts).
- SCs are for units that pertain solely to TMs (categories).
- BCs are for units that pertain to CMs (categories) without references to concrete users (facts).
- UCs are for units that mix user actions (facts) with references to LMs (categories).

It's worth noting that such a routine is ecumenical and can be straightforward when supported by OPs.

8.4.1.2 Elicitation and Earmarking

Taking for granted that requirements can be expressed in various multimodal formats (textual, numerical, graphical, etc.), the primary objective is to parse these requirements to make them intelligible. This initial processing, commonly known as elicitation, should remain as nominal as possible so as not to encroach on requirements interpretation. Given the need for a nominal approach, GenLMs may significantly contribute to elicitation of requirements.

Using GenLMs to elicit requirements is straightforward and can be accomplished independently of additional resources. Further leveraging them for earmarking can be achieved in two steps. The first step involves training and tuning the model with documents and data resources related to the business, organization, and systems. The second step is to prompt the model to earmark requirement units based on the entry point profiles mentioned above (Figure 8.11).

While earmarks may not be final and observations can be reconsidered, they should be viewed as a necessary step before requirements analysis (OODA orientation).

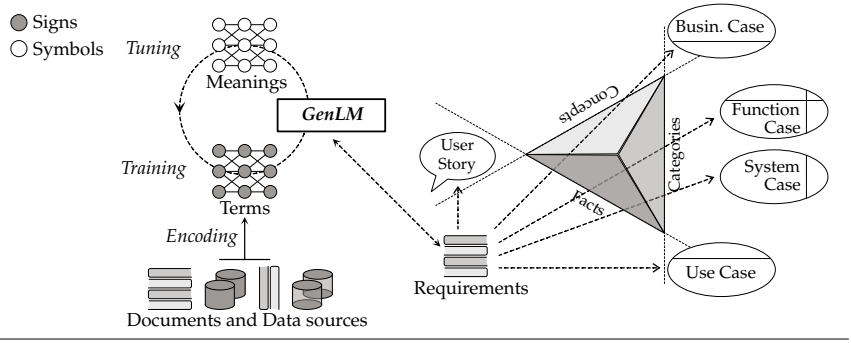


Figure 8.11 Using GenLMs to earmark requirements.

8.4.2 Requirements Analysis: Lines and Anchors

To be effective, requirements must be mapped onto footprints in EAs. At their simplest, requirements make explicit references to managed categories, allowing for a direct mapping to models. However, mapping requirements to the entire range of an enterprise's symbolic assets is generally more intricate. As a preliminary step, a context must be clearly established: on one hand, there are thesauri, directories of symbolic assets (documents, datasets), models, and ontologies; on the other hand, there are requirements management tools, with or without generative language capabilities. Requirements analysis can then proceed with footprints, anchors, granularity, and variants.

8.4.2.1 Drawing Lines

Two distinctions are often made regarding requirements: one between functional and nonfunctional requirements, and the other between UCs and business domains. Nonfunctional requirements can be characterized by a two-pronged criterion:

- Experienced by users: They can be assessed by application users, i.e., through UCs.
- Dealt with independently: They can be factored out and addressed by function or SCs.

The distinction between UCs and domains is more straightforward: the former is defined by actors and business operations, the latter by organizational units and business objects.

8.4.2.2 Anchors

Entry points determine the requirements perspective: users (stories and UCs), organizations (BCs), systems (functional cases), and subsystems (SCs). The aim of the analysis is first to anchor requirements to realms: business models (concepts), managed entities (categories), or legacy systems (facts). Then, analysts can leverage OPs interoperability to ensure the consistency of the requirements (Figure 8.12):

- Thesauri, for the alignment of terms used in requirement chunks with the meaning of concepts used in business models.
- Taxonomies, for the alignment of rules in requirements with partitions (facts) and/or schemas (categories).
- Domains, for the alignment of activities in requirements with the responsibilities of business and/or systems organizational units.

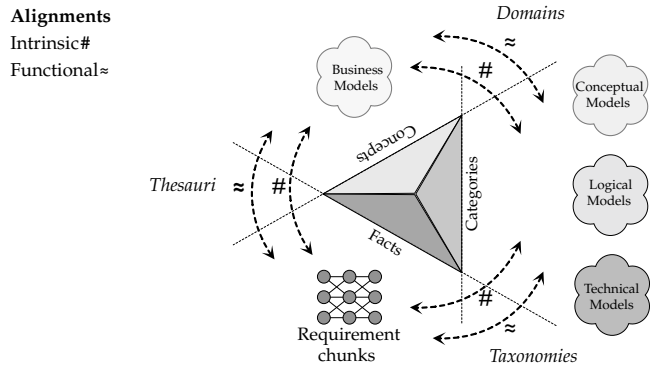


Figure 8.12 Anchoring requirement to ontologies.

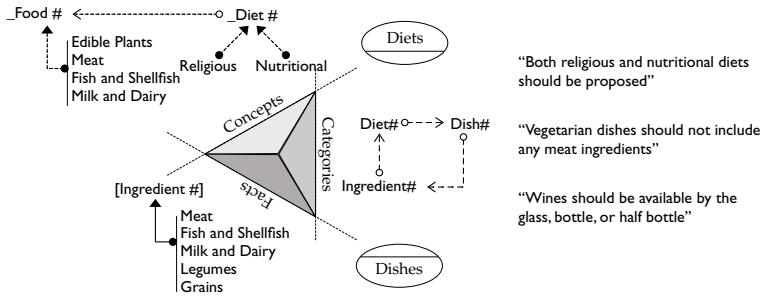


Figure 8.13 Use case (Dishes) and business case (Diets) requirements (notations in Annex).

Considering use and BCs, requirements are meant to be rooted in facts and concepts, respectively, and to aim at new categories and/or align with legacy ones. Taking into account intrinsic anchors (#) and functional relations or properties (≈), the mapping of requirements can thus produce three kinds of outcomes (Figure 8.13):

- Requirement units can be anchored to manage intrinsic categories. For instance, the statement in use case Dishes "Vegetarian dishes should not include any meat ingredients" can be satisfied by adding a subtype to the Dish# category with a functional constraint on the ingredient property.
- Requirement units can only be functionally associated with intrinsic managed categories. For example, the statement in the business case for diets, "Both religious and nutritional diets should be proposed," necessitates

two additional intrinsic distinctions: one between nutritional and religious diets, and another between food and ingredients.
- Requirement units cannot be intrinsically anchored or functionally associated with any managed categories. For instance, the statement "Wines should be available by the glass, bottle, or half bottle" requires the creation of a new category for Wine#, along with a new domain.

Anchoring requirements to symbolic representations only goes halfway and should not preempt an explicit analysis of all available options. For instance, the vegetarian requirement could be represented by a condition on the relationship, a powertype, or a subtype. To that end, analysts must consider the granularity of the requirement units.

8.4.3 Requirements Analysis: Granularity and Variants

8.4.3.1 Requirements Granularity and Aristotle's Classical Three Unities of Drama

As commonly agreed by BP modeling methods, requirements should be elicited into single units that cannot be broken down into further levels of activity. Such a cursory definition falls short of addressing architectural concerns because it ignores the critical distinction between business activities and the way they are meant to be executed by supporting systems. That's not an issue for functional and SCs because they are fully defined within the information realm, or for BCs, which can be defined in conceptual terms. However, something more thorough is clearly needed for UCs that are set across heterogeneous realms. Aristotle's classical three unities of drama may fit the bill.

Unity of action (process): Requirement units should have one main course of action initiated by a single external event, directly or through a primary actor, with a clear description of the intended course of action, with all necessary resources identified. Subplots, if any, can involve the collaboration of secondary actors, but they should return to the controlling main plot once completed. This condition ensures that all possible dependencies can be identified.

Unity of place (location): Requirement units are intended to be fulfilled within a single address space (either physical or symbolic), where all resources can be accessed directly and immediately, eliminating the need to rely on communication capabilities between locations. This is essential for defining and managing functional architectures independently of technical ones.

Unity of time (event): Requirement units must be timed in relation to a single time scale within which operations can be synchronized. This is essential for managing the inner workings of business logic and the synchronization of BPs independently of communication channels and mechanisms.

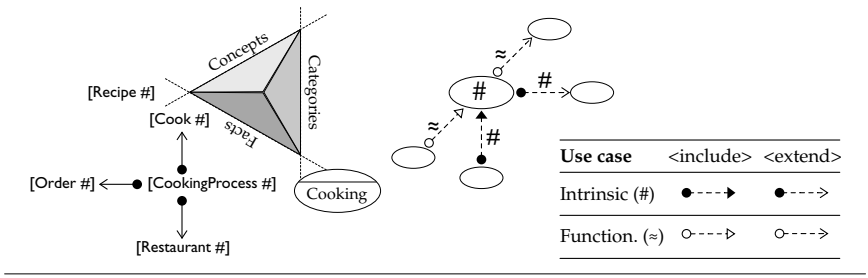


Figure 8.14 Use case and Aristotle's three unities for action (recipe), time (order), and place (restaurant).

Applying Aristotle's rule to the definition of UCs can be achieved through actors, address spaces, and triggering events, which are identified in physical (for time and place) and symbolic (for the description of activities) environments. For instance, a cooking use case would be identified by a recipe (process description), the restaurant (place), and a cooking time span initiated by the order (event), all with intrinsic identities (Figure 8.14).

The granularity of requirement units obtained by applying Aristotle's three unities rule is translated into UCs through established <include> and <extend> connectors (Jacobson). It must be emphasized that UCs are not categories, which means that inheritance semantics cannot be directly applied to <include> and <extend> connectors. Instead, since UCs are identified in actual environments, whether physical or symbolic, the semantics of these connectors should be aligned with the semantics of fact connectors, namely parthood for <include> and association for <extend>, with intrinsic (#) and functional (≈) markers for existential constraints. Still, given that use case diagrams combine UCs with function and SCs, we maintain the standard dotted notation.

While applying the Aristotle principle to UCs may seem like a complex undertaking, it offers significant guarantees regarding their consistency. This can be illustrated with our simple cooking example. Assuming the use case is defined by the restaurant (unity of place), a recipe (unity of action), and an order (unity of time), the objective is to represent the variants in a way that supports consistency checks (Figure 8.15).

Regarding **action**, <include> connectors relate to a subset of activities (or subplots), intrinsic connectors for mandatory activities (e.g., preparing ingredients), and functional connectors for optional inclusions (e.g., cooking the ingredients). Alternatively, <extend> connectors relate to contributing activities, with intrinsic connectors for necessary contributions (e.g., serving) and functional connectors for activities that can be performed independently (e.g., preparing dressing).

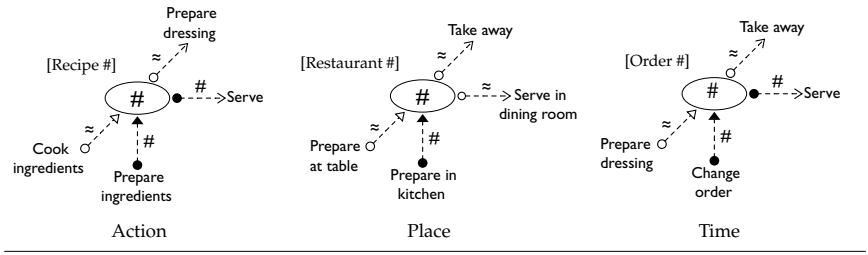


Figure 8.15 Granularity and UCs.

Regarding **place**, <include> connectors relate to subsets of space, intrinsic connectors for exclusive spaces (e.g., kitchen to prepare ingredients), functional connectors for nonexclusive spaces (e.g., cooking done at the diner's table). Alternatively, <extend> connectors relate to different spaces, intrinsic connectors for places allowing for direct communication (e.g., serve in dining room), functional ones otherwise (e.g., take away).

Regarding **time**, <include> connectors relate to controlled execution, intrinsic connectors for synchronous execution (e.g., change order), and functional connectors for asynchronous execution (e.g., prepare dressing). Alternatively, <extend> connectors relate to different execution time frames, intrinsic connectors for synchronous execution (e.g., serve in dining room), and functional connectors for asynchronous execution (e.g., take away).

Using the Aristotle principle to define UCs' building blocks allows for fine-grained consistency checks. For instance, preparing dressing appears as a functional subset with regard to the unity of action, and as a functional associate with regard to time. Conversely, functional connectors are used with regard to both space and time of the take-away variant; given the intrinsic dependency between space and time dimensions, any connectors discrepancy would raise questions.

8.4.3.2 Object Variants, Powertypes, Delegation

Requirements often arrive as they are, rarely aligned with ontological intrinsic anchors or Aristotle's three unities. When addressing these issues, one must disentangle the mesh of variants and rules that constitute the alternative embodiments of complexity: the former is declarative, while the latter is procedural. The objective of the analysis is to determine how to make the best use of each and minimize complexity.

As discussed in Chapter 2, variants can be organized through hierarchies and associations that rely on a common syntax but different semantics for facts, concepts, or categories. By applying uniform syntax across realm-specific semantics, the aim is to structure requirement variants into intrinsic hierarchies. Assuming

that this intrinsic backbone can be preserved, the focus can shift to whether functional (nonintrinsic) hierarchies should be maintained as they are or translated into associations. This issue, known as functional dependencies in relational models, can be addressed through powertypes.

8.4.3.2.1 Powertypes

Powertypes (see Chapter 2) are types whose instances represent collections (or partitions) of instances of other types. They are introduced to represent sets of individuals that do not qualify as categories on their own, i.e., categories whose instances are meant to be managed as surrogates. Given the syntactic neutrality of collections and aspects, partitions and powertypes can be used interchangeably across ontological realms, allowing for the translation of variants from facts to categories (Figure 8.16).

Starting with a nondescript fact, such as wine, and without any knowledge of dependencies, the first question is whether some wines (subsets) have specific properties (aspects) in common or share the same values. Let's consider wine years; while this may clearly be viewed as an intrinsic property, its value is not necessarily known for each wine. In our example (Figure 8.16), the choice is to use an optional attribute (a). Regions raise a slightly different issue: they may appear intrinsic, were it not for the fact that regions cannot be defined as precisely as years. This would place regions on the path of a functional aspect, to be represented by a powertype (b).

8.4.3.2.2 Delegation

From a functional perspective, a sourcing category delegates to one (or more) powertype the management of some of its properties. That scheme can be

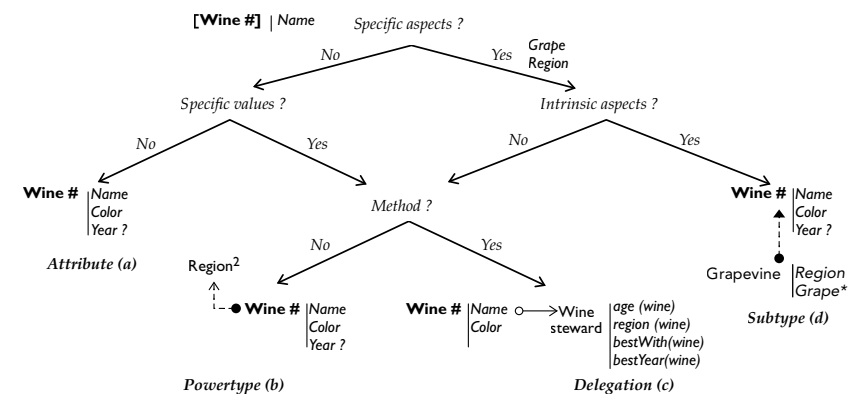


Figure 8.16 Ontological analysis of dependencies.

generalized to methods, with powertypes becoming categories on their own: in our case, a Wine steward category (c) could be tasked with access to attributes (e.g., region) as well as functions (e.g., age). Functional delegation is particularly important for representing requirements variants by categories because it enables the translation of some hierarchies into associations, a well-known improvement in object-oriented design. For instance, a wine steward could provide access to information (e.g., region, grapes) as well as knowledge (e.g., best year or wine suggestions). However, delegation cannot be applied to intrinsic aspects because such aspects are integral to ontological identities. In our example, the grapevine and region should be represented through a subtype (d).

The power type and delegation pattern can be generalized to apply to variants in both objects and processes. By providing a principled transition from inheritance to delegation schemes, its use with OPs paves the way for distributed agentic architectures.

8.4.3.3 Processes Variants and Rules

Compared to object variants, whose footprints can be directly identified through partitions, process variants are generally more challenging, given the pervasiveness of rules binding static, dynamic, and time-related variants.

8.4.3.3.1 Rules Taxonomy

To unravel the intricacies of conditions (the rule's domain) and actions (the rule's co-domains), ontological analysis must first consider the triggering conditions of the rules, whether intensional or extensional. From an engineering perspective, the focus should be on the categories that will represent the rule's co-domain, i.e., the footprint of the rule's actions. The objective would be to filter rules through OPs in order to align their domains and co-domains with realms (Figure 8.17).

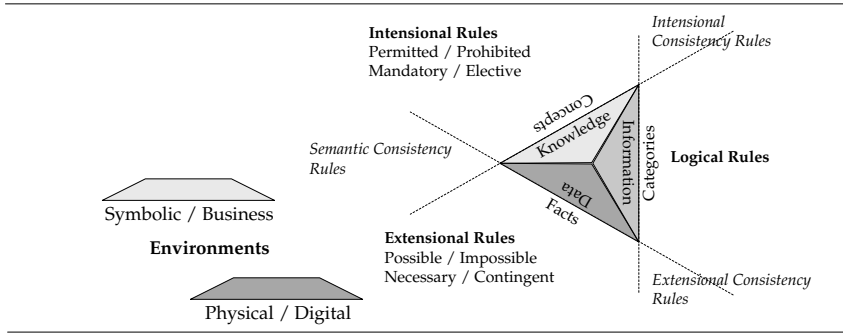


Figure 8.17 Rules taxonomy.

One could thus obtain six categories, three for endogenous rules:

- Extensional rules, intended to reflect facts' possibilities and necessities. For instance, rules about the monitoring and control of cooking processes (physical environment) and procurement processes (symbolic environment).
- Intensional rules, intended to reflect deontic (i.e., socially defined) constraints and conditions. For instance, the rules about religious and nutritional diets mentioned above.
- Logical rules, designed to represent extensional and intensional rules.

And three for consistency rules:

- Extensional consistency rules (ExtCR), designed to address the alignment of systems with the physical and symbolic environments of organizations.
- Intensional consistency rules (IntCR), designed to address the alignment of systems with the objectives of organizations.
- Semantic consistency rules (SemCR), designed to address the alignment of meanings across communication channels and interfaces. For instance, rules defining terms related to vegan and vegetarian diets, as well as the terminology used in messages exchanged with online payment systems.

A further analysis should distinguish between consistency rules that can be implemented locally (i.e., within entities' intrinsic boundaries), such as computation rules, and those that require some agency to coordinate and/or synchronize cascading consequences across intrinsic entities.

It is worth noting that, however, challenging the disentangling and filtering of rules, regardless of their meaning, may be, it is the kind of nominal undertaking that GenLMs are especially good at.

8.4.3.3.2 Rules Mappings

Unraveling rules according to their footprint will enable their alignment with requirements, cases, and models that are organized on the same principle (Figure 8.18).

- SemCR can be implicit when requirements are related to shared assets because consistency is meant to be secured there. However, they have to be explicit for stand-alone requirements (SA).
- IntCR may appear in BPs, BCs, and FCs. As such, they would be implemented in CMs and LMs.
- ExtCR may appear in UCs and SCs. As such, they would be implemented in LMs and TMs.

These mappings are of utmost importance for system architectures and, therefore, engineering processes: IntCR for the organization of processes and domains, and ExtCR for the organization of functional and nonfunctional capabilities.

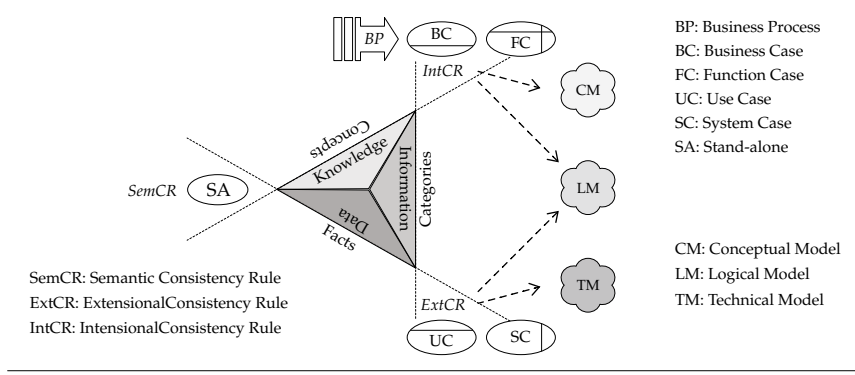


Figure 8.18 Rules mapping.

8.5 Engineering with Ontological Prisms (and Generative Artificial Intelligence)

This last section examines how OPs can be utilized to define flow-based work units and workflows, as well as the benefits they offer for the integration and transparency of MBSE processes, with a focus on DDD and validation.

8.5.1 Work Units

Once requirements are aligned with business concepts, managed categories, and legacy components, engineering work units and milestones can be defined and requirements distributed thereof, taking into account organizational, functional, and operational constraints.

8.5.1.1 Development Models

Beyond the variety of methodological tenets and tools, there are essentially two basic development patterns, each with its own merits. Iterative patterns are characterized by the same activity (or a group of activities) being carried out repetitively by the same organizational unit that shares responsibility, until a specific exit condition (either simple or combined) is met. Phased patterns, on the other hand, are characterized by sequencing constraints between differentiated activities that may or may not be carried out by the same organizational units. Taking a cue from Occam's parsimony principle, iterative development models should be the default option, provided that two prerequisites are met: with regard to continuous delivery, no milestones should be set upfront; and for shared ownership, teams should operate in time capsules free of external interventions. When these conditions cannot be met, phased models are required. Still, the apparent dichotomy may

be misleading, as it suggests alternative schools of thought. This can explain why iterative approaches are often chosen regardless of prerequisites, or why phased processes are frequently reduced to fixed sequences of predefined, one-size-fits-all activities. In fact, not only are decisions best made on a case-by-case basis, but better alternatives can often be achieved by refactoring requirements and introducing model-based methods that combine iterative and phased approaches.

MBSE methods allow combined patterns: on one hand, they provide well-defined targets for iterative development; on the other hand, they replace predefined activity-based phases with engineering flow-based work units. Decisions about work units can thus be made iteratively, with a primary focus on regrouping stand-alone requirements that can be met in time capsules independently of external events, i.e., without constraints on the contents and scheduling of intermediate deliveries. All being said, while stand-alone work units may seem like low-hanging fruit to be picked at will, the demarcation with higher-hanging units is not fixed and can be redrawn as requirements are elicited and/or refactored. That issue has been a major obstacle to the extension of MBSE to upstream engineering activities, materialized by the gap between logical and CMs, which hampers a seamless integration of flow-based software engineering. That gap can be bridged by using OPs to identify engineering flows and define work units accordingly.

8.5.1.2 Flow-based Work Units

Work units are defined as to combine the benefits of iterative and model-based engineering processes. The first objective is thus to identify business time frames and sort out organizational dependencies. That will set the first (domain) loop within which CMs are addressed. The aim of the second (service) loop is to determine intrinsic categories (#) in LMs, taking into account function and UCs. These categories are meant to be further extended with aspects defined by UCs and developed in the third (application) loop. The fourth (implementation) loop will address coding (Figure 8.19).

Defining work units and workflows bottom up by the changes operated on engineered flows, instead of top down from predefined activities, does away with a historical paradox: software engineering has failed to apply lean and Just-in-time processes to symbolic artifacts, something that manufacturing industries have successfully achieved with physical ones for more than half a century. Judging by what has been accomplished by manufacturing processes during that period, there should be a wide range of potential benefits for system and software engineering processes.

Concomitantly with the engineering loops, the consistency of models can be maintained by applying two types of transformations: knowledge-driven transformations between conceptual and LMs, and language-driven transformations between LMs and TMs.

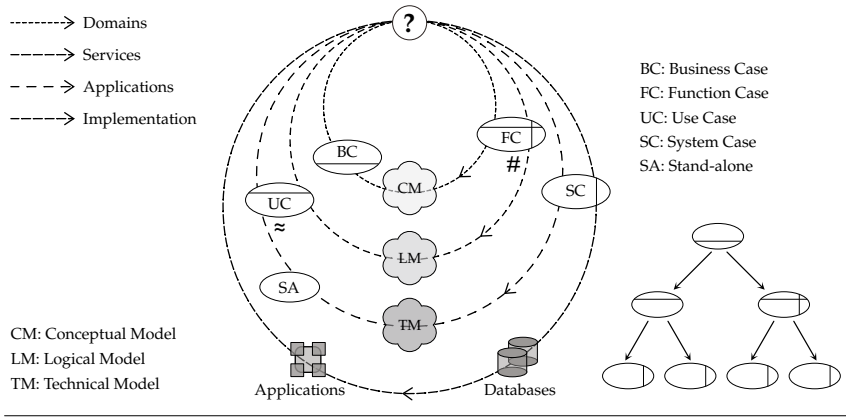


Figure 8.19 Knowledge-driven (CM > LM) and language-driven (LM > TM) model transformations.

8.5.2 Model-based Engineering

As viable systems immersed in competitive environments, enterprises' sustainability is primarily determined by their ability to interpret data, manage information, and utilize it as knowledge. Managing information is carried out through domains, while utilizing it is done through processes, which creates a kind of dilemma due to the intrinsic discrepancies in purposes and life cycles. Domains are intended to manage shared and lasting assets with stable structures and semantics, whereas processes are driven by specific purposes that are meant to change with opportunities in business environments. This dichotomy mirrors the one between knowledge (domains) and language (processes), as illustrated by the DDD approach and its ubiquitous language.

8.5.2.1 Domain-driven Design

A brainchild of Eric Evans (see Bibliography), DDD replaces a process perspective with a domain one, and carries on with engineering directly from the complexity of business logic. Requirements are organized in terms of conceptual domains (for business entities, the equivalent to intrinsic anchors) and bounded contexts (for business semantics, the equivalent to functional aspects). A so-called ubiquitous language is meant to deal with the semantic overlaps between bounded contexts and to support shared access to business entities. The relevance of the DDD approach comes from its analysis of domain complexity and shared access. However, while rightly identified at a conceptual level, these issues are not fully addressed as such.

To begin with, the domain (shared, stable) versus process (specific, changing) dilemma is overlooked, and the mapping between conceptual and LMs is supposed to be addressed through a high-level ("ubiquitous") programming language. Mirroring that modeling shortcut, there is no room for a distinction between organization, systems, and software, and consequently between business and engineering concerns. Then, object-oriented design serves as an engineering bypass between business domains and systems applications, avoiding possible modeling breaks. The implicit assumption is that symbolic representations of BPs and systems architecture can be directly aligned through programs. These lacunae can be addressed through OPs.

8.5.2.1.1 Roots, Aggregates, and Bounded Contexts

The distinction between the identity and structure of objects, on one hand, and the semantics and use of features, on the other hand, is arguably a core tenet of DDD. This distinction can be neatly aligned with intrinsic and functional ontological dimensions. Accessing intrinsic aggregates through a single root (#) guarantees the continuity and integrity of shared business entities. Concomitantly, bounded contexts are introduced to manage the shared functional aspects of business entities without breaching the continuity and consistency of aggregates and anchors. In our example (Figure 8.20), roots and aggregates are anchored to environments (Staff #, Customer #, Supplier #) or identified at the enterprise level (Function #, Recipe #, Ingredient #). These intrinsic business entities (roots and aggregates) are defined by categories in conceptual domains, along with intrinsic aspects. Bounded contexts, for their part, are represented by categories in logical domains (Figure 8.20).

It should be noted that (i) except for roots (Staff #, Customer #, Supplier #), conceptual and logical domains may or may not be aligned, and (ii) business

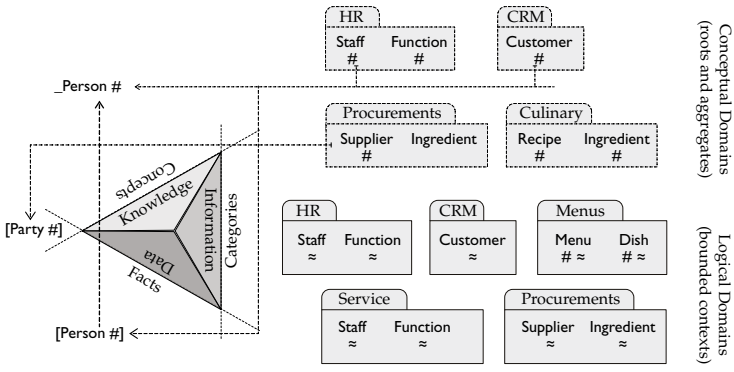


Figure 8.20 Ontology-driven domains.

entities may or may not be rooted in conceptual domains (e.g., Menu # and Dish #). While a level of indirection between conceptual and LMs should be a prerequisite for the versatility and adaptability of EAs, it raises challenging issues regarding the design of shared access by processes. This highlights the inherent orthogonality between processes (specific and changing) and architecture (shared and stable), and more generally between direct communication (processes) and mediated communication (models).

8.5.2.1.2 Ubiquitous or Domain-specific Languages

As far as DDD is concerned, the objective of ubiquitous languages is to build a semantic bridge between domain analysis and software design, thereby tying concepts to implementations. However, this objective raises serious questions:

- Binding conceptual and technical semantics goes against the tenets of object-oriented design as well as layered architectures.
- The proposed ubiquitous language can either be implemented through directories of standard entities and operations, lacking grammar or unifying semantics, or through domain-specific languages (DSLs).

These questions have been at the core of systems engineering since its inception as a discipline; they have gained new relevance with GenAI, typically in relation to knowledge (models) and language (processes).

8.5.2.2 Model-driven Architecture

8.5.2.2.1 Models, Domains, Languages

When employed within a three-pronged ontological perspective, terms such as "conceptual model" or "physical model" can be confusing. Using domains is more specific, as it corresponds to a set of categories whose identification and semantics are managed by the same authority. Combined with finer granularity, this more definite understanding of domains allows for greater flexibility in their use, particularly concerning the sequences and mappings of conceptual, logical, and technical domains.

While DDD methods address the integrity issues of the shared domain, they effectively sidestep the associated problem of differing semantics. Alternatively, DSLs and GenLMs operate explicitly or implicitly (respectively) within bounded semantic contexts.

8.5.2.2.2 Models and Architecture

The OMG's model-driven architecture offers a more principled understanding of modeling layers (Figure 8.21):

- CIMs describe business entities and processes independently of supporting systems.

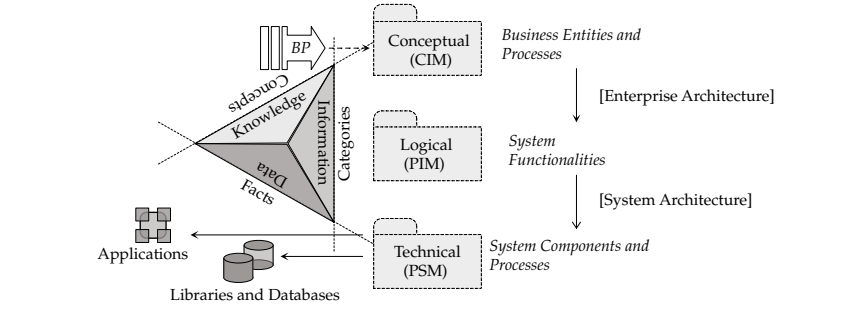


Figure 8.21 OPs and models transformation.

- PIMs describe business entities and processes as supported by systems independently of platform technologies.
- PSMs describe system components as implemented by specific technologies.

These layers correspond to the conceptual, logical, and technical modeling layers and provide a rationale for their transformation: first, EA (conceptual to logical transformation); then, system architecture (logical to technical transformation); and finally, platform and programming language (code generation from TMs).

8.5.2.3 From Models to Code

The raison d'être of MBSE is to address concerns separately: business and organization (CMs), symbolic representation (LMs), and system implementation (TMs). It also determines models' capabilities: interpretable, actionable, or executable, respectively.

8.5.2.3.1 From Business Processes to Executable Models

DSLs can be used to bypass CMs and translate BPs into LMs. Such models are actionable, as they can emulate the expected behavior of the system within logical realms. Moreover, since the semantics used to represent categories are established within bounded contexts, the transformation of LMs into technical ones can be carried out regardless of domain-specific semantics. All that is needed is comprehensive catalogs of resources and agreed-upon grammars.

8.5.2.3.2 Interpretation of Conceptual Models

As illustrated by DDD's quest for an "ubiquitous" language, the core issue of MBSE is to ensure ecumenical (i.e., nonspecific) access to specific domains. This issue can also be expressed in terms of conceptual debt between business and system representations, or between conceptual and LMs. From an ontological

perspective, this issue corresponds to the gap between foundational ontologies and siloed ones. The validation of the conceptual/logical transition is examined further, while the broader ontological issue will be discussed in Chapter 9.

8.5.2.3.3 Actionable and Executable Models

Contrary to CMs, LMs are supposed to be fully actionable, leaving no room for interpretation. Consequently, their translation into executable (technical) models, and then into code, can be carried out regardless of domain-specific knowledge. This is something that GenLMs excel at, as illustrated by the so-called vibe coding helpers.

8.5.2.4 Generative Artificial Intelligence Agents and Coding

The surge of GenLMs, along with their cognitive limitations, has revived collaboration models between specialized agents entrusted with specific tasks. A long-standing endeavor in artificial intelligence (AI) (Ferber, 1999), agents can be broadly defined as software components designed to achieve specific objectives within a bounded context, given a set of resources and tools. Newly equipped with GenAI capabilities, this new generation of agents is expected to possess a degree of autonomy in observation, problem-solving, and even learning, enabling them to assess, make, and implement decisions on their own. Ideally, these cognizant agents could thus collaborate with conversational agents to address a wide range of issues. In practice, however, communication and cognition are entangled, generating nonfactual and nonsensical hallucinations, not to mention the lack of transparency, traceability, and accountability. These concerns can, to a certain extent, be addressed when agents operate within strictly bounded factual and semantic contexts, as it's the case with TMs and coding.

8.5.3 Ontological Validation

The aim of ontological validation is to focus on intrinsic and functional consistency, regardless of business-specific aspects and rules. With OPs that can be done at the model level between domains, and at the ontology level between realms.

8.5.3.1 Domain Consistency

Models are organized into domains set under a single authority. Most MBSE frameworks support domain consistency with regard to references, structures, inheritance, and semantics. With OPs, intrinsic identities and functional aspects allow for a more generic approach, independent of the modeling languages. For instance, an intrinsic connector between Dish and Diet (Figure 8.22, left) represents a business-specific statement about dishes identified by diets; however, it also introduces a potential inconsistency between functional connectors to ingredients and diet intrinsic rules.

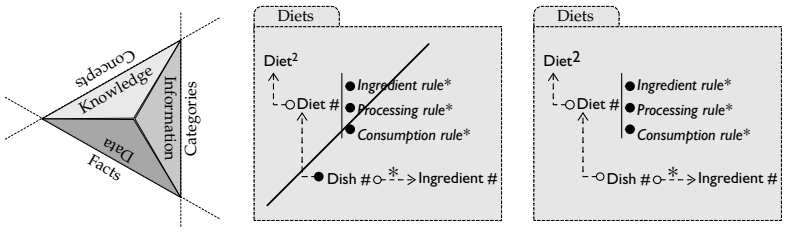


Figure 8.22 Domain consistency: if the reference to diet is intrinsic (left), there is a potential inconsistency between the diet intrinsic rule and the dish functional connector.

While the same level of consistency can be achieved with modeling language such as Unified Modeling Language, they do that in more specific ways and without reference to ontological constraints, by nature, more ecumenical.

8.5.3.2 Ontological Consistency

Compared to domain consistency, which pertains to homogeneous representations, ontological consistency pertains to the alignment of models with their context and purpose. Addressing this issue within the traditional modeling paradigm is problematic due to its implicit assumption that systems, contexts, and purposes can be uniformly represented by categories, thereby concealing potential misalignments of modeling semantics. That pitfall can be avoided with OPs that allow for an explicit distinction between representations and their respective abstraction semantics.

8.5.3.2.1 Abstraction Semantics: Parthood vs Inheritance

As we have seen before, we are dealing with two different abstraction paradigms: inheritance with regard to categories and parthood with regard to facts and concepts. By taking advantage of the semantic neutrality of ontological connectors (see Chapter 2), ontological validation can focus on intrinsic and functional consistency independently of business-specific considerations. Regarding UCs, the semantics of <include> and <extend> connectors, as originally defined by Ivar Jacobson, do not pertain to category abstractions but rather to the composition of execution flows. Therefore, the concepts of abstraction and inheritance, as commonly understood, are irrelevant unless they are significantly revisited.

The fault lines between abstraction semantics should be addressed at both ends of the model-based engineering process: upstream, between concepts and categories, and downstream, between categories and facts.

For instance, while religious and nutritional diets can be viewed as intrinsically different concepts, they appear as overlapping functional subsets of an observed

dataset when considered as facts. That ontological duality can be implemented through functional subtypes representing the conceptual distinction, and power types representing operational distinctions. Nutritional and ethic subtypes are functional because they may overlap; powertypes are used to document dietary constraints (Figure 8.23).

8.5.3.2.2 From Concepts to Categories

The distinction between ontological concepts on the one hand, and CMs on the other hand, may be redundant, as it's the case with siloed ontologies and more generally when domains and responsibilities can be aligned. But isomorphic realizations cannot be assumed for heterogeneous ontologies, which entails an explicit distinction between ontological concepts (knowledge) and their projection as CMs (information).

Taking nutritional and religious diets as an example, _Diet # could initially be considered an anchor; however, the alignment of hierarchies would prevent categories from mixing multiple functional inheritances and intrinsic functional connectors, e.g., between religious diets and institutions (Figure 8.24).

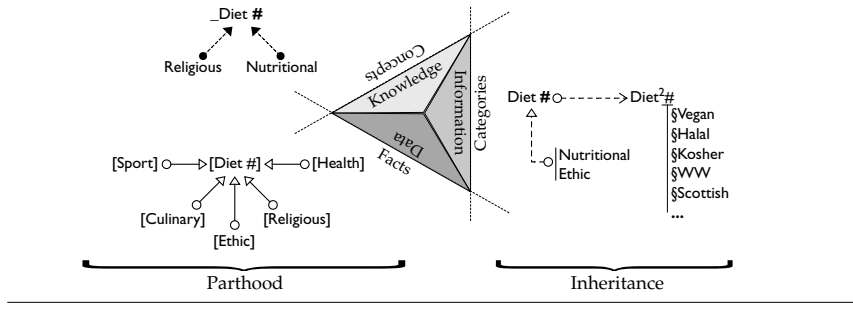


Figure 8.23 Abstraction semantics and ontological realms.

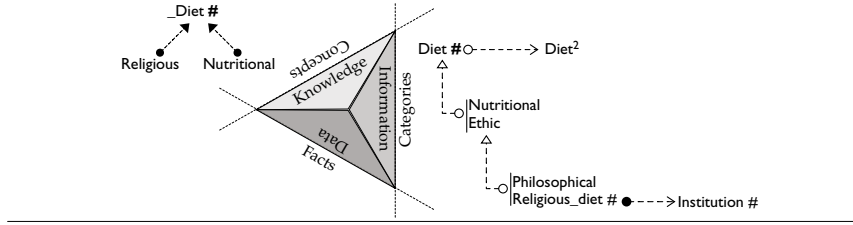


Figure 8.24 From detached meanings (west side) into purposeful projections (east side).

Instead, introducing a Religious_diet # category with no intrinsic connector to the Diet # category will give more latitude to business domains while maintaining a reference to the _Diet # concept. Concomitantly, the Diet[2] power type will ensure semantic consistency.

8.5.3.2.3 From Categories to Facts

Verification (or external consistency) considers the mapping of surrogates, the digital representations managed by systems, to their individual counterparts in environments. At system level, this issue has been addressed by Barbara Liskov (Liskov, 1993) in software design: the so-called Liskov Substitution Principle states that the sets of identified individuals being represented should not depend on the level of abstraction: if S is a subtype of T, then instances of T may be replaced with instances of S without altering any of the desirable properties of the software components. Translated to ontological analysis, the principle would state that the surrogates managed through a logical domain (i.e., the instances of its categories) must be aligned with the targeted facts in environments, regardless of the levels of abstraction considered.

For instance, staff on the ground are composed (parthood) of employees and temps (Figure 8.25a), while the database introduces functional inheritance to organize staff according to functions (Figure 8.25b). The objective is to ensure that individuals identified in environments correspond to the surrogates managed by databases, regardless of abstraction levels.

Breaches of that revisited substitution principle stem from the misguided use of intrinsic and functional inheritance: representing individuals (e.g., members of the staff) through functional categories (e.g., cook or waiter) instead of intrinsic ones (e.g., temp or employee) will produce inconsistent outcomes.

On a broader perspective, the substitution principle illustrates the continuity between software design (models and surrogates) and EAs (systems and environments). With OPs, the substitution principle can be used to design and manage test cases.

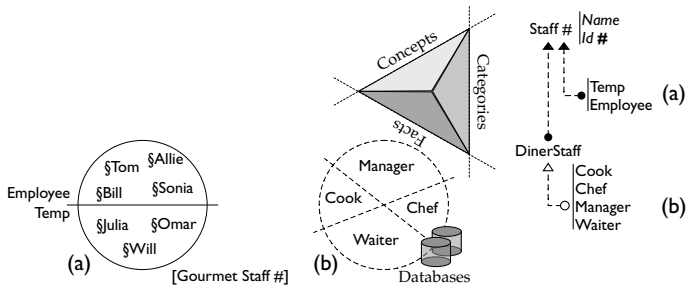


Figure 8.25 Ontological substitution principle (Omar is a temporary chef).

8.5.4 Risks Management with Ontological Prisms

Quality is a quantity: it represents the probability that nothing will go amiss. In this section, we assume that comprehensive and in-depth preventive actions have been enacted concerning model consistency and validation. Additionally, given bounded technical contexts, code is supposed to be reliably generated from TMs and tested automatically. However, systems operate in unbounded environments where, as Donald Rumsfeld once famously said, beyond what we know (bounded contexts), there are things that we know we don't know and things we don't know we don't know. As mentioned in Chapter 7, known unknowns can be addressed through stochastic and strategic horizons. Could ontologies assist with unknown unknowns?

8.5.4.1 Threats and Black Swans

Testing systems serve a dual purpose: they are preventive with regard to operations carried out in actual circumstances, and predictive with regard to forthcoming unexpected situations. Assuming that everything has been addressed regarding (i) the quality of models and code, and (ii) the management of unexpected disruptions in environments, what remains to be tested? On the one hand, security issues stem from deliberate attempts to exploit system vulnerabilities. On the other hand, black swan phenomena, a term coined by essayist and statistician Nassim Nicholas Taleb (Taleb, 2007), characterize unexpected and massively disruptive events that are resistant to stochastic analysis.

Between deliberate attacks and black swan events, tests may also consider quirks in users' behaviors, which could make decisions based on the visual context and options presented by user interfaces, before they consider the underlying business content and logic of the situations. In a standard testing context, user behavior is expected to be governed by the logic embodied in models and, consequently, to be adequately covered by corresponding test cases. Since these test cases are not intended to cover spontaneous behaviors, they leave systems exposed to disruptive and unpredictable knee-jerk reactions, reflecting, at the individual level, the black swan events in broader environments.

8.5.4.2 Tests and Ontological Prisms

A growing number of security breaches seem to stem from the ability of GenAI to identify vulnerabilities in system defenses. Designing test plans that address this issue is inherently challenging for the obvious reason that random searches cannot be easily scripted. Initial attempts to apply AI and ML to automated test generation have been hindered by a reliance on explicit knowledge and, consequently, by the need for assisted learning.

Part of these hurdles can now be overcome by adding GenLMs to programming languages and development frameworks. That's because when it comes to

the effectiveness of random schemes, two points are decisive: contexts must be strictly bounded, and being smart and knowledgeable is irrelevant. Programming languages can take charge of the first point, while GenLMs can unquestionably address the second. However, users' knee-jerks don't take place in a void but within known functional circumstances. Given the porosity of nonsymbolic and symbolic cognitive layers, the effectiveness of functional test cases may be compromised by unfathomable mental associations (e.g., between the respective colors of user interface elements and food items), triggering unexpected quirks that could thwart the best-designed handrails. Enter OPs and their explicit representation of facts, concepts, and categories: facts for managing test cases and user quirks; concepts for identifying stray patterns and establishing learning paths; and categories for managing what systems expect from users (Figure 8.26).

Given GenLMs properly trained and tuned, they could be harnessed with requirements and datasets in order to generate random user behaviors. By leveraging their language (Chapter 5), reasoning (Chapter 6), and decision-making (Chapter 7) capabilities, OPs could be used to categorize failures in terms of perceptions, projections, and experience. First, the system under consideration would be able to identify nonfactual and/or nonsensical inputs (perceptions); then it could do the same for unaccounted for or inconsistent variants in business objects and/or processes (experience); finally, the system could detect unauthorized or unfounded operations (projections). Not surprisingly, these tiers correspond to the ones defined for entropy in Chapter 4.

8.5.4.3 Tests and Systems Entropy

The immersion of enterprise in digital environments, combined with the ubiquity of GenAI capabilities, means that a growing part of disruptive events cannot be addressed by planned schemes, which brings back cybernetics and the concept of entropy.

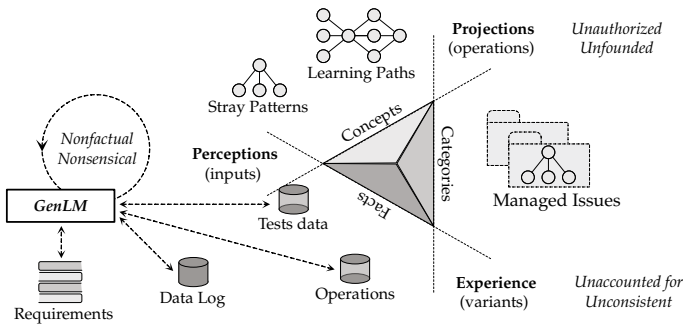


Figure 8.26 Managing tests with OPs.

In ontological terms, entropy is best understood as the part of observations that cannot be accounted for by a truth system. Applied to EAs, changes in entropy would stem from changes in complexity due to:

- A discrepancy between the terms used by users and the meanings recognized by systems.
- A partition of facts that cannot be accounted for by the categories managed by systems.
- The raising of issues that are undecidable within the existing organization of domains and responsibilities.

These sources of entropy can be neatly aligned with the unfathomable disruptions and best addressed through OPs.

Key Points

The merging of EAs into their digital environments, combined with the pervasiveness of GenAI technologies, gives renewed relevance to Stafford Beer's vision of the enterprise as a viable system. The sustainability of such a system depends on its ability to perceive changes in the environment and adapt its organization, systems, and behaviors accordingly. This has two major consequences for systems engineering. First, there is a need for seamless integration of software components and BPs. Second, there is the intricate weaving of systems and knowledge engineering. This can be achieved using the Pagoda blueprint for EAs and OPs for their digital twins.

The Viable System Paradigm (Beer, 1995) suggests that change should be the norm, which can only be achieved through well-defined points of change with clearly identified footprints. Four entry points are proposed: business, function, user, and SCs. Footprints are defined according to the standards of MBSE, namely CMs, LMs, and TMs.

Requirements capture and elicitation combine two perspectives. With regard to EA, GenLMs can be used to earmark requirements for entry points: business objectives, user requirements, business domains, and SCs. Requirements units could then be defined according to Aristotle's three unities principle for action (roles), time (events), and space (location). These requirements units could subsequently be anchored to OPs.

With requirement units anchored to OPs, the objective is to untangle variants and rules. First, we use taxonomies to translate variants into powertypes or subtypes. Then, based on the mapping of rules' conditions (the rules' domain) and actions (the rules' co-domains) with empirical (facts), deontic (concepts), or logical (categories) realms, rules can be rearranged as semantic, extensional,

and intensional constraints between facts and concepts, facts and categories, and concepts and categories, respectively.

The primary objective of refactoring requirements through OPs is to enable a flow-based definition of work units and, thereby, bottom-up model-based engineering workflows. In line with the imperative of continuous change, engineering processes can thus be governed by two different time frames. One, focused on business domains, pertains to changes in architectural assets that are intended to be shared and stable; the other, focused on processes, pertains to changes driven by specific business opportunities. Stand-alone developments, which do not affect shared resources, are set in time capsules.

Last but not least, OPs allow for a distinction between intentional validation of models (between concepts and categories) and extensional verification of systems (between categories and facts). Test cases can be designed accordingly, aiming at endogenous flaws (scripted tests) or exogenous disruptions (generative tests).

9

The Engineering of Knowledge

> *I always tell the truth: not all of it, because one cannot get there ... Words are missing ... It is even through this impossibility that truth is bound to reality*
>
> Jacques Lacan

The aim of ontological prisms is to provide an ecumenical knowledge engineering framework that supports the development of both siloed and heterogeneous ontologies. The first part of this book presented the architecture of ontological prisms, the second part covered core cognitive functions, and the third part introduced the modus operandi. This chapter concentrates on the engineering of knowledge itself.

Outline

Taking for granted that knowledge is inherently dynamic, pushing edges across contexts and concerns, its engineering cannot be circumscribed in silos and time capsules. To meet the challenge, knowledge engineering with ontological prisms (KEOPS) combines incremental and iterative developments (Figure 9.1).

Incremental Development

Incremental moves operate like Jacquard looms, which are machines used to weave complex patterns and motifs into the fabric of textiles. Three rods, one for each realm, are moved parallel to the boundaries of the prisms. Inward moves introduce new facts, categories, or concepts from nondescript entries and use intersecting gears to map these new entries to existing ones. Outward moves translate successful mappings of facts, categories, and concepts into interoperable data, information, and knowledge, respectively.

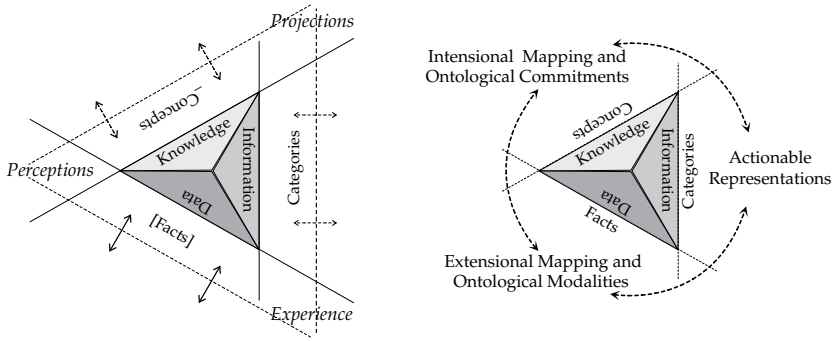


Figure 9.1 KEOPS combines incremental (left) and iterative (right) moves.

Iterative Development

Concomitantly, iterative moves traverse extensional, intensional, and logical realms. Extensional iterations consider native or man-made objects or phenomena that can be observed in symbolic or physical environments and are deemed relevant. Intensional iterations pertain to concepts that cannot be substantiated and consequently rely on ontological commitments addressing the mapping of concepts to facts. Logical iterations address the representations of selected extensional or intensional elements, using formal and empirical languages.

The game-changing benefit of this double-edged approach is to address the integration of heterogeneous representations at different stages of ontologies life-cycle, obtained through the interoperability mechanisms described in Chapter 3: perceptions (alignment of meanings between observations and intentions), projections (interoperability of temporal and organizational representations), and experience (feedback and learning from doing).

9.1 Mapping Extensions

While knowledge is not necessarily driven by purposes, it nevertheless aims at some territory and is driven by some topics of interest. The first step is therefore to identify individuals deemed to be relevant within the territory considered. Such individuals can be directly classified according to their nature, whether physical or symbolic. For the physical individuals, they can be further divided into natural entities or artifacts. That first step should also delineate prior knowledge

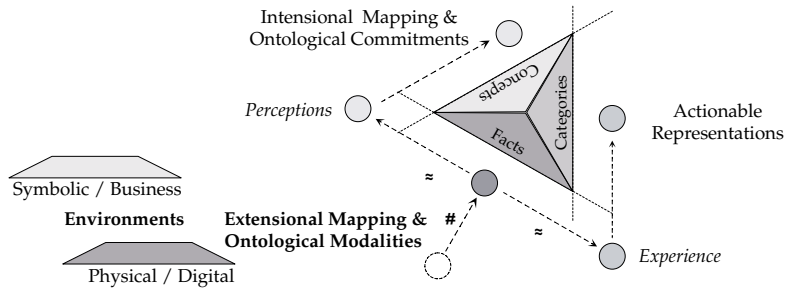


Figure 9.2 Mapping territory: intrinsic identity (#) and functional aspects (≈).

by distinguishing individuals who are already represented by digital surrogates, regardless of the organization or system involved.

The first step (inception) is to identify relevant individuals in terms of intrinsic identity (#) and functional aspects (≈). The second step (integration) is to use perceptions and experiences to integrate individuals with existing concepts and categories, respectively (Figure 9.2).

9.1.1 Ontological Footprint

As emphasized in Chapter 1, ontological prisms are agnostic about the essence of things. Instead of making upfront decisions about foundational entities, KEOPS starts with questioning: what should be known about the things being considered? In this way, knowledge engineering differs from system engineering, as detailed in Chapter 8, because it does not begin with requests but with facts, possibly documental ones, and it does not target actual systems but rather symbolic ones commonly known as knowledge management systems (KMS).

9.1.1.1 Contexts and Concerns

Regardless of purposes, ontologists must outline the territory they wish to map, as well as their objectives. We begin with extensional ontologies, which can be considered a default option. Intensional (or "pure") and logical (or applied) ontologies will be examined in relation to truth commitments and actionable representations, respectively.

9.1.1.1.1 Anchors

The edges of knowledge are inherently foggy; therefore, contexts can be more effectively outlined using anchors – objects or phenomena that can be unambiguously

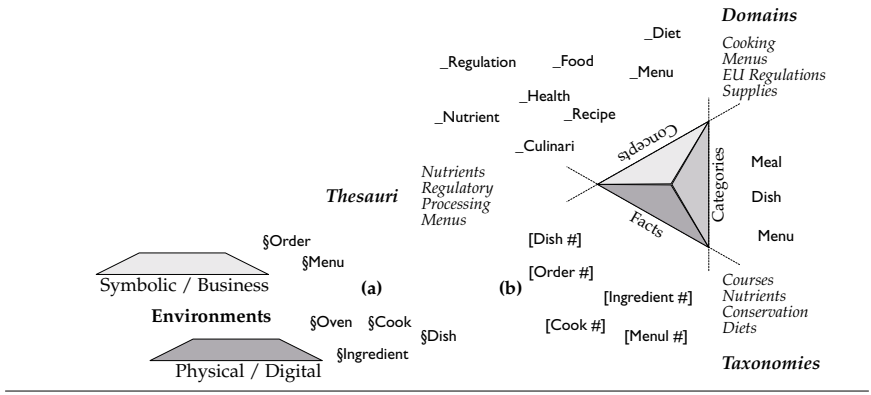


Figure 9.3 Deriving types (b) from observed individuals (a).

identified upfront as relevant. The essence and aspects of these anchors will then be used to develop and elaborate on the territories in question.

Consider a leading firm in the prepared meal industry that is interested in the benefits of ontologies in relation to compliance with relevant European Union (EU) regulations. Aside from observed (§) individuals, physical (e.g., §Cook, §Oven) or symbolic (e.g., §Order, §Oven, §Menu) (a), the initial outline can be provisional, combining statutory (e.g., _Regulation) or subjective (e.g., _Health, _Food) concepts; tentative thesauri (e.g., *Nutrients*, *Processing*, *Regulatory*) and taxonomies (e.g., *Nutrients*, *Diets*, *Conservation*); legacy categories (e.g., Meal, Supplier, Ingredient); and business (e.g., *Menus*, *Supplies*) or statutory (e.g., *EU Regulations*) domains. That will lay the groundwork for identifying types from observed individuals, such as [Dish #], [Ingredient #], [Menu #], [Cook #], [Oven #], and [Order #] (Figure 9.3).

Then, the choice of anchoring facts must be selective, as they will determine development paths. Compared to inward (or siloed) conceptual modeling, which focuses on the relationships and features of entities, ontological modeling takes an outward perspective and first examines boundaries and modalities.

9.1.1.2 Boundaries

Attempts to set limits on scope and definite objectives prematurely are often inconclusive, as territories must be explored before they can be marked. Alternatively, ontological prisms can be used to outline boundaries through thesauri (semantic boundaries), taxonomies (operational boundaries), and domains (organizational boundaries).

Semantic boundaries can be drawn through dedicated thesauri, which would be progressively aligned. For instance, it will be necessary to align terms employed

for nutrients by the food industry with the ones defined by regulatory authorities, and then for the mapping of terms with business and regulatory concepts.

Operational boundaries can be drawn with regard to taxonomies. Given sets of facts, ontologists must determine relevant partitions that will serve their purpose, e.g., nutrients, fat, sugars, or salt, while taking into account existing taxonomies such as courses or diets.

Organizational boundaries will be considered later with regard to projections and actionable representations.

9.1.2 Ontological Blueprints

9.1.2.1 Ontological Modalities

Taking a leaf from Gabriel García Márquez's literary masterpiece, at the beginning, "The world was so recent that many things lacked names, and in order to indicate them, it was necessary to point." This aptly characterizes ontologists' opening moves: they must begin by assigning labels, or more precisely, signs, to things (see Chapter 5). Using thesauri to ensure that signs are unique, ontologists then consider what should or could be known about the things being labeled. This is when ontological modalities come into play.

Modalities (introduced in Chapter 2) can be viewed as ontological stereotypes as they define features and behaviors independently of classes, providing a declarative alternative to imperative meta-models. This allows for an iterative approach that is unconstrained by conceptual legacies. Instead of prematurely enforcing inheritance from foundational entities, declarative modalities adopt Occam's razor principle of parsimony, which states that issues should be addressed using the smallest and least constraining possible set of elements.

Due to their freewheeling status, specific modalities can be introduced or withdrawn as needed. Moreover, the intrinsic or functional nature of modalities is left to the discretion of ontologists, who may decide, for instance, that biological identification is an intrinsic modality of security actors. KEOPS relies on seven main modalities: temporality, instantiation, identification, agency, communication, structure, and veracity.

9.1.2.1.1 Temporality

Temporality epitomizes the difference between foundational concepts, such as perdurants/occurrents and endurants/continuants, and ontological modalities. KEOPS uses four exclusive temporal modalities to characterize life cycles:

- Continuous, when individuals remain relevant until further notice, e.g., invoices.
- Instantaneous, when individuals have no life cycle, e.g., billing.

- Timespan, when individuals have a predefined life cycle, e.g., ordering or cooking processes.
- Periodic, for sequences (collections) of repeated individuals, e.g., monthly reports.

Temporality is an intrinsic modality meant to be determined in relation to instantiation and identity principal modalities.

9.1.2.1.2 Realm, Instances, and Tally

Realm is an intrinsic and exclusive modality that characterizes the origin of individuals: extensional (physical or symbolic), intensional, or logical. For instance, one can first introduce actual diners (extensional, physical), regular diners (extensional, symbolic), and customer records (logical, digital). Notations (“ ”, §, [], _) allow the use of synonyms.

Instanciation is an intrinsic and exclusive modality that characterizes the nature of individuals: physical, symbolic, digital, or nominal. It must not be confused with realm (facts can be symbolic) or with the identity principle (which is not exclusive).

Tally is an exclusive (but not necessarily intrinsic) modality addressing constraints on the number of individuals:

- No instance, for stereotyped descriptions, e.g., security protocol in case of fire alarm.
- Unique instance, for particular occurrences, e.g., private events in premises, or process control, e.g., fire alarm systems.
- Finite instances, for collections of instances, e.g., team members.
- Undetermined number of instances, e.g., suppliers.

The tally modality is designed to serve as a generic complement to the cardinality property supported by standard modeling methods.

9.1.2.1.3 Identity Principle

Identity Principle is a nonexclusive modality that characterizes the way individuals can be identified, typical examples are:

- Organic, for natural or biological individuals, e.g., animals, grapevines.
- Nominal, for literal identities, e.g., surnames.
- Social, for identification determined by external organizations, e.g., EU protected designation of origin (PDO).
- Organizational, for identification determined by positions within organizations, e.g., First Desk.
- Digital, for coded identification, e.g., time stamp.
- Material, for identification determined by structure, e.g., cars.
- Temporal, for identification determined by time frames, e.g., service in restaurants.

- Functional, for identification determined by outcome, e.g., recipe.
- Operational, for identification determined by execution, e.g., recipe cooking.

Identity principles (IPs) can be combined or composed. Combined IPs can be set independently, such as a person's biological and social identities. In contrast, composed IPs are set jointly, for example, cooking processes identified by a recipe and a time stamp.

9.1.2.1.4 Agency

Agency is an intrinsic and exclusive modality that characterizes the behavior of individuals. KEOPS introduces three basic options:

- Passive, no agency, e.g., menus.
- Reactive, act according to instructions, instantly or otherwise, e.g., cooking equipment.
- Autonomous, act on their own, e.g., customers.

Agency modality is taking center stage with the development of digital agents that possess generative artificial intelligence (GenAI) capabilities. When viewed through ontological prisms, these capabilities can be further refined in terms of observation, reasoning (Chapter 6), and decision-making (Chapter 7).

9.1.2.1.5 Communication

Communication is a nonexclusive modality that characterizes the language(s) that can be used by individuals to communicate:

- None, when facts appear as black boxes.
- Physical (analog), for communications that can only be achieved through observation, e.g., weather phenomena, sounds, and visual signs.
- Digital, for the exchange of messages or documents regardless of the semantics involved.
- Symbolic language, when the content of messages or documents can be interpreted through given syntax and semantics.
- Natural language, when the content of messages or documents can be interpreted through given syntax, semantics, and pragmatics.

Communication modality comes as a complement to agency modality.

9.1.2.1.6 Structure

Structure is an exclusive modality meant to reflect parthood theory, typically:

- Elementary, for unstructured individuals, e.g., a cook or oven.
- Aggregation, for a set of individuals that are functionally bound together while keeping their own separate identities, e.g., order and meals.

- Composition, for a set of individuals that are structurally bound (identified) together and cannot be accessed independently, e.g., dish, fat, sugar, and salt composition.
- Collection, for a set of individuals viewed as functionally identical, e.g., a table's diners.
- Container, for a set of individuals functionally undetermined, e.g., mailing list.

As noted above for instantiation modalities, structure modalities are initially defined for individuals before being used to define categories. It follows that during the mapping of territory, the distinctions between entities and aspects are supposed to be undecided. More importantly, the semantics of structure modality, or parthood theory, which pertains to individuals, is inherently different from the semantics of abstraction, which pertains to categories.

9.1.2.1.7 Veracity

Veracity, or ontic modality, aims to characterize truth commitments. In addition to basic past, actual, and virtual veracity, other options could be considered, such as:

- Nominal, for facts reduced to signs, known as tokens in the context of large language models.
- Observed, for identified individual objects or phenomena, e.g., customer, ingredient.
- Managed, for instances of artifacts already represented by categories, e.g., dishes, ovens, orders.
- Asserted, deduced, simulated, or projected individuals.

Veracity conveys the complexity of ontological dimensions and how they can be addressed through ontological prisms.

9.1.2.2 Modalities and Processes

The aim of ontological modalities is twofold: first, to bridge the gap between foundational and domain-specific approaches to ontologies; and second, to replace the reuse of imperative meta-modeling schemes with declarative ones. Applying modalities at the early stages of knowledge engineering enables ontologists to hit the ground running with clear perspectives and reliable yet adaptable guidelines, free from the biases and unwarranted commitments of meta-models or foundational entities (Figure 9.4).

For instance, the modalities used to characterize [Recipe #], [Menu #], and [Dish #] illustrate typical patterns of symbolic and virtual, symbolic and actual, and physical and actual objects, respectively. Concurrently, the modalities of [Cooking process #] illustrate the distinction between the descriptions of a virtual process ([Recipe #]) and its realization ([Cooking process #]) and outcome ([Dish #]).

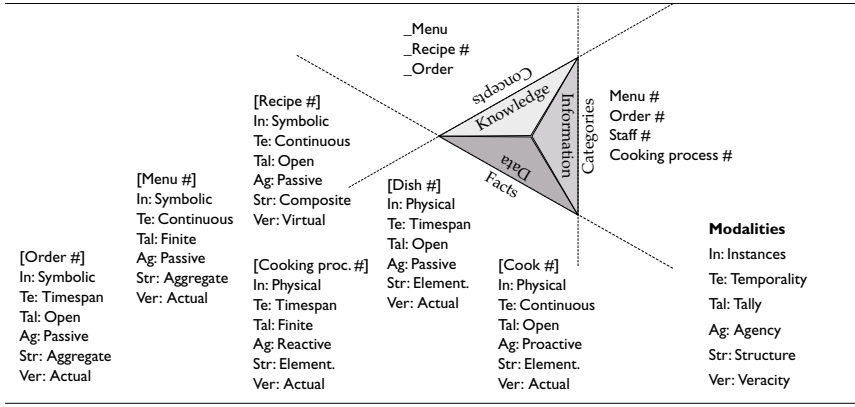


Figure 9.4 Using modalities early to hit the ground running with ontologies.

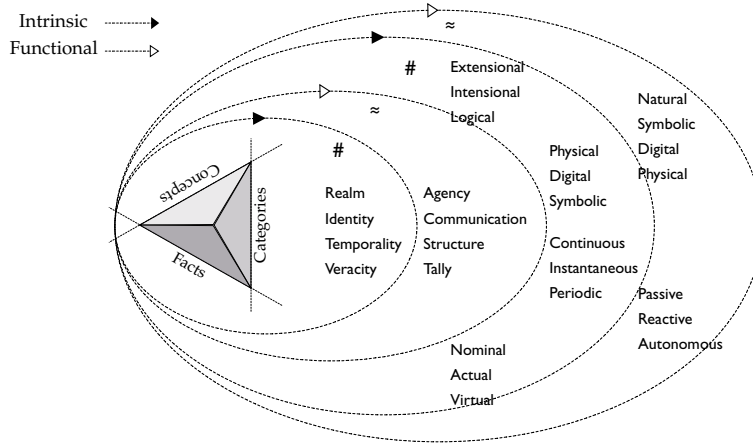


Figure 9.5 Iterative knowledge engineering combining intrinsic (#) and functional (≈) increments.

Given an initial set of facts and modalities, ontological thesauri and taxonomies are used to align them with known concepts and legacy categories through iterative processes that combine intrinsic (#) and functional (≈) increments (Figure 9.5).

Approaches that combine iterations for tasks and increments for products significantly enhance the transparency and traceability of development processes. These benefits are critical for ontology engineering because, unlike models,

non-siloed ontologies are inherently works in progress. Therefore, they must keep track of the semantic and conceptual steps that link facts, concepts, and categories, whether to support reasoning (Chapter 6) or decision-making (Chapter 7).

9.1.2.3 Blueprints, Profiles, and Templates as Ontological Commitments

As suggested by the examples above, modalities can be organized into patterns. KEOPS distinguishes three types of ontological patterns (Δ): blueprints, introduced at organizational or conceptual level (intensions), profiles, introduced at project or domain level (extensions), and templates, applied to categories (Figure 9.6).

Since blueprints, profiles, and templates share the same modalities, they can be used to align extensional, intensional, and logical descriptions. Furthermore, because they are discretionary, they help ensure the consistency of commitments made by ontologists, knowledge engineers, and knowledge managers.

9.1.2.3.1 Blueprints

Ontological blueprints are patterns of modalities that address concepts, thus providing an alternative to foundational and meta-model approaches. Blueprints can be intrinsic, functional, or both, depending on the modalities involved. Since modalities are declarative and abstract, blueprints can be elective and tailored to specific contexts and concerns, provided that the modalities are set consistently. While rooted in concepts, blueprints are built from ecumenical modalities, allowing their employment at the domain or project level, provided they don't blur their difference with profiles and, consequently, the distinction between organization and domain-specific symbolic assets.

Cutting to the core, blueprints can first be defined through intrinsic modalities, such as the identity of physical (e.g., ΔAppliance) and symbolic (e.g., ΔRole)

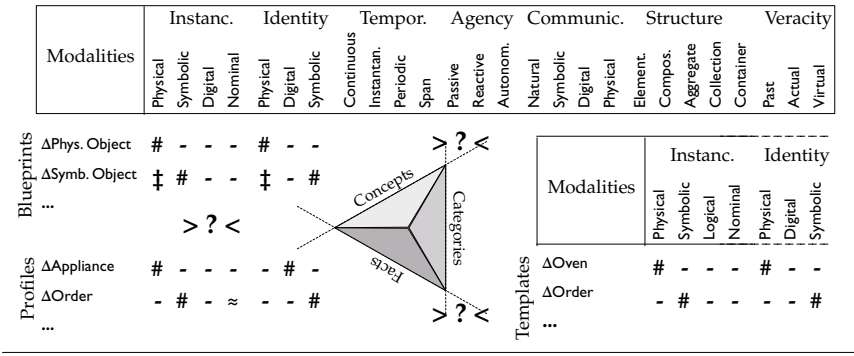


Figure 9.6 Blueprints, profiles, and templates are made of intrinsic (#), functional (≈), exclusion (‡), and unspecified (-) commitments.

intensions. They can also be defined as piecemeal increments characterizing temporality, agency, communication, etc. Such basic blueprints can be used as such or combined into more comprehensive or specific blueprints. For instance, knowledge managers could introduce core blueprints for physical objects and agents with physical identities, and symbolic agents with symbolic identities. Moreover, symbolic agents would be entitled to autonomous agency, and physical agents would have reactive agency.

As noted before, blueprints are not meta descriptions meant to be inherited; rather, they are declarative and discretionary casts intended to be combined. For instance, the ΔRole blueprint could be combined with blueprints for physical agents (typically people), digital agents (typically systems), or symbolic agents (typically organizational units), depending on the context (Figure 9.7).

Along with being discretionary, profiles and blueprints also represent commitments. For instance, the ΔRecipe profile is intended to be realized by instances in symbolic environments. Its symbolic identification is intrinsic (#); it is meant to be virtual and passive (until realized by an actual and active cooking process) and to be a composite continuant that can be accessed in natural language. However, commitments don't have to be exclusive. For instance, a ΔDish profiling could be twofold: one profile relating to symbolic instances for elements in menus, the other relating to physical dishes. They would differ in terms of temporality – continuous for symbolic and time-bound for physical – and in communication, natural language for the former, physical for the latter.

	Modalities	Instanc.				Identity			Tempor.				Agency			Communic.				Structure					Veracity		
		Physical	Symbolic	Digital	Nominal	Physical	Digital	Symbolic	Continuous	Instantan.	Periodic	Span	Passive	Reactive	Autonom.	Natural	Symbolic	Digital	Physical	Element.	Compos.	Aggregate	Collection	Container	Past	Actual	Virtual
Blueprints	ΔRole	-	#	-	-	-	-	#	-	-	-	-	-	-	#	#	-	-	-	-	-	-	-	-	-	-	-
	ΔGoal	-	-	-	#	-	-	-	-	-	-	#	-	-	-	-	-	-	-	-	#	-	-	-	-	-	-
Profiles	ΔRecipe	-	#	-	-	-	-	#	#	-	-	-	-	-	-	#	-	-	-	-	#	-	-	-	-	-	#
	ΔStaff	#	-	-	-	-	-	#	-	-	-	#	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-
	ΔCkg Process	#	-	-	-	#	-	-	-	-	-	#	-	#	-	-	-	#	-	-	#	-	-	-	-	#	-
	ΔOrder	-	#	-	-	-	-	#	-	-	-	#	#	-	-	#	-	-	-	-	-	#	-	-	-	#	-
	§Cook	#	-	-	-	-	-	#	#	-	-	-	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-
	§Dish	#	-	-	-	#	-	-	-	-	-	#	-	#	-	-	-	-	#	#	-	-	-	-	-	#	-

Figure 9.7 Individuals (§) can be directly characterized by modalities, or they can do so through profiles and blueprints.

9.1.2.3.2 Profiles

Profiles go the other way around: like blueprints, they are made of modalities, but their aim is to address domain or project issues. As such, they are best used to characterize domain-specific or siloed representations without preempting decisions regarding representations shared at the organizational or ontological level.

The use of profiles is a two-step routine carried out iteratively: one step involves attributing tentative profiles to selected individuals, while the other step recursively applies the profiles' intrinsic and functional modalities to sets of targeted facts, checking their consistency. For instance, ΔRecipe (symbolic environment, natural language, nominal existence) doesn't match ΔCooking Process, which does not allow for communication in natural language. Ontologists considering individuals (e.g., §Cook or §Dish) are thus presented with three complementary options: they can directly use modalities (Figure 9.7, bottom), combine direct modalities with profiles (Figure 9.7, middle), or even combine them with both blueprints and profiles, provided they are not conflicting.

Using (extensional) profiles offers double-edged benefits: on one hand, they provide early, sound, yet adaptable foundations for ontology development; on the other hand, they can be combined with (intensional) blueprints to align domain-specific ontologies with foundational knowledge architectures, thereby ensuring interoperability across domain-specific ontologies.

9.1.2.3.3 Templates

Like profiles and blueprints, templates are made of modalities and serve two objectives. One is to identify tried and tested representations, and the other is to facilitate their mapping with blueprints and profiles, and consequently, the reuse of categories.

From a broader yet practical perspective, blueprints, profiles, and templates are particularly well-suited to support agreed-upon industry standards for crossroads ontologies due to their declarative, discretionary, and unconstrained nature.

9.1.2.4 Modalities and Patterns Representation

Given their declarative nature, blueprints, profiles, and templates can be directly represented through knowledge graphs (KGs), with more specific representations offering greater possibilities.

9.1.2.4.1 Hypergraphs and Vectors

Hypergraphs are to graphs what metamodels are to models, with a key difference: while metamodels lack a commonly agreed formal definition, hypergraphs are established mathematical constructs that support formal computation. More importantly, hypergraphs form the basis of a wide range of neuro-symbolic

algorithms. With hypergraphs, blueprints, profiles, and templates could be represented by supersets (or hypernodes) characterized by modalities and sets of concepts, facts, and categories.

Like hypergraphs, vectors are well-established mathematical constructs that support formal computation. No less importantly, vectors constitute the bread and butter of generative language models (GenLMs). In the context of ontological prisms, semantic vector spaces would be defined by discrete dimensions corresponding to modality options.

9.1.2.4.2 Web Ontology Language and Prolog

Modalities can be represented as triplets such as {insta.phys.#, ident.symb.#} or {tempo.span._}. As illustrated by the Web Ontology Language (OWL)/Protégé kernel accompanying the book (see Chapter 10), modalities could be directly implemented with OWL. For instance (Figure 9.8):

In Prolog, such triplets would become predicates such as insta(phys,#), ident(symb,#), and tempo(span,_), which could then be combined to form blueprints, profiles, or templates. For instance:

> ***blueprint****(staff, insta(phys,#), ident(symb,#), tempo(span,_), agenc(auto,#), commu(natu,#), struc(atom,#), verac(actu,≈)).*

Compared to the OWL passive representations, the use of the Prolog logical engine would make these representations fully executable, allowing for transparency, traceability, and consistency checks.

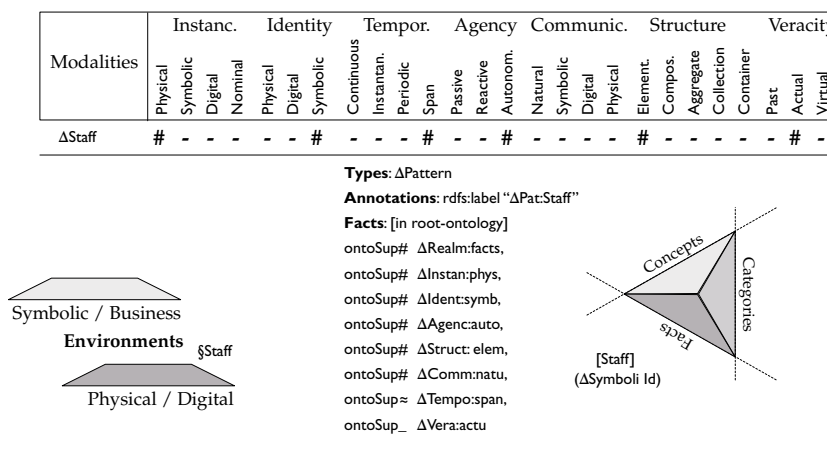


Figure 9.8 Modalities and pattern representation with OWL.

9.1.3 Perceptions: Semantics and Concepts

Perceptions occur when facts meet concepts and signs meet symbols, placing perceptions in symbolic environments. Assuming that names can be consistently assigned to facts within a project's territory, the aim of ontologists is to align the meanings of new terms (for facts) with those that have been agreed upon thus far (for concepts). To that end, the modality-based approach offers a pragmatic alternative to the more formal schemes mentioned in Chapter 3. In addition to their practicality, the declarative nature of modalities allows for configurations tailored to domain-specific needs. For instance, the temporality options could be refined for real-time system ontologies. More generally, ontologies are not models, and they do not have to provide operational guarantees; they only need to offer bounded truth commitments, which can be better expressed through ontological modalities than through inheritance from foundational entities.

Put in an ontological perspective, perception alignments can be carried out at two levels, semantic for the terms used across environments (a), and conceptual for their alignment with the concepts already known (b) (Figure 9.9).

9.1.3.1 Semantic Alignments

Semantic alignments pertain to the resources deemed relevant at the environmental (fact) level. The aim is to ensure that identified elements from data or documents that share a given set of modalities – intrinsic, functional, or both – are defined consistently. In our example, ontologists are considering the semantic alignments between: the meanings of terms used for observed facts, such as dishes, ingredients, and cooking processes; the meanings of terms in regulatory

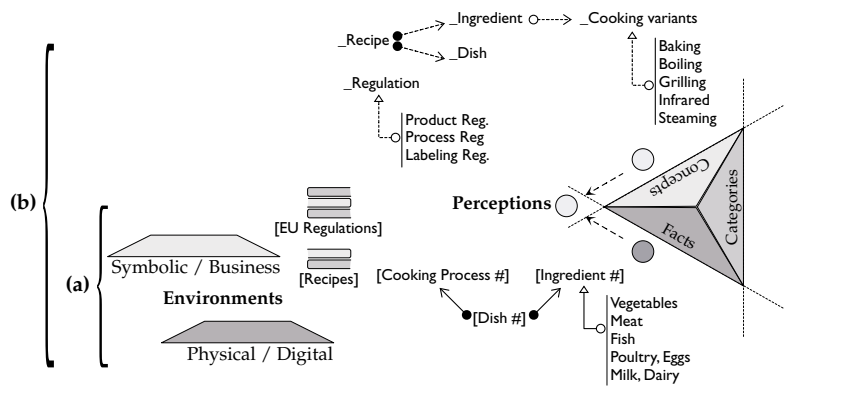


Figure 9.9 How to align meanings within environments (a) and between environments and concepts (b).

and recipe documents; and the meanings of known concepts addressing the same issues. To that end ontologists can use modalities directly or through blueprints, profiles, or templates; alternatively, these elements can be embedded in GenLMs.

9.1.3.1.1 Symbolic Approach

Assuming that knowledge managers are responsible for ensuring the consistency of blueprints and profiles, ontologists can address semantic alignments either directly with modalities or indirectly through profiles and blueprints. As noted above, the objective at this stage is not to finalize definitions but to use modalities to outline the ontological footprints of new elements.

To begin with, ontologists can use modalities to address homonymies (the same term with different meanings) and synonymies (different terms with the same meaning) between ontologies and across levels of representation.

In our first example, ontologists start with a ΔRole blueprint and ΔCustomer and ΔStaff profiles, and must consider three options: §Regular, which refers to customers known personally who pay in cash; [Customer], for managed customers with database records; and 'Customer', as mentioned in regulatory documents. Domain-specific [Customer] complements the ΔCustomer profile with regard to structure (individual customers only) and veracity. 'Customer' in regulations shares the same name but has modalities that cannot be fully aligned in terms of temporality (timespan vs continuous) and structure (elementary vs collective); it appears that the misalignment of modalities between the homonymous document term and profile is due to a distinction between new individual customers and registered ones, possibly corporate. In this case, 'Customer' could be marked as a ΔRole pending further notice, possibly with the introduction of a corresponding _Customer concept. As for the observed §Regular customers, symbolic instantiation and identification modalities should be added to their representation to align it with the [Customer] option (Figure 9.10).

At that point, a comparison between ΔRole, ΔCustomer, and ΔStaff can serve to illustrate the discretionary nature of blueprints and profiles based on modalities. As a blueprint, ΔRole can be unequivocally defined as symbolic and not continuous, while both ΔCustomer and ΔStaff can be expected to have their modalities aligned with ΔRole. Nevertheless, it appears that knowledge managers took a different view and decided on a continuous ΔCustomer (once a customer, always a customer) and a physical ΔStaff (for actual workers on the ground). It follows that, while there may be no homonym profile or blueprint for §Regular individuals, corresponding modalities could be aligned with the ΔCustomer profile, suggesting an upgrade from profile (project level) to blueprint to be shared at the organization (ontology) level. Instead of forcing convoluted lineages from discretionary foundations, this example illustrates how declarative modalities allow for discretionary patterns based on explicit business options.

	Modalities	Instanc.				Identity			Tempor.				Agency			Communic.				Structure						Veracity		
		Physical	Symbolic	Digital	Nominal	Physical	Digital	Symbolic	Continuous	Instantan.	Periodic	Span	Passive	Reactive	Autonom.	Natural	Symbolic	Digital	Physical	Element.	Compos.	Aggregate	Collection	Container	Past	Actual	Virtual
(a)	ΔRole	-	#	-	-	-	-	#	-	-	-	-	-	-	#	#	-	-	-	-	-	-	-	-	-	-	-
(b)	ΔCustomer	-	#	-	-	-	-	#	#	-	-	-	-	-	#	#	-	-	-	-	-	-	-	-	-	-	-
	ΔStaff	#	-	-	-	-	-	#	-	-	-	#	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-
(c)	[Customer]	-	#	-	-	-	-	#	#	-	-	-	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-
	'Customer'	-	-	-	#	-	-	-	-	-	-	-	-	-	#	#	-	-	-	#	-	-	-	-	-	-	#
	§Regular	#	-	-	-	#	-	-	#	-	-	-	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-
(d)	_Customer	-	≈	-	#	-	-	#	#	-	-	-	-	-	#	≈	#	-	-	-	-	-	-	-	-	-	#
	[Regular]	≈	#	-	-	≈	-	#	#	-	-	-	-	-	#	#	-	-	-	#	-	-	-	-	-	#	-

Figure 9.10 (a) Blueprint, (b) profiles, (c) inception, (d) consolidation.

As noted above, modalities and patterns can be directly represented with OWL through triplets and ontological connectors, respectively. Both can be concurrently represented as Prolog predicates and lists. Leveraging the Prolog engine, this representation allows for straightforward consistency checks and adjustments, as well as full transparency (modalities) and traceability (alignments).

Alternatively, the use of triplets and lists points to vectors, and consequently to embeddings and GenLMs.

9.1.3.1.2 Generative Approach

As explained in Chapter 5, GenLMs combine deep learning technologies and natural language processing to explore the content of documents and engage in conversations about them. In addition to their natural language proficiency, GenLMs rely on three pillars: vectors to represent meanings, embeddings to encapsulate, memorize, and reuse contextual elements from specific domains of expertise, and KGs to manage external contexts. With vectors used to represent modalities, embeddings used to represent blueprints, profiles, and templates, and KGs used to manage ontological contexts, GenLMs can help elicit the meaning of concepts in documents and align the labeling of facts accordingly (Figure 9.11).

First, GenLMs would be trained on relevant documentation. Then, blueprints and profiles would be embedded. Finally, ontologists would query GenLMs with observed modalities, such as "Physical, reactive, digital, actual," to obtain candidate profiles.

9.1.3.2 Conceptual Alignments

While semantic alignments operate on meanings at the environment (project) level, conceptual alignments consider these meanings at the ontological (organization)

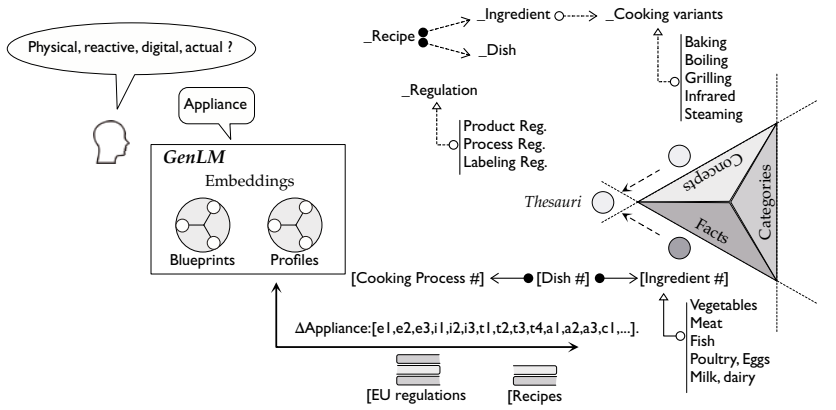


Figure 9.11 Using modality vectors and GenLMs to assess semantic alignments.

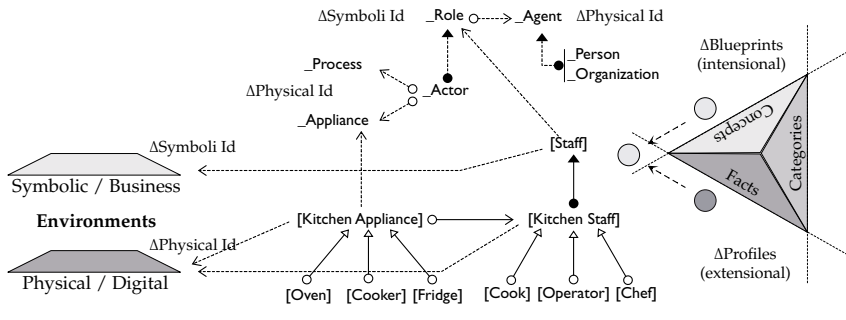


Figure 9.12 Using modalities (Δ) to align extensional and intensional representations (notations in Annex).

level. Using ontological prisms means addressing the consistency of heterogeneous representations that combine facts (extensions) and concepts (intensions). Similar to semantic alignments, this can be done directly with modalities or indirectly through profiles and blueprints.

9.1.3.2.1 Modalities Alignment

Consider, for instance, individual appliances and staff in the kitchen's (extensional) environment. While both can be unequivocally characterized by physical instantiation and identification modalities in the physical environment, the latter, [Kitchen Staff], can also be identified as [Staff] in the symbolic environment (Figure 9.12).

That's not an issue as long as the identity modality is not considered intrinsic, as is the case with KEOPS. However, that's not an option with the instantiation modality, which is deemed to be intrinsic.

That example illustrates the inherent flaw in foundational approaches, as they rely on a two-pronged implicit homogeneity assumption: first, on unequivocal and universal foundational entities, and second, on a shared life cycle (or synchronicity) of ontology projects and knowledge management. It follows that, regardless of business specificities, knowledge managers will have little room for a distinction between physical and symbolic representations, and consequently, between representation levels. As for ontologists responsible for new developments, they will be constrained to align from the outset specific projects with prior representations. That would enforce a one-size-fits-all approach to lineages based on discretionary foundations (staff should be categorized as either perdurants/occurrents or endurants/continuants) and would force heterogeneous content (business management concepts versus kitchen application specifics) into uniform and synchronized representations.

By contrast, ontological prisms and declarative modalities allow for discretionary patterns based on explicit business options. For instance, in the extensional realm, [Kitchen appliance] and [Kitchen staff] represent physical individuals, the former having a physical identity and the latter possessing both physical and symbolic identities. Concurrently, in the intensional realm, _Appliance represents a concept intended to be realized in a physical environment through the [Kitchen appliance] instances with intrinsic (#) digital identification; these instances are intended to be actual, reactive, and composite continuants controlled through digital interfaces. By contrast, _Staff represents a concept meant to be realized in a purely symbolic environment.

9.1.3.2.2 Profiles and Blueprints Alignment

Knowledge managers could introduce blueprints and profiles to support representation commitment, as illustrated by examples from Kitchen project:

- A ΔRecipe blueprint is to describe types realized in symbolic environments, with ID intrinsic (#) symbolic ID, instances intended to be virtual, passive, and composite continuants, and accessed through natural language.
- A ΔAppliance blueprint is to describe types to be realized in physical environments, with intrinsic (#) digital ID, instances intended to be actual, reactive, and composite continuants, and controlled through digital interfaces.
- A twofold ΔDish profile: one version representing symbolic elements in menus, while the other corresponds to physical dishes. They would differ in terms of temporality – continuous for symbolic and time-bound for physical – and in communication, symbolic for the former, physical for the latter.

It should be noted that if the corresponding references across realms do not entail representation constraints, their instantiation and identification modalities must still be consistent. This can be directly verified between _Appliance and [Kitchen appliance], but not so between _Staff and [Kitchen staff] with regard to physical identity; in that case, the alignment is conditioned by the association between _Role and _Agent, the latter with a physical identity modality. Using blueprints and profiles ensures that consistency checks can be performed routinely.

9.1.3.2.3 Applying Profiles and Blueprints

When undertaking the alignment of facts with concepts, it is important to remember the heterogeneity of representation semantics: sets and subsets of individuals for extensions and sets and subsets of meanings for intensions. Furthermore, it should be emphasized that these are commitments to consistent and actionable representations, not articles of faith in some ontological truth. In our example, the key commitment addresses the relationships between agents, roles, and actors (Figure 9.13).

Ontologists begin with observations and intrinsic (#) and functional (≈) modalities (Figure 9.13a). Assuming a new environment and no prior profiles, ontologists can directly compare observed modalities with existing organization blueprints and establish relevant references between extensional and intensional realms. Then, given generic blueprints for _Agent, _Person, _Organization, and _Role, they can rely on associated commitments about symbolic (_Role, _Agent, _Organization) and physical (_Appliance, _Person) identification of realizations (Figure 9.13b).

Profiles are introduced to characterize domain-specific environments. They can extend concepts or blueprints with domain-specific elements or be directly defined

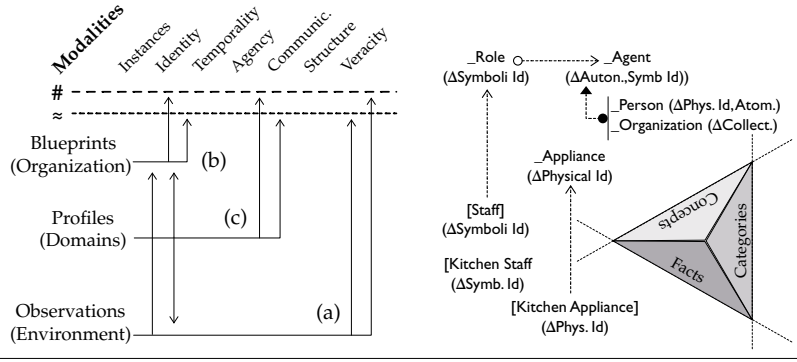


Figure 9.13 Conceptual alignment through blueprints and profiles.

from modalities (Figure 9.13c). In our example, a corporate blueprint ΔBoard (of directors) can be introduced based on the _Role concept, characterized by communication in natural language and a collective structure. Alternatively, a kitchen domain profile ΔCooking can be introduced based on the _Process concept, characterized by real-time execution.

To conclude this segment, it is important to underline the uniqueness of ontological blueprints, profiles, and templates compared to models and meta-models.

- They are based on modalities, which are both ecumenical (independent of domains) and declarative (independent of realizations).
- They represent commitments about finite sets of nonoverlapping modalities.
- They are constructed through composition and consequently do not rely on abstraction or inheritance.

Since modalities are defined independently of domain semantics, related commitments can only pertain to nodes and ontological connectors, without being tied to aspects. It follows that aspects can be introduced iteratively across facts, concepts, and categories, and their semantics defined accordingly.

9.1.4 Experience: Aspects, Definitions, Categories

Perception pertains to the alignment of words and meanings, while experience relates to the ways in which they can be attached to facts through definitions. With KEOPS, definitions are expressed in terms of aspects and the byproducts of classifications. It follows that both aspects and classifications are intended to be established concurrently with perceptions.

9.1.4.1 Definitions and Taxonomies

The pivot introduced by ontological prisms between facts and categories highlights that definitions are not, contrary to the correspondence theory of truth mentioned in Chapter 1, self-sufficient entities. Instead, they are symbolic tools designed for specific purposes.

9.1.4.1.1 The Matter of Definitions

As semantic artifacts situated between words and reality, definitions are more like pontoons than bridges. As mentioned at the beginning of the book, meanings are best understood as scaffolding that braces language to experience. For this reason, definition games are futile, as demonstrated by the plethora of verbose, redundant, overlapping, circular, or conflicting attempts to establish consensual definitions. Instead, an Occam's approach to definitions should meet two straightforward criteria:

- Effectiveness: Applying candidate definitions to targeted individuals must yield clear and unambiguous answers (or mutually exclusive subsets).

- Usefulness: The resulting answers (or subsets) must directly support well-defined purposes.

Such a routine adheres to Occam's razor parsimony principle by producing outcomes that are consistent (i.e., they ensure "internal" truth with no ambiguity), sufficient (they fulfill their purpose), and simple (because mutually exclusive classifications induce less complexity than overlapping ones).

9.1.4.1.2 Thesauri and Taxonomies

Thesauri are ecumenical, attaching names to every nook and cranny of environments and organizing meanings into all-inclusive webs. In contrast, taxonomies are selective, as they consider subsets defined for specific purposes. It follows that definitions in thesauri can be seen as monadic (or unary) nonexhaustive classifications, which could consequently be aligned with a corresponding implicit taxonomy. For example, the definitions of dairy or meat ingredients in a thesaurus would be ontologically irrelevant without being set in relation to taxonomies (Figure 9.14).

The objective of ontologists is to refine the mapping of territories by combining perceptions, materialized through thesauri, and experiences, achieved through taxonomies. With KEOPS, this can be accomplished by carrying out two-pronged iterations: introducing new aspects, probing their relevance with taxonomies, and refining thesaurus definitions.

9.1.4.2 Aspects and Partitions

Facts are initially understood through aspects (or properties), and the objective of ontologists is to organize these aspects into structured entities. However, as illustrated by various attempts at culinary ontologies, even those targeting well-defined and concrete domains can result in discordant representations, allowing only for superficial alignments of character strings. Such divergent outcomes, which

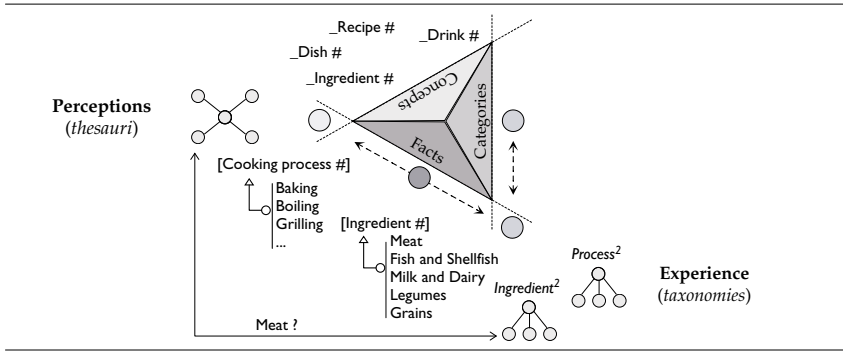


Figure 9.14 Matching monadic definitions in thesauri with comprehending taxonomies.

can be regularly observed, clearly negate the correspondence theory of truth, which assumes an inherent association between the names of concepts and their essence. Instead of targeting presumed universals, KEOPS relies on relevance and modalities as guiding principles.

9.1.4.2.1 Relevance, Powertypes, Aspects

While aspects can be added from a project's inception, introducing too many of them too soon may lead to irrelevant or misattributed ones. This difficulty can be mitigated if aspects are introduced in accordance with concerns through partitions and powertypes. Since partitions and powertypes are designed to align with concepts, they provide a more principled approach to definitions and, in return, can enhance the semantic alignment of aspects across siloed ontologies, such as those related to regions (origin and labeling) or nutrients (health and labeling). In the context of food, concerns could relate to health, baby food, pet food, dietary needs, processing, and more. These domains of interest would translate into partitions and powertypes along with characteristic aspects such as (Figure 9.15):

- Origin of ingredients, animal or vegetal
- Geographical origin of agricultural products
- Details of industrial processes
- Details of cooking processes
- Specific segments such as infants, health conditions, and pets
- Labeling of nutrients, additives, or regions

Definitions will be added concurrently to ensure the consistency of meanings between aspects and partitions.

9.1.4.2.2 Modalities and Aspects

The relevance approach used to introduce aspects, which is driven by domains of interest, can be combined with one framed by blueprints and profiles, with guidelines provided by modalities, as illustrated in Table 9.1.

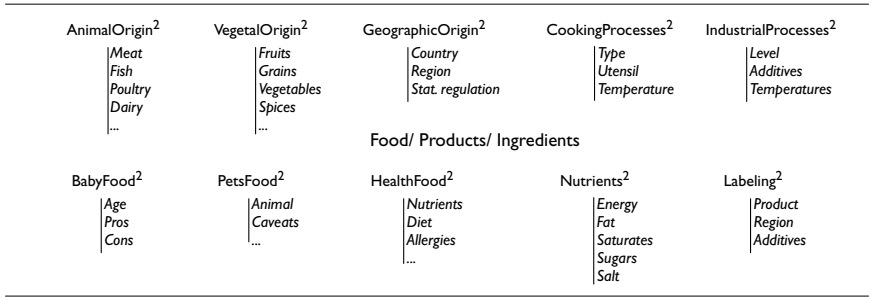


Figure 9.15 Examples of powertypes and associated aspects.

Table 9.1 A sample of aspects accompanying modalities.

Modalities						
Instantion	**Identity**	**Temporality**	**Agency**	**Communic.**	**Structure**	**Veracity**
Type	Type	Time frame	Roles	Semantic layers	Type	Context
Abstractions	Context	Timestamps	Operations		Elements	Ownership
Realizations	Rankings	States	Authorizations	Languages	Cardinalities	History
...	...	...	...	Channels	...	...
				...		

- Instantiation modalities entail references to concepts, such as [Dish] to _Recipe, or realizations, such as _Food to [Ingredient].
- Identification entails references to context, which may include organizational or mechanical aspects, as well as potential rankings.
- Temporality is associated with time frames, time stamps, and possibly states.
- Agency is characterized by roles, operations, and authorizations.
- Communication is characterized by languages, semantic layers, and channels.
- Structures are accompanied by operators, elements, and cardinalities.
- Veracity is determined by context, ownership or authorship, and history.

Leveraging domain-specific relevance and ontological modalities to identify aspects and associate them with entities brings significant benefits in terms of focus and consistency.

9.1.4.3 Aspects and Schemas

Pun notwithstanding, knowledge grows at the edges, where new facts can be elucidated by concepts or explained by existing categories. On that account, many domain-specific ontologies are better understood as conceptual models. By taking advantage of ontological prisms, ontologists can explore potential alignments between their remit and existing ontologies. This can be achieved through thesauri (for concepts) and taxonomies (for categories).

9.1.4.3.1 Thesauri and Taxonomies

Taking a cue from Archimedes, definitions constitute the fulcrum of ontologies, enabling changes in knowledge. However, this can only be achieved if definitions are flexible and adaptable in relation to concepts (abstraction) and categories (realization). At this point, it must be remembered that entries in thesauri pertain to meanings, while those in taxonomies pertain to individuals. Matching one with the other is at the core of an ontologist's job description. For instance, the definitions of cooking processes (thesaurus) must be congruent with the

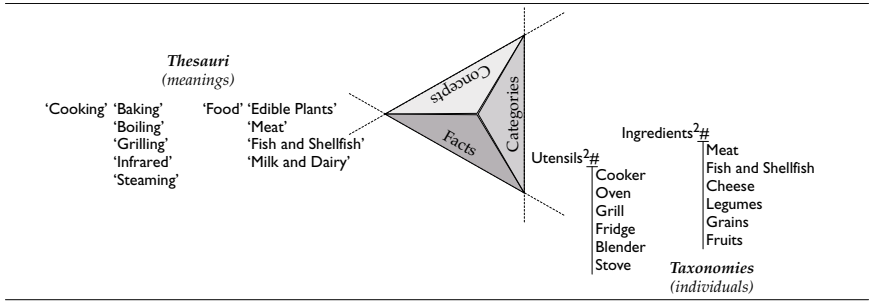


Figure 9.16 Ensuring the consistency between the meaning of concepts and the classification of individuals.

classification of cooking appliances (taxonomy); likewise, there should be congruence between food concepts and cooking ingredients (Figure 9.16).

Given consistent definitions of concepts and classifications, the objective is to compare these classifications with existing (or legacy) categories as represented by database schemas. More generally, the objective would be to align semantic spaces defined by taxonomies with semantic vectors used by language models to represent meanings.

9.1.4.3.2 Taxonomies and Schemas

Since relational databases are a de facto standard for structured data representation, they can be viewed as "flat" or 2D representations of knowledge. Relational databases use schemas to represent individuals as tuples of properties, some of which form identifying keys, with normalization ensuring that non-key properties are fully determined by keys. As illustrated by a number of tools, attempts to align the facts under consideration with known categories should rely on potential matches between these facts and normalized relational schemas. Then, using the routine introduced in Chapter 8, ontologists will examine the alignment of relevant taxonomies with the schemas of existing powertypes. Successful alignments could then be used to translate flat schemas into ontological categories. For instance (Figure 9.17), an empirical taxonomy of kitchen utensils may reference an existing category of kitchen appliances (Figure 9.17a), suggesting a potential alignment with the cooking function power type (Figure 9.17b). Since that alignment is not possible, subtypes are introduced instead (Figure 9.17c), and the reference to the cooking function power type is updated (Figure 9.17d).

It should be noted that such changes to categories are tentative steps in the iterative and incremental KEOPS process. Their consistency with ontological truth commitments is still to be verified.

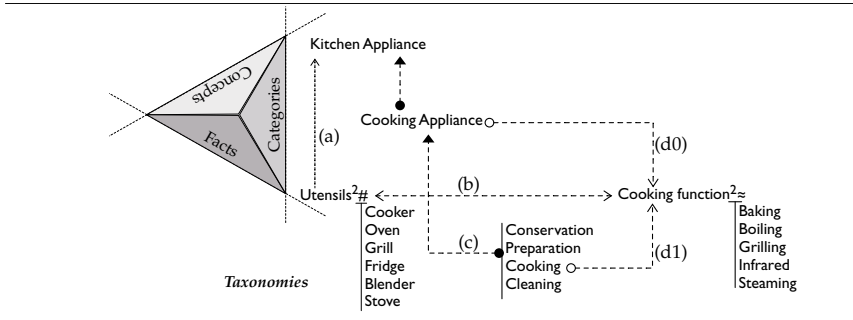


Figure 9.17 Empirical taxonomy: reference to category (a), potential match with power type (b), update of category (c, d).

9.2 Mapping Intensions

Contrary to models, which are meant to represent elements within specific and bounded contexts and concerns, ontologies are agnostic about reality. This absence of actual boundaries must be compensated for by discretionary truth models and explicit commitments. Such models are made of truth-bearers (nodes) and truth-makers (connectors), which are subject to three kinds of commitments: semantic, with regard to environment perceptions, conceptual, with regard to the consistency of intensions, and logical, with regard to the consistency of projections (Figure 9.18).

A number of methods, formal or functional, exist to ensure the consistency of commitments related to semantically homogeneous graph representations, such as Shapes Constraint Language for checking that data definitions are aligned with rules, and the Prolog engine, which allows running first-order logic on representations. Our aim here is to use modalities as ontological commitments to address the overall consistency of heterogeneous representations.

9.2.1 Ontological Commitments

9.2.1.1 Conceptual vs Ontological Commitments

Conceptual commitments pertain to the internal consistency of concepts independently of their interoperability with facts or categories. Taking, for example, an ontology supporting compliance with EU regulations, the primary objective of ontologists would be to translate EU documentation into a set of conceptual anchors such as _EU Regulation#, _Product#, _Origin#, _Process#, or _Label#. Given that even standalone projects targeting siloed ontologies and using homogeneous representation semantics are facing scalability barriers, managing the exponential complexity of continuous and unbounded developments across multiple domain-specific

ontologies, while using heterogeneous representation semantics and ontological commitments, appears to be a sine qua non of knowledge engineering.

The KEOPS approach is two-tiered: the first tier (modalities) relies on declarative and discretionary primary commitments about a finite number of foundational issues; the second tier (blueprints, profiles, templates) organizes primary commitment into trustworthy, reusable patterns carried out across heterogeneous realms.

Primary commitments regarding modalities can be carried out across representation levels according to ontological hierarchies, with intrinsic modalities being retained along intrinsic hierarchies and becoming functional along functional hierarchies. For instance, if physical identity is defined as intrinsic to cooking appliances, then physical identity should also be intrinsic to subsets (or subtypes) such as ovens. Alternatively, if an elementary structure is intrinsic to legal entities, that structure is only a functional option for functional subsets (or subtypes) such as enterprises (Figure 9.19).

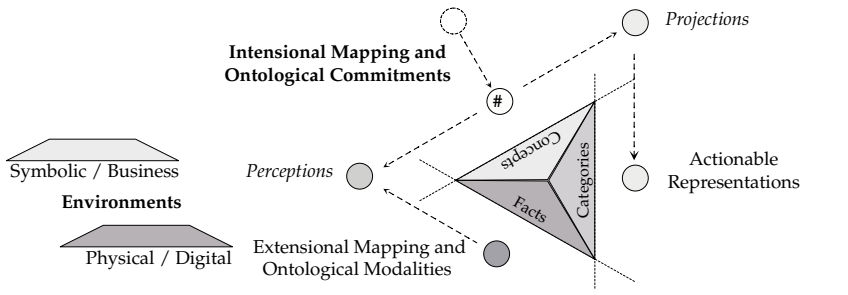


Figure 9.18 Conceptual commitments ensure the consistency of intensional and extensional mappings.

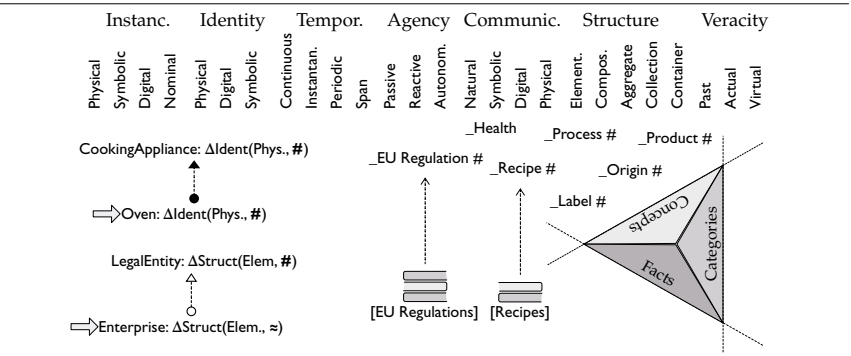


Figure 9.19 Using modalities as ontological commitments between representation levels.

However, such commitments can only be made along ontological connectors, i.e., between levels of homogeneous representations. Otherwise, e.g., between concepts and documents pertaining to EU regulations or recipes, commitments will rely on blueprints and profiles.

9.2.1.2 Heterogeneous Commitments

Symbolic perception commitments formally associate percepts with intensional meanings, whether they are newly introduced (e.g., _Organic Label) or preexisting (e.g., _Recipe).

The point about regulatory classifications is that, being fully documented, they belong to both extensional (facts) and intensional (concepts) realms. However, since the content of documents, regardless of their origin, does not come with trustworthy ontological commitments, knowledge managers must take on this responsibility. In our example, further ontological commitments are required regarding the EU classification of products in general, as well as for specific categories such as cheese and wine. Moreover, the organic labels introduced with the new line of organic prepared meals (Ecologic, Biologic) must be aligned with EU organic regulations (Farming, Food).

9.2.1.3 Instanciation, Identification, Structures

Given a semantic alignment between the newly introduced symbols and the existing concepts, the role of knowledge managers is to revisit and possibly improve the entire set of conceptual commitments. That can be achieved at two levels, literal and symbolic, the former attaching signs to facts, the latter symbols to signs (see Chapter 5 about signs and symbols).

Signs are percepts permanently attached to facts and can thus be interpreted directly, independently of contextual semantics. In contrast, the meaning of symbols is determined in reference to contextual semantics. For instance, in the case of EU organic and labeling regulations, compliance with PDO entails literal commitments regarding labels such as 'Camembert' or 'Château Margaux', which must be attached to extensional instances (e.g., [cheese] or bottles of [wine], respectively). This is in contrast to symbols (e.g., the _Organic label), which are intended to be attached to intensional or logical entities. Blueprints allow for explicit commitments and thus enhance conceptual consistency. For instance (Figure 9.20):

- An ΔPDO blueprint characterizes elementary nominal instances (literals) without separate identity (e.g., 'Camembert', 'Château Margaux').
- A ΔQuality label blueprint characterizes symbolic instances with symbolic identity with possible composite structure (e.g., _Organic Label, _Organic Recipe).
- A ΔFluid blueprint characterizes containers of uncountable liquid things with functionally defined span temporality and reactive agency (e.g., [Wine]).

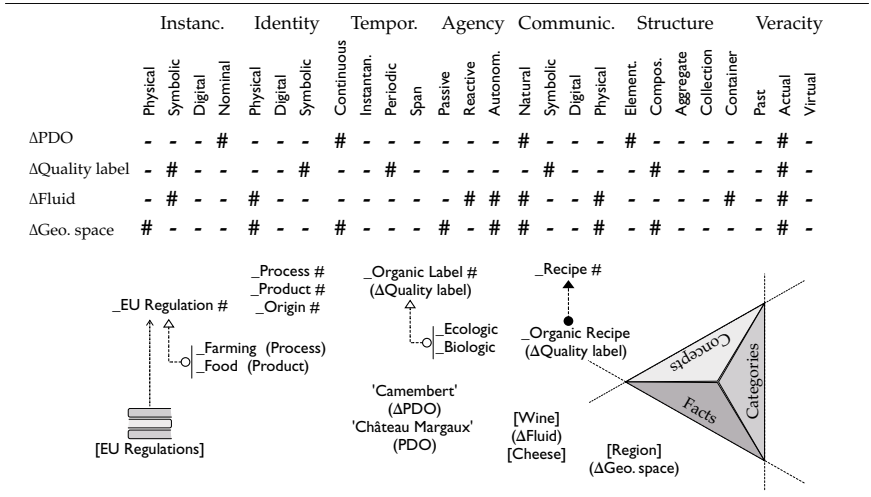


Figure 9.20 Blueprints addressing instantiation, identification, and structures.

- A ΔGeo. Space characterizes composite structures with physical instances and identity, supporting symbolic communication and functionally defined veracity (e.g., [Region]).

9.2.1.4 Agency and Communication

The pervasiveness of GenAI technologies, combined with their opacity, has made agency a critical issue at both the organizational and operational levels. For the former, the issue revolves around who is entitled to do what – people or systems. For the latter, it concerns what actions are being taken, such as observation, reasoning, judgment, or decision-making. It follows that building so-called agentic AI architectures requires explicit commitments regarding agency and communication capabilities that span the realms of facts, concepts, and categories. A sample of tentative blueprints illustrates the issue (Figure 9.21).

The ΔAgent blueprint can serve as a root commitment, mandating only functional modalities regarding instantiation, identity, agency, and communication. Next comes the critical distinction between ΔPerson (intrinsic physical instances and identity, autonomous agency, natural language communication), ΔOrg.unit (intrinsic symbolic instances and identity, autonomous agency, natural language communication), and ΔSystem (intrinsic symbolic instances and identity, reactive agency, symbolic communication). Finally, two more specific blueprints: ΔDevice (physical instances without autonomous agency or natural language communication) and ΔAI Agent (digital instances and identity with potential functional autonomous agency and natural language communication).

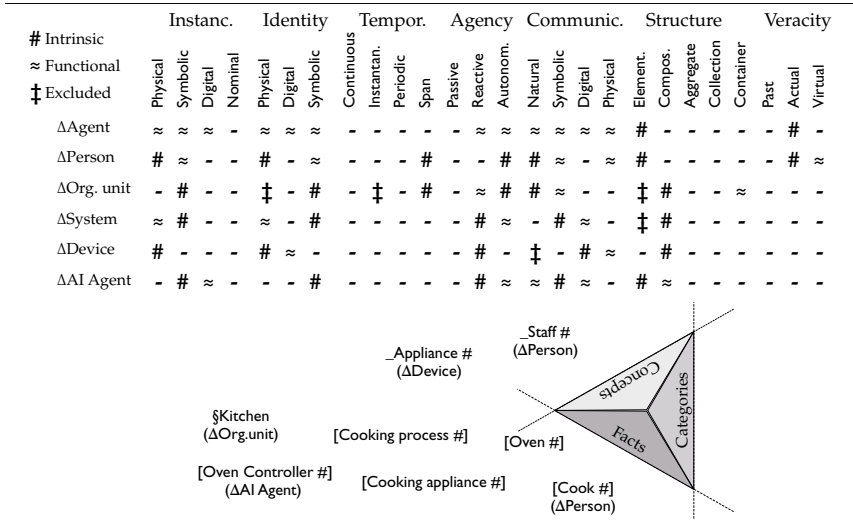


Figure 9.21 Some tentative blueprints about agency and communication.

9.2.1.5 Veracity

As it is often said, stories shape the world; hence, the benefit of introducing explicit veracity commitments in heterogeneous ontological representations. When ontological veracity is considered, anything can happen at any time, in any realm, regardless of instantiation or identification modalities. This can be illustrated by representing "Babette's Feast," an iconic movie by Gabriel Axel, which is based on a story by Karen Blixen (Figure 9.22).

Starting with heterogeneous environments: individuals of digital (the movie) and physical (the book) artifacts (a). Both are managed individuals (§) embodied in instances of corresponding categories, with references to directors and authors as managed categories (b). The story itself, as well as the Dinner sub-story, appears as instances (§) in the symbolic environment (c) of the _Fiction concept (d). The meal portrayed on paper and screens appears as a virtual instance (§) of the _Feast concept, and the 'Café Anglais' mentioned in the story, while a nominal fact in the fiction, is a reference to a real place in Paris when the story was first published. Concomitantly, the Meal and Story categories make references to corresponding concepts.

This example may appear a fiction too far, yet it bears direct business relevance for competitive strategies or the protection of intangible assets. In our example, the blueprint ΔInt.prop. addresses intellectual property for facts as well as categories; the ΔAuthor profile directly addresses (no concepts) the managed

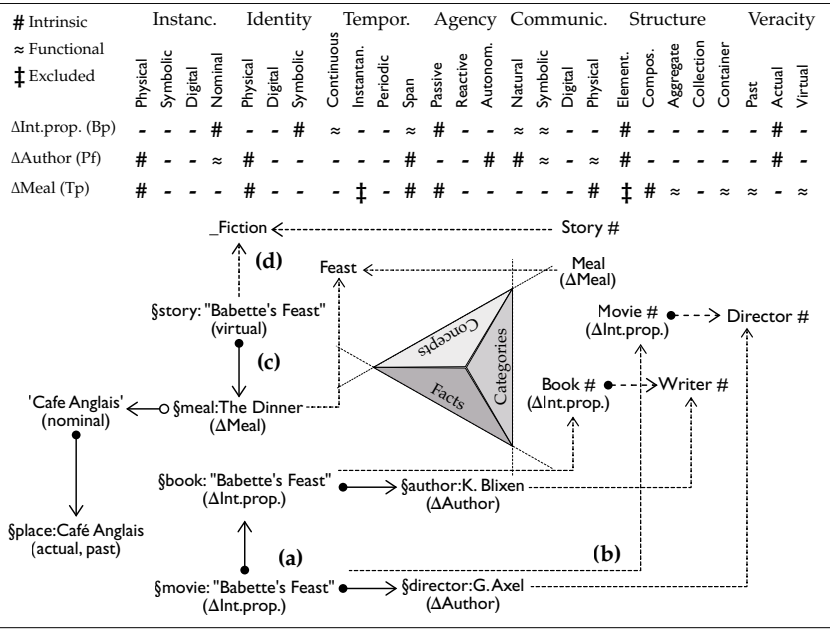


Figure 9.22 Representing *Babette's Feast* fictional reality through blueprints (Bp) and profiles (Pf).

representations (categories) of movie directors and book writers; and the template ΔMeal addresses the representation of the eponym category.

9.2.2 Projections and The Mapping of Time

Projections characterize the temporal, organizational, and architectural intervals between concepts and their realization into shared categories.

9.2.2.1 Ontological Timescales

Ontologies are, by nature, synchronic, reflecting knowledge of a given context at a specific time. In contrast, knowledge is inherently diachronic, reflecting galaxies in continuous expansion. To avoid becoming detached and obsolete, ontologies must be underpinned by some temporal connection to knowledge.

A psychological (individual) understanding of temporal knowledge was briefly mentioned in Chapter 5 with Fodor's Language of Thought hypothesis, which associates mental representations with cognitive events. More generally, the distinction between physiological and cognitive time frames appears to align more closely with the extensional and intentional realms of ontological prisms, respectively.

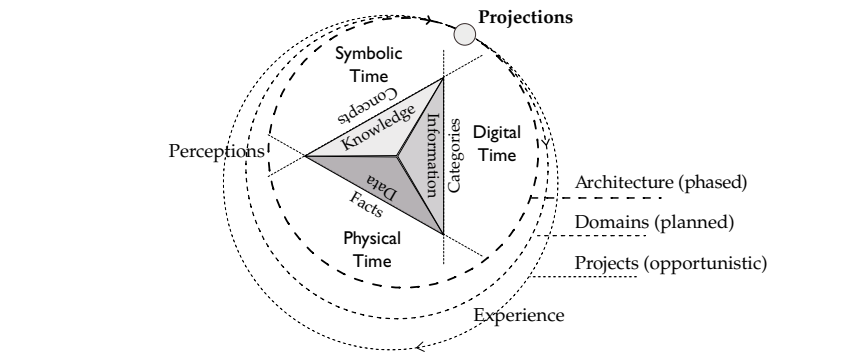


Figure 9.23 Ontological time and projections.

This individual perspective can be transposed to what Stafford Beer referred to as the Brain of the Firm, which, in our case, involves ontological prisms. Actionable knowledge would thus be situated in three temporal dimensions:

1. A continuous, movie-like physiological time flow is established by events that underpin instant and timeless operations, which can be carried out without entailing commitments or accountability.
2. Cognitive time capsules for reasoned operations that can be carried out by committed individuals regardless of knowledge externalities.
3. Collective time frames for reasoned operations involving knowledge externalities that must be carried out by committed organizational units.

These levels correspond to physical time (external events), symbolic time (intentions), and digital time (managed representations), governing extensional, intensional, and logical realms, respectively (Figure 9.23).

With ontological prisms, projections can thus be organized at three levels: architecture, domains, and projects.

9.2.2.2 Architecture Projections

Architectural projections relate to the realization of concepts as categories. From an ontological perspective, the objective is to translate parthood semantics, which are used to represent conceptual abstractions, into object-oriented semantics, which are employed for category abstractions.

Assuming that anchors have been aligned through references, such as _Staff # (concept) and Staff # (category), a naive option would be to introduce a DinerStaff intrinsic subtype with functional subtypes that mirror the conceptual subsumptions of the _Kitchen role. However, while concepts for cook, chef, manager, or waiter can be defined independently without direct drawbacks, doing the same

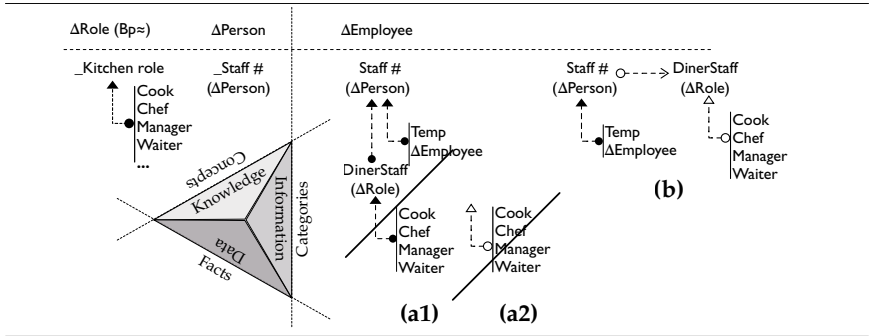


Figure 9.24 Blueprints, templates, and the separation of concerns.

with the corresponding categories would create exclusive roles, which would significantly affect the value of managed categories. In our example, given the intrinsic Staff # hierarchy, instances of cook, chef, manager, and waiter would need to be uniquely identified (Figure 9.24a1).

Introducing the intrinsic blueprint ΔPerson, the functional blueprint ΔRole, and the intrinsic template ΔEmployee provides guardrails for projections. Intrinsic patterns (blueprints or templates) encompass the identity principle, which means that when the same pattern is used across realms (e.g., ΔPerson), the principle is intended to apply uniformly to intrinsic structures, regardless of representation semantics. In contrast, functional patterns do not entail such a commitment, allowing more latitude for knowledge engineers. For instance, if the DinerStaff category is committed to the ΔRole functional blueprint, it leads to a functional specialization. However, that second option (Figure 9.24a2) may still raise some feasibility issues.

While blueprints address the consistency of concept realizations, templates focus on their feasibility, yet allowing for cross-realm commitments, as illustrated by the blueprint ΔRole applied to the DinerStaff category. Nevertheless, a distinctive use of blueprints and templates aims to maintain the separation of intrinsic and functional concerns. The former pertains to identities and structures, regardless of domain semantics, which are intended to be addressed by the latter. In our example, the separation of concerns is compromised by the overlap between the ΔRole functional blueprint and the ΔEmployee intrinsic template, both of which share the same ΔPerson intrinsic blueprint. As a result, the functional realization (DinerStaff) would be committed to different identification patterns, either ΔPerson or a combination of ΔPerson and ΔEmployee.

Assuming object-oriented representation semantics, the separation of concerns can be secured by replacing functional inheritance with delegation. With KEOPS, the same outcome can be achieved by keeping intrinsic (#) and functional (≈) lineages separated (Figure 9.24b).

9.2.2.3 Domains Projections

Domain projections are governed by knowledge engineering processes and the status of relevant resources and assets. They are intended to address domains related to heterogeneous representations of shared and boundless ontologies. Domain projections must be organized to ensure the consistency of conceptual and categorical realms, with the former focusing on objectives and the latter on realizations. Compared to systems engineering processes, which are often contingent on external business time frames, knowledge engineering processes can be better aligned with internal time frames, even when external constraints must be considered. An example in our culinary context would be the introduction of new domains that support compliance with EU regulations.

Concomitantly, organizational projections will address management issues, such as the assignment of intrinsic and functional responsibilities regarding concepts and categories between organizational units. Assuming that the corresponding commitments do not infringe upon internal factors related to knowledge engineering, the phasing of projections could also take into account external considerations. For instance, a merger between enterprises would lead to the consolidation of certain departments and, therefore, domains.

9.2.2.4 Project Projections

Project projections are initiated by ontologists and project managers; they are intended to address endogenous representations of well-defined ontologies. While such projections may induce changes across aspects, categories, or even domains, they are not intended to affect users outside the projects being considered. These projections are conducted on a continuous basis, depending on the deliverables of projects and the availability and status of the relevant resources and assets. For instance, in our culinary context, we could have a pilot project aimed at exploring a new dietary segment.

9.3 Actionable Knowledge

Compared to Chapter 8, which has taken an outward perspective focused on the use of ontological prisms in systems engineering, this chapter takes an inward view of the engineering of knowledge itself. This segment concludes with the actionability of knowledge engineering results (Figure 9.25).

While systems engineering processes aim to deliver packaged products in stable versions, knowledge engineering deliverables are rarely structured and are inherently subject to change and expansion in both content and scope. When used to support systems engineering, this issue can be managed by leveraging the distinction between information, data, and knowledge within ontological prisms,

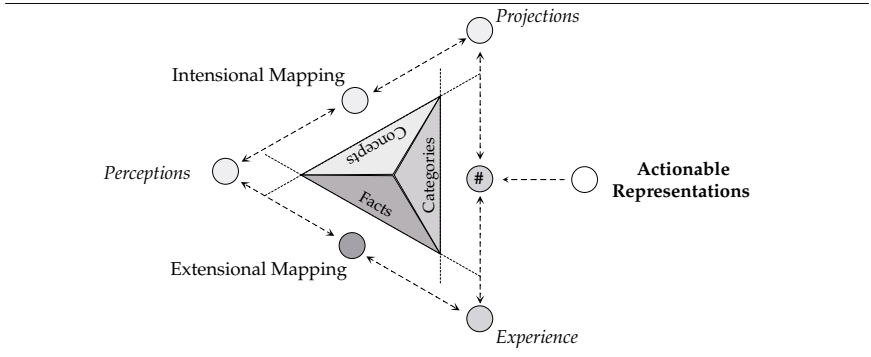


Figure 9.25 Actionable knowledge.

allowing them to be managed independently. However, this is not an option for the engineering of heterogeneous knowledge, which must continuously ensure the consistency of facts, concepts, and categories, as well as the operational integration of experiences, perceptions, and projections. Taking into account the amorphous nature of knowledge engineering outcomes, KEOPS adopts a reverse perspective and defines the results of knowledge engineering through use cases.

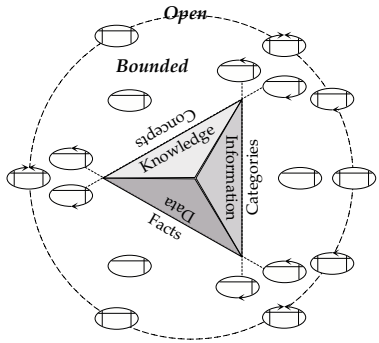
9.3.1 Ontologies Use Cases

9.3.1.1 Principle

As far as enterprises are concerned, knowledge must serve purposes, which means that representations must be actionable. However, since purposes are by nature specific, they entail fragmented perspectives. These perspectives can be characterized depending on their scope regarding domains (bounded or open) and realms (homogeneous or heterogeneous). This dual taxonomy can be used to define ontological use cases with regard to scope and purpose:

- Bounded domains ensure built-in consistency of words (facts) and meanings (concepts), while unbounded ones do not.
- Homogeneous realms ensure built-in consistency of abstraction semantics, while heterogeneous ones do not.

When systems are seen as black-boxes, user perspectives are typically represented through use cases (see Chapter 8). Use cases can also be applied to KMS, but with a proviso: as noted above, ontologies pertain to symbolic artifacts, not digital ones. It follows that operational requirements can be replaced with symbolic commitments.



	Domains	
Realms	Bounded	Open
Homogeneous	BH	OH
Realization	BR	OR
Abstraction	BA	OA

Figure 9.26 Ontological prisms use cases.

9.3.1.2 Taxonomy

Based on the footprint of use cases on ontological prisms, six basic blueprints are suggested, which can be further characterized by level, intrinsic (#) or functional (≈) (Figure 9.26).

9.3.1.2.1 Bounded Use Cases

Bounded ontological prisms use cases (OPUCs) are meant to refer to siloed ontologies, set by facts (data), categories (information), or concepts (knowledge) identified upfront.

- Bounded homogeneous (BH) OPUCs have their footprint contained within the same realm.
- Bounded realization (BR) OPUCs aim to realize bounded conceptual or logical domains.
- Bounded abstraction (BA) OPUCs aim to abstract from bounded territories or domains.

9.3.1.2.2 Open Use Cases

Open OPUCs can refer to multiple domains, which may or may not, identified upfront.

- Open homogeneous (OH) OPUCs have their footprint straddled across different domains, all set within the same realm.
- Open realization (OR) OPUCs aim to realize open conceptual or logical domains.
- Open abstraction (OA) OPUCs aim to obtain higher levels of representation from open territories or domains.

These ontological use cases provide guardrails securing shared and distributed access to heterogeneous ontologies.

9.3.2 Use Cases and Ontological Commitments

Ontology use cases are better understood as declarations of intent rather than requirements, primarily because ontologies are not models and actual engineering is not at stake. Since they pertain to ontologies, such use cases can be defined as commitments regarding the kind of changes being considered. This shift from requirements (an imperative approach) to commitments (a declarative approach) is at the core of issues related to transparency, traceability, and accountability, as illustrated by a representative sample of committed and uncommitted use cases (Figure 9.27).

9.3.2.1 Committed Use Cases

Committed use cases are defined in relation to ontological prisms, or more generally, with knowledge architecture.

9.3.2.1.1 Homogeneous Commitments

Homogeneous commitments are intended to be carried out within the same realm, whether that involves concepts (e.g., siloed ontologies), categories (e.g., system modeling), or facts (e.g., data and process mining). These commitments will establish intended boundaries, which will be enforced by ontological prisms with built-in guardrails. For instance, process mining could be conducted within specific domains (BH), while data mining and clustering would target still unbounded datasets (OH).

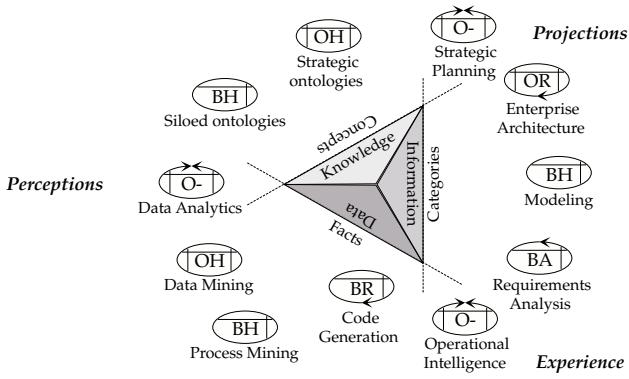


Figure 9.27 Representative OPUCs.

It should be noted that, given the built-in two-pronged ontological distinction between intrinsic and functional connectors on one hand, and syntactic and semantic constructs on the other (see Chapter 2), consistency commitments apply to homogeneous use cases, regardless of domain-specific conditions.

9.3.2.1.2 Realization Commitments

Realization OPUCs straddle realms, aiming at facts (derived from concepts or instantiated from categories) or categories (defined from concepts). Ontological prisms will rely on profiles and blueprints to assist ontologists in their endeavors. Code generation, such as Vibe coding, epitomizes BR OPUCs. Alternatively, OR OPUCs will be introduced when different domains must be taken into account (OR), for instance, process management that overlooks the operational realization of system categories.

9.3.2.1.3 Abstractions Commitments

Abstraction OPUCs straddle realms, aiming at concepts or categories (both abstracted from facts) or at concepts that are abstracted from categories. Requirements analysis, which assumes an explicit commitment to domains, typifies BA OPUCs. Alternatively, causal chain analysis, which allows for indeterminate root causes, is better associated with OA OPUCs.

9.3.2.2 Uncommitted Use Cases

Uncommitted use cases are of two kinds: explanatory, for obvious reasons, and functional.

9.3.2.2.1 Explanatory Use Cases

Exploratory use cases are not intended to make changes to actionable knowledge and, consequently, don't entail active commitments. They are designed to address unknown unknowns, where boundaries can be pushed simultaneously across domains through abstraction and realization. Exploratory use cases gain special relevance with the pervasiveness of GenAI capabilities, which combine pattern matching (for abstraction) with generative capabilities (for realization). Such use cases (O-) are positioned where realms intersect, as exemplified by GenLMs (for documents), data analytics (perceptions), planning (projections), and operational intelligence (experience).

GenLMs (see Chapter 5) operate on documents by treating tokens (textual or visual) as facts and attempting to extract emerging concepts. In this context, the back-and-forth exploratory moves between recorded realizations (documented facts) and emerging abstractions (meanings) are embodied in deep learning algorithms.

Data analytics operates on the working assumption that perceptions are partial and biased, addressing these issues through stochastic analysis that combines

discriminative and generative statistics. Discriminative models focus on finding correlations through the segmentation or clustering of datasets (facts), while generative models aim to build causal relationships that could explain them (concepts).

Strategic planning operates on projections involving third parties whose intentions are, by nature, unknown, thereby undermining the relevance of projected commitments. In this context, the aim of exploratory use cases is to identify patterns of intent based on perceptions and to combine them with hypothetical projections.

Operational intelligence operates on experience by integrating dynamic models (categories) with data and process mining (facts). Traditional approaches have focused on the real-time monitoring and control of business processes, along with automated decision-making routines. However, the prevalence of GenAI technologies raises questions about the transparency, reliability, and accountability of automated decisions made in real time. From an ontological perspective, the issue lies in the status of neuro-symbolic knowledge, which is literally embodied in the distinction between operations (systems) and intelligence (ontologies). This issue can be addressed by combining explanatory use cases that are unconstrained by ontological commitments with functional use cases that are committed to cognitive operations.

9.3.2.2.2 Functional Use Cases

While ontological use cases are established in relation to knowledge architecture, functional use cases are defined by the nature of the functions and operations encompassed by knowledge engineering processes. Cognitive functions are set across ontological realms and are intended to shape user experience according to cognitive perspectives such as discovery, experiencing (or practical exploration), learning, teaching, reasoning, and decision-making. These functions are meant to be presented through interfaces designed by organizations (Figure 9.28).

Built-in functions are predefined transverse operations designed to enhance transparency and traceability. Edit operations are primitively focused on ontological components: facts, concepts, categories, aspects, blueprints, and profiles. They enable basic create, read, update, delete operations, accompanied by guidelines and consistency checks. These operations can be initiated directly or through other use cases. Built-in functions and edit primitives are intended to be realized through libraries, which can be activated directly or from OPUCs.

9.3.2.3 Roles and Environment

Finally, in line with the principles set forth in Chapter 4, the aim of KEOPS is to leverage the existing organization of remits and roles as much as possible. General remits focus on the management of shared resources and assets, including data

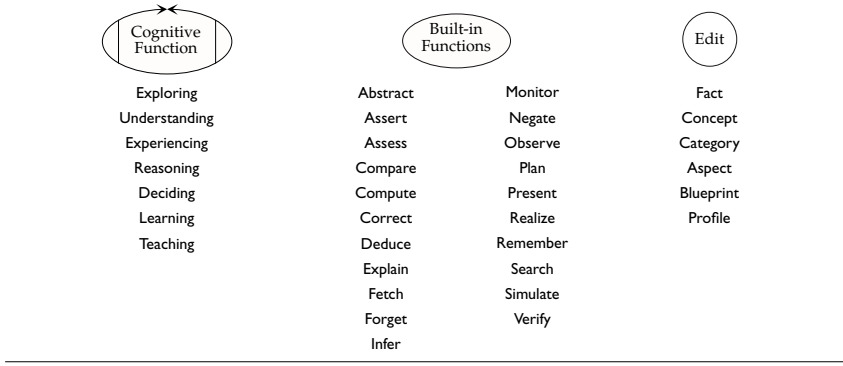


Figure 9.28 A sample of cognitive and functional use cases.

and document managers (facts), knowledge managers (concepts), enterprise architects, and systems managers. Dedicated remits concentrate on the interoperability of shared resources and assets, such as domain managers, thesaurus managers, and ontologists.

A KEOPS kernel prototype has been developed as a proof of concept using the OWL with the Protégé platform. This allows for both online and on-site standalone utilization, as well as seamless portability for projects. The details of that kernel will be provided in Chapter 10. Moreover, by leveraging the native OWL/Resource Description Framework implementation, interfaces can be developed to either populate ontological realms, using Extensible Markup Language, Structured Query language, or Unified Modeling Language, or to extend their reach, such as with GenLMs or Prolog.

Key Points

KEOPS offers principled yet pragmatic solutions to central software engineering issues that become critical when considering the engineering of knowledge, as well as to challenges specific to knowledge development.

First, the goal is to free ontologists from procedural approaches, which are a significant hindrance in software engineering and have become a major obstacle to the engineering of knowledge. The KEOPS solution is twofold, iterative, and incremental. On one hand, developments iterate through extensional, intensional, and logical realms. On the other hand, intrinsic and functional threads are woven into the knowledge fabric.

Equally challenging is how to address abstractions, which is a critical issue for the integration of foundational and domain ontologies without resorting to

cumbersome and ineffective Rube Goldberg machines. The KEOPS answer is to introduce ontological modalities, a kind of declarative stereotypes associated with ontological intrinsic and functional qualities. Ontological modalities can be further combined into blueprints (intensions), profiles (extensions), and templates (categories). This allows for the replacement of imperative inheritance of convoluted lineages with declarative and purpose-built patterns based on explicit business options.

The third major issue addressed by KEOPS is specific to the engineering of knowledge as it relates to the unstructured, porous, and shifting nature of deliverables. The KEOPS solution is to revisit use cases in relation to their impact on knowledge architecture, and more specifically, ontological prisms. Actionable knowledge can thus be framed by the nature of commitments made by use cases: committed (domain-specific or unbounded, homogeneous or heterogeneous) or uncommitted (exploratory or functional).

Finally, KEOPS resets the engineering of knowledge from the perspective of agentic architectures that support the collaboration of AI agents. Mirroring the procedural issue raised above, collaborating agents cannot operate on traditional layered stacks, which are structurally sequential. Instead, agentic collaboration must rely on conversational loops, which combine direct conversation (experience), integration of contexts (perceptions), and anticipations of further developments (projections).

10

KEOPS OWL Kernel

Architecture begins where engineering ends

Walter Gropius

This book begins with the philosophical and semiotic background of ontologies, spends some time on the mechanics of ontological prisms, focuses on their language and cognitive capabilities, and concludes with their benefits for knowledge engineering. This closing chapter aims to demonstrate the feasibility of the entire approach through a Web Ontology Language (OWL)/Protégé kernel.

10.1 OWL/Protégé Kernel

10.1.1 Overview

In line with the objectives of ecumenism and parsimony of ontological prisms, the KEOPS kernel relies solely on core OWL features, leaving as much room as possible for users' customized extensions without the risk of confusing overlaps. Consequently, the overview of the online OWL/Protégé addresses only the relevant features.

10.1.1.1 Protégé Class Hierarchy

Protégé entries are organized into nodes and edges, which are presented as a hierarchy of classes under the root owl: Thing. The KEOPS kernel introduces two significant distinctions: on one hand, it introduces Aspect to represent edges with properties, making a step toward labeled property graphs; on the other hand, the KEOPS kernel does not assign any particular meaning to the Protégé hierarchy, except for the management of entries, and instead relies on stereotyped connectors, such as ontoRef_ (Figure 10.1).

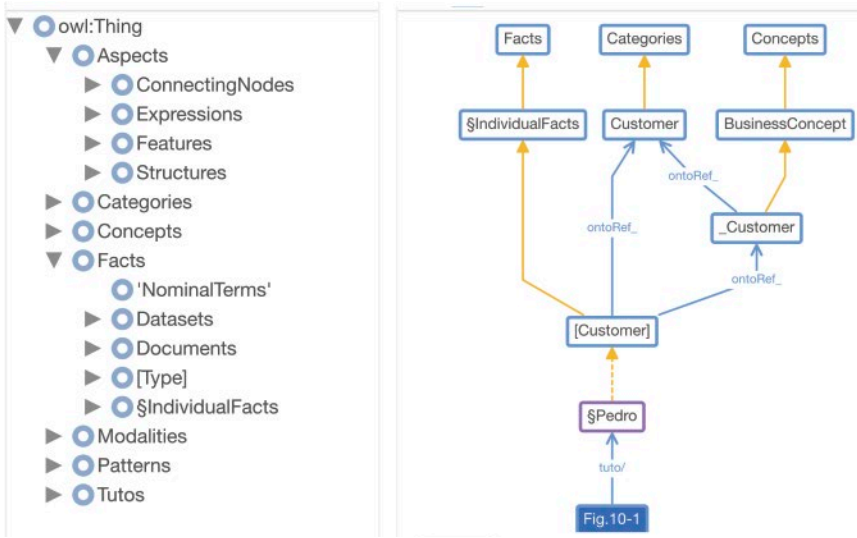


Figure 10.1 OWL/Protégé KEOPS kernel.

Native OWL/Protégé connectors appear as unlabeled. Solid lines represent type hierarchies, while dotted lines indicate instantiation.

10.1.1.2 Protégé Properties

OWL Classes can be characterized by object, data, and annotation properties.

10.1.1.2.1 Object Properties

Object properties are references that define edges between nodes. The KEOPS kernel introduces a limited number of stereotyped connectors that are free of any domain-specific semantics.

10.1.1.2.2 Data Properties and Datatypes

Data properties are attributes that characterize the nodes themselves, independent of their edges. Datatypes are built-in data property standards in OWL/Protégé. The KEOPS kernel does not interfere with Protégé data properties and datatypes.

10.1.1.2.3 Annotation Properties

Contrary to object and data properties, which pertain to content, annotation properties relate to their processing. The kernel does not rely on annotation properties.

10.1.1.2.4 Individuals

The KEOPS kernel uses Protégé individuals as defined but introduces the '§' prefix to their names to compensate for the absence of Protégé colors in the printed book.

10.1.2 KEOPS Kernel Extensions

10.1.2.1 Kernel Nodes

The KEOPS kernel categorizes OWL classes into two groups: primary (or standalone) nodes and aspect (or dependent) nodes. Primary nodes are used for OWL classes, which can be identified independently as facts, concepts, or categories. The kernel employs different notations for facts (e.g., [Staff]), concepts (e.g., _Person), categories and aspects (e.g., Staff, Person), and individuals (e.g., §Bob). OWL instantiation connectors appear as unlabeled dotted lines (Figure 10.2).

The kernel aspects are classes whose instances' identification and semantics are defined through their association with primary nodes. Aspects are used to represent structured object properties as well as connecting edges.

10.1.2.2 Kernel Ontological Connectors

The KEOPS kernel introduces three types of ontological connectors and four types of ad hoc connectors.

Ontological connectors distinguish endogenous, exogenous, and instance connectors. The kernel endogenous connectors are set between nodes belonging to the same realm (homogeneous representation semantics), whether at the same level of representation (ontoRel_) or at different levels (ontoSub_ and ontoSup_). Exogenous connectors (ontoRef_) address nodes regardless of realms (heterogeneous representation semantics) and consequently of representation level.

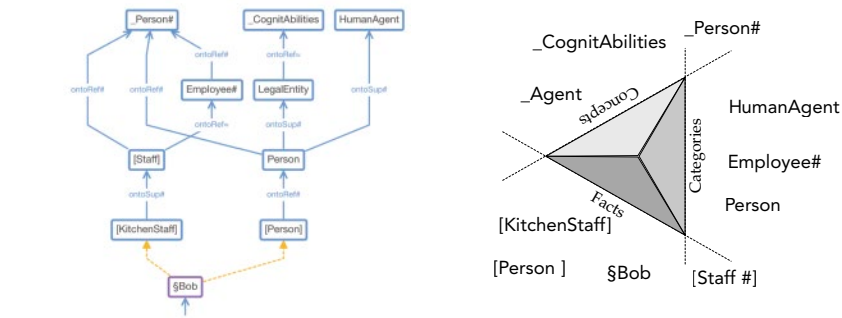


Figure 10.2 KEOPS kernel and ontological realms.

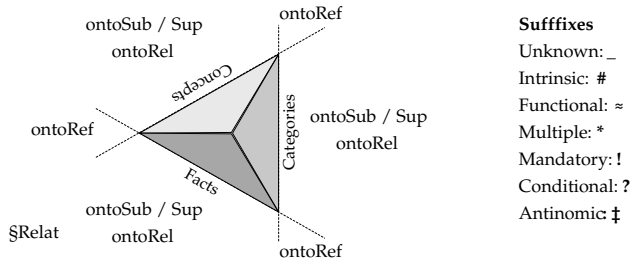


Figure 10.3 KEOPS kernel's connectors and suffixes.

Ad hoc or nonontological connectors address individual joints (§Relat_), inferences (logicRel_), semantics (semRel_), and engineering (engiRel_) relations.

Ontological suffixes '#', '≈', and '_' can be added to these connectors to denote intrinsic, functional, or indeterminate characteristics, respectively. Additional suffixes can be used to indicate mandatory ('!'), conditional ('?'), antinomic ('‡'), and multiple ('*') connections (Figure 10.3).

For instance, the observed [Dish] type has an intrinsic endogenous relation with a [Recipe] type, a functional reference with a CookingAppliance category, and a multiple endogenous relation with an [Ingredient] type. The [Ingredient] type has two exogenous references, one intrinsic with its homonym category and one mandatory (and thus implicitly functional) to the Allergy category. Moreover, the Ingredient category may have an endogenous relation to the FoodRegulation category, and an exogenous reference or through the _CarbonFootprint concept (Figure 10.4).

10.1.2.3 Aspects, Features, Properties

Ontologies are to models what movies are to pictures; they rely on continuous evolution rather than intermittent maintenance. This inherent changeability has direct consequences for features as well as the terms employed, typically properties or attributes. What starts as an observed attribute (e.g., a price) can later evolve into a composite one (e.g., a price with currency), a labeled edge (pointing to market price), a connecting node (linking product, contract, and currency), and finally, a node on its own.

While models and, to some degree, siloed ontologies require one and only one representation, heterogeneous ontologies must allow for differences between domains and traceable changes over time. With OWL/Protégé, features can be fully documented through data properties, addressing business-specific content as well as integrity constraints. However, in line with its ecumenical ambition and to avoid redundancies or inconsistencies with external developments, the

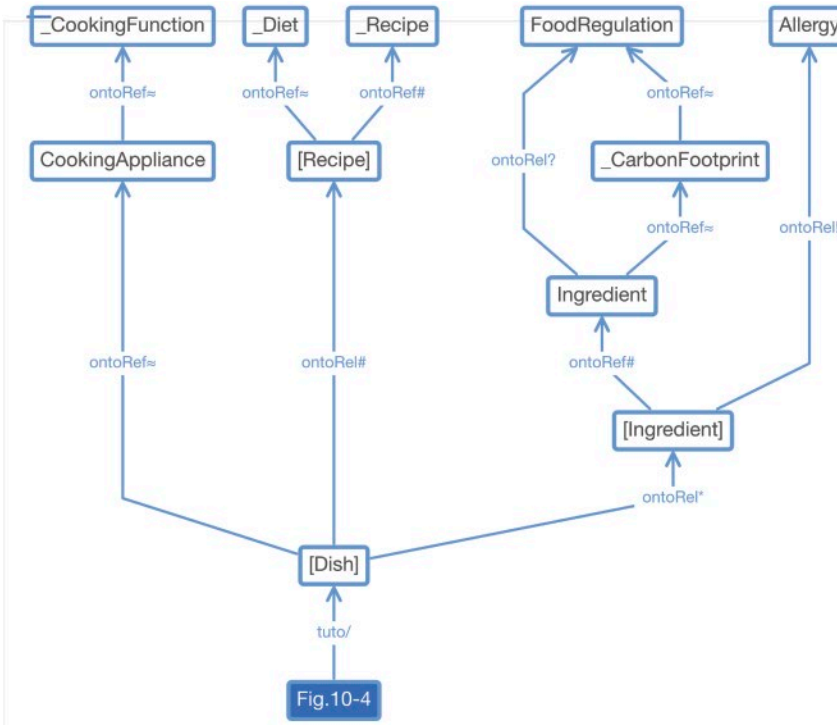


Figure 10.4 KEOPS kernel's nodes and edges.

KEOPS kernel makes no direct use of data properties except for tutoring examples. Instead, data properties are represented as elementary aspects prefixed with '+'. In the example, a data property + seatedOrder is represented by a mandatory (ontoRel!) order aspect used to distinguish between seated and away orders (Figure 10.5).

It must be remembered that aspects in ontological prisms are syntactical constructs that acquire their semantics once attached to nodes, making them semantically neutral. This allows for their uniform application to types (facts), concepts, or categories.

10.1.2.4 Semantics of Ontological Connectors

The aim of KEOPS kernel connectors is to ensure the ecumenism of knowledge architectures regardless of domain semantics. The ontoSub_ and ontoSup_ connectors are used between nodes set at different levels of homogeneous representation. Their semantics – parthood for facts and concepts, and inheritance for categories – are implicitly defined according to the realm of the targeted

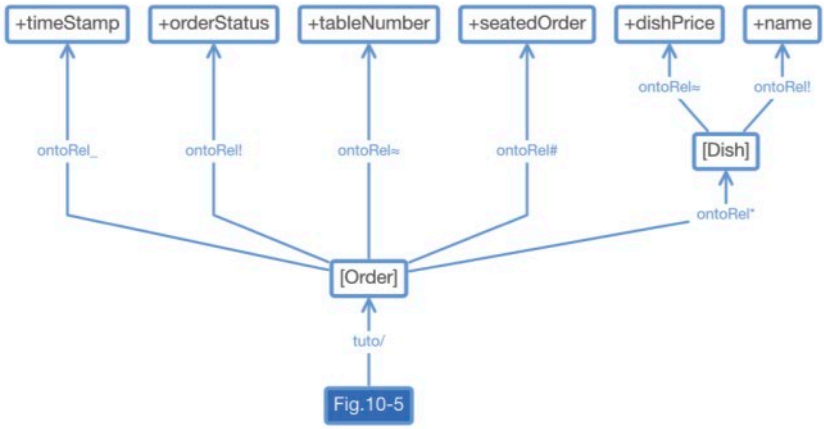


Figure 10.5 Properties.

nodes. While the kernel does not enforce consistency, it can be directly observed by comparing the formats of source and target names.

The ontoRel_ connectors are used between nodes that are set at the same levels of homogeneous representation. As with hierarchies, their semantics are implicitly determined by the realm of the targeted nodes, and their consistency can be directly verified. In contrast, the ontoRef_ connectors can be used regardless of representation semantics, allowing them to be used between nodes that are set across heterogeneous representations of facts, concepts, or categories. Given their semantic neutrality, they can be used as a default option pending further clarification.

For instance (Figure 10.6), [Dish] uses a functional reference to the Cooking-Appliance category, and an intrinsic relation to [Recipe]. Using connectors set across representation levels is more challenging, as illustrated by those between [Dish] and [Cooked Dish], which combine an intrinsic upward (ontoSup#) connector with an indeterminate downward (ontoSub_) counterpart: from a bottom-up perspective cooked dishes may be viewed as an exclusive subset, while a top-down approach would blur the lines when considering cold dishes or pastry.

That kind of indeterminacy, which in our example also appears with _CookingFunction and CookingAppliance, is an integral part of heterogeneous ontologies due to their unbounded semantics and inherent changeability. These issues can be overlooked in the synchronic representation of siloed ontologies, but they become critical for the interoperability and diachronic representation of heterogeneous ontologies. They can be managed through kernel-structured connectors (see below).

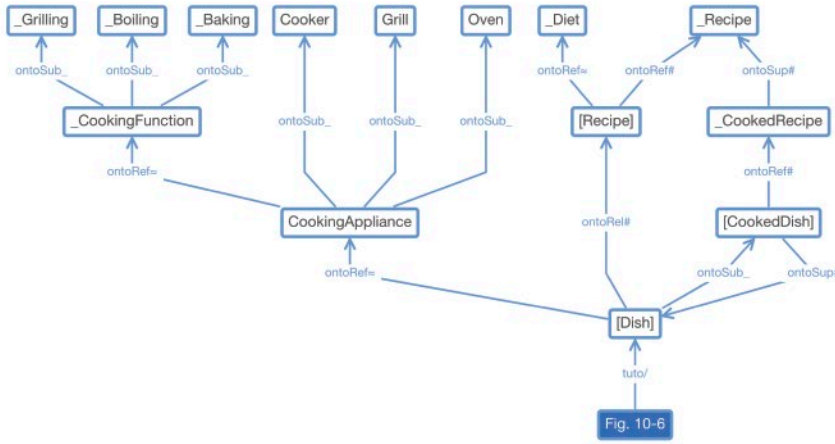


Figure 10.6 Hierarchies semantics.

The §Relat connector pertains to individuals, and since differences in representation semantics are irrelevant for individuals, §Relat connectors can be applied uniformly. For instance, assuming that orders are first written on paper, the ontologist will use the §Relat to represent the concrete joint between table (e.g., §Table4), waiters (e.g., §Waiter2), dishes (e.g., §Steak T4), and possibly nominal mentions (e.g., 'Medium rare') (Figure 10.7).

This example illustrates a use of the distinction between ontological and individual connectors: at the ontological level, the relation between seated orders and tables is mandatory but not intrinsic because customers can change tables during meals, but the relation becomes intrinsic when orders are recorded.

10.2 Representing Territories

The KEOPS kernel can be further detailed through the ways it supports the proposed playbook, starting with the mapping of territories. It is important to remember that the aim is not to develop reference models but rather to illustrate the benefits of an ecumenical approach to representations.

10.2.1 Symbolic and Physical Environments

Ontological prisms adopt a pragmatic approach grounded in both facts and artifacts, encompassing physical as well as symbolic dimensions.

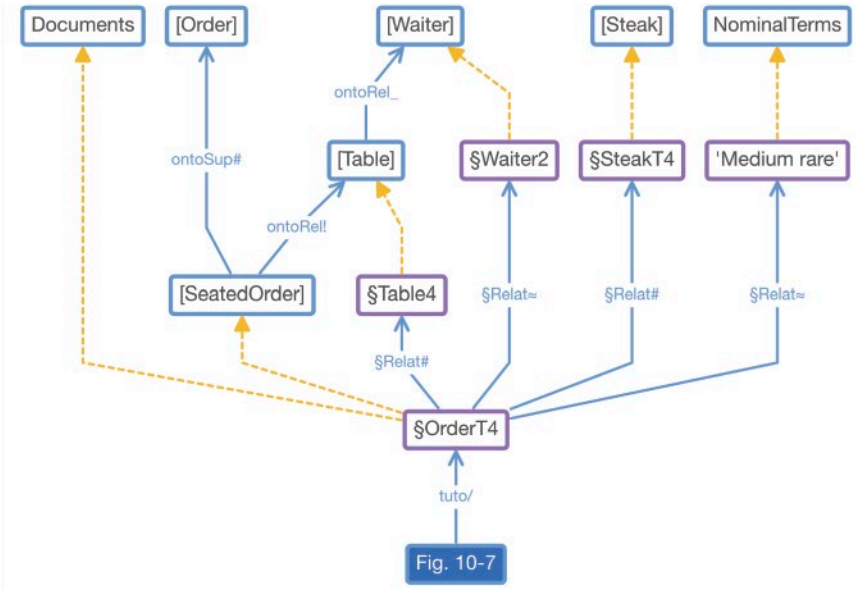


Figure 10.7 Individual connectors.

10.2.1.1 Datasets and Documents

Along with individuals or typed individuals, facts can be represented as [Document] and [Dataset], introduced as predefined types. Datasets are collections of uniformly structured elements, such as orders or customers, while documents are multimodal collections of variously structured elements that may (e.g., recipes) or may not be already typed. Unstructured data is represented by [DataObject], and packaged data by [DataProduct] (Figure 10.8).

In our example, §Orders:April appears as an instance of the [Orders] dataset, which is a collection of [Order] instances. Likewise, §Recipes:April appears as an instance of a [Recipes] document, which is a collection of poorly structured [Recipe] instances. Datasets and documents can also serve as gateways to categories and concepts, respectively, with the former accessed through taxonomies and the latter through thesauri.

10.2.1.2 Partitions, Taxonomies, Powertypes

Ontologies begin by making distinctions. Beyond differences in terminology, we can identify three terms: partitions operate first on direct observations and datasets; taxonomies follow, using partitions to build labeled hierarchies; and finally, powertypes translate taxonomies into categories. For instance, “hot” and “cold” dishes may be relevant as a partition but are of little use as a

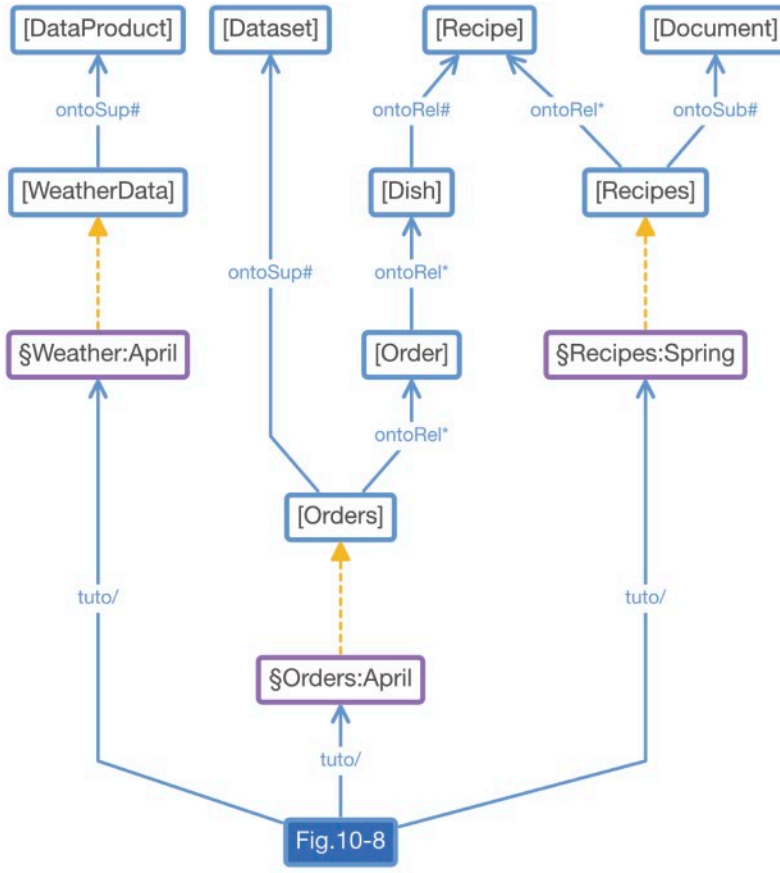


Figure 10.8 Individuals, datasets, documents.

taxonomy. In contrast, partitioning diets into culinary and religious categories could lead to a useful taxonomy, but not necessarily to effective powertypes (Figure 10.9).

Powertypes are types whose instances represent collections of instances of other types that may or may not qualify as categories on their own, i.e., categories whose instances are meant to be managed as surrogates. This flexibility is especially useful for heterogeneous ontologies, which, in contrast to models, are intended to vary over time and across domains. Due to their semantic neutrality, powertypes can be used interchangeably across ontological realms, ensuring the continuity of representations from flat partitions to hierarchical taxonomies and inheritance hierarchies. This can be particularly useful for connecting artifacts, concepts, and categories.

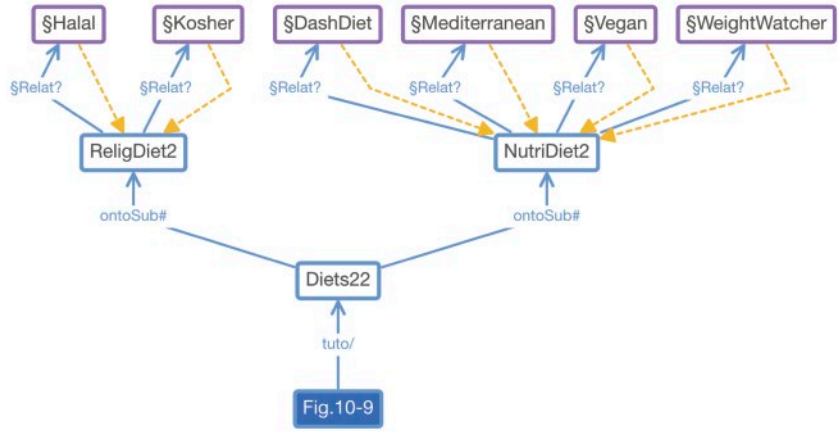


Figure 10.9 Partitions and taxonomies.

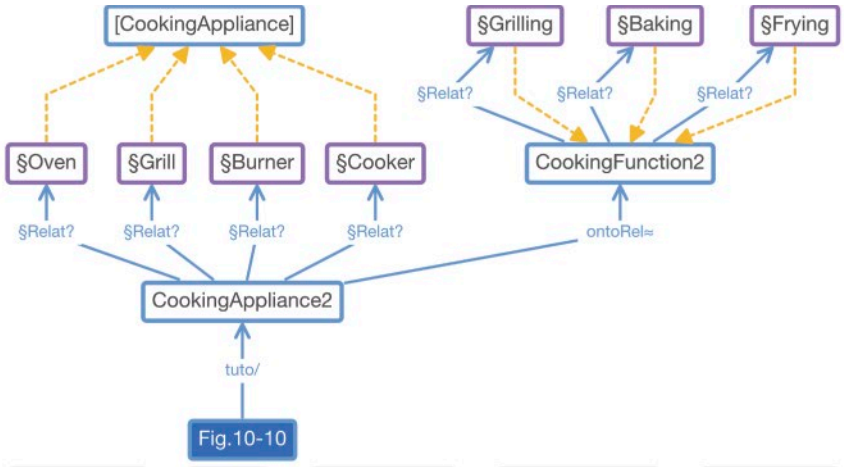


Figure 10.10 Partitions and powertypes.

For instance, a CookingAppliance2 partition can be introduced to target observed cooking appliances and then compared to a CookingFunction2 powertype defined at the category level (Figure 10.10).

10.2.2 Representation of Variants

Representing variants is at the core of ontologies, especially for heterogeneous ones. While partitions, taxonomies, and powertypes can serve as preliminary

steps, they cannot scale with complexity and will eventually need to be translated into structured representations.

10.2.2.1 Structured Connectors

The KEOPS kernel resets the semantics of established loop, merge, and split constructs in terms of ontological intrinsic and functional connectors: ontoSub and ontoSup between levels of representation, ontoRel within levels of representation, and ontoRef across realms' representation semantics. Inbound connectors are those set from parent to structures, while outbound connectors are those from structures to elements.

10.2.2.1.1 Inbound Connectors

OntoSub inbound connectors are used when parents and elements can be identified uniformly: intrinsic ones (ontoSub#) when elements can be fully identified as such (e.g., seated and away orders); and functional ones (ontoSub≈) when elements cannot be exclusively identified as such (e.g., kitchen and diner staff) (Figure 10.11).

OntoRel inbound connectors are used when parents and elements are of different types: intrinsic ones (ontoRel#) are used when identities of parents and elements are intrinsically bound (e.g., Recipe and Dish); functional ones (ontoRel≈) when they are not intrinsically bound (e.g., Party as Person or Organization).

10.2.2.1.2 Outbound Connectors

Outbound connectors replicate the nature of their corresponding inbound connectors (ontoSub or ontoRel), but intrinsic or functional qualifications are

	Same types *ontoSub*	**Different types** *OntoRel*
Intrinsic # (Bound identity)	Order # *OntoSub #* Seated/Away	Recipe # *OntoRel #* ●----->
Functional ≈ (Independent Id)	Staff # *OntoSub ≈* Kitchen ≈/Diner ≈	Party # *OntoRel ≈* ○----->

Figure 10.11 Structures inbound connectors.

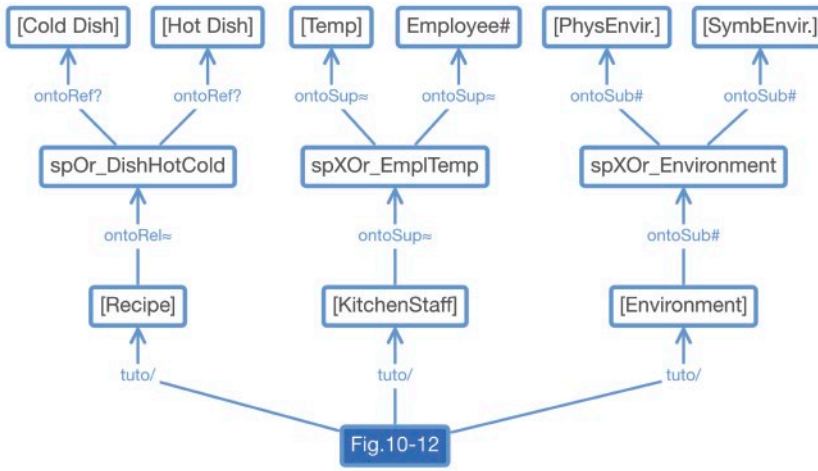


Figure 10.12 Structures.

determined by context. For example, outbound connectors to persons or organizations are functional because identities are not bound. Instances of [Party] are intrinsically connected to instances of agents (ontoRel#), while for agents, the connection is functional (ontoRel≈).

These options can be illustrated by three examples (Figure 10.12):

- When the spOr_DishHotCold construct is used by [Recipe], it's through a functional relation: same realm, different classes.
- When the spXOr-Empl_Temp construct is used by [KitchenStaff], it's through a functional subsumption: same realm, different levels.
- When the spXOr-Environment construct is used by [Environment], it's through an intrinsic subsumption: same realm, different levels, exclusive variant.

The [KitchenStaff] example illustrates the discretionary nature of representations: the subsumption is represented upward (ontoSup) from an observer perspective (facts), but it could also be represented downward (ontoSub) from a business analyst (concepts) or database manager (categories). Hence, the benefits of combining alternative representations, in particular, arise for heterogeneous and/or diachronic knowledge representations.

10.2.2.2 Combining Structures

Variants can be set across the same targets as illustrated by dishes with an intrinsic spXOr_ for the exclusive partition between cooked and raw dishes, and a functional spOr_ for the nonexclusive partition between hot and cold dishes.

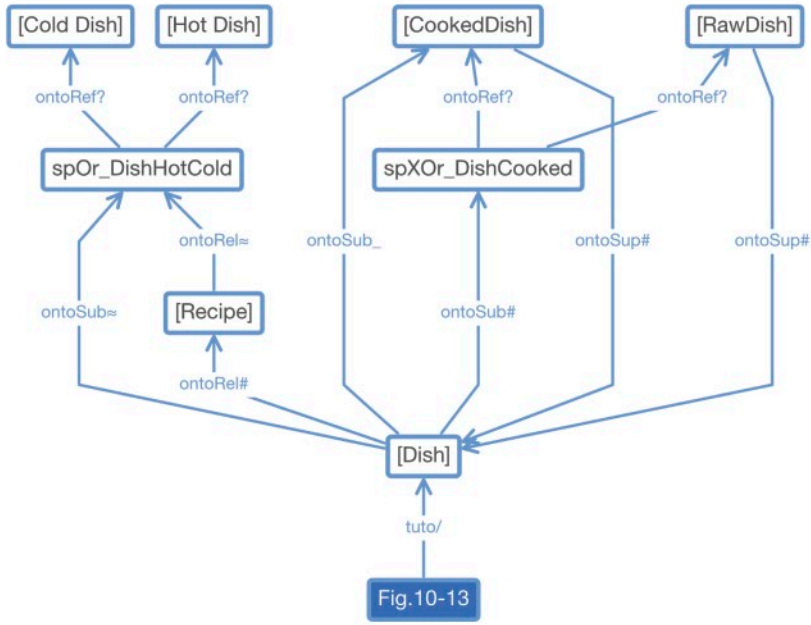


Figure 10.13 Combining structures.

The latter could also be used to characterize functional relationships (ontoRel≈) from recipes (Figure 10.13).

Alternative representations of variants can be justified by domain specificities and/or by the knowledge life cycles across heterogeneous representation semantics. However, as emphasized in the previous chapters, the parthood semantics used for representing facts and concepts cannot be aligned with the inheritance semantics used for representing categories. Leveraging their semantic neutrality, KEOPS kernel constructs can be used to connect variants across heterogeneous representations. For example, starting with [KitchenStaff], one can navigate through category variants related to Employee# and LegalEntity, and further explore concepts while checking the consistency of ontological connectors throughout (Figure 10.14).

10.2.3 Constraints, Expressions, Rules

Constraints, expressions, and rules can be defined using a Simple Knowledge Organization System (SKOS) and Shapes Constraint Language (SHACL) and implemented through interfaces like the Common Logic Interchange Format (CLIF) or plug-in OWL reasoners such as Pellet or HermiT. While SKOS

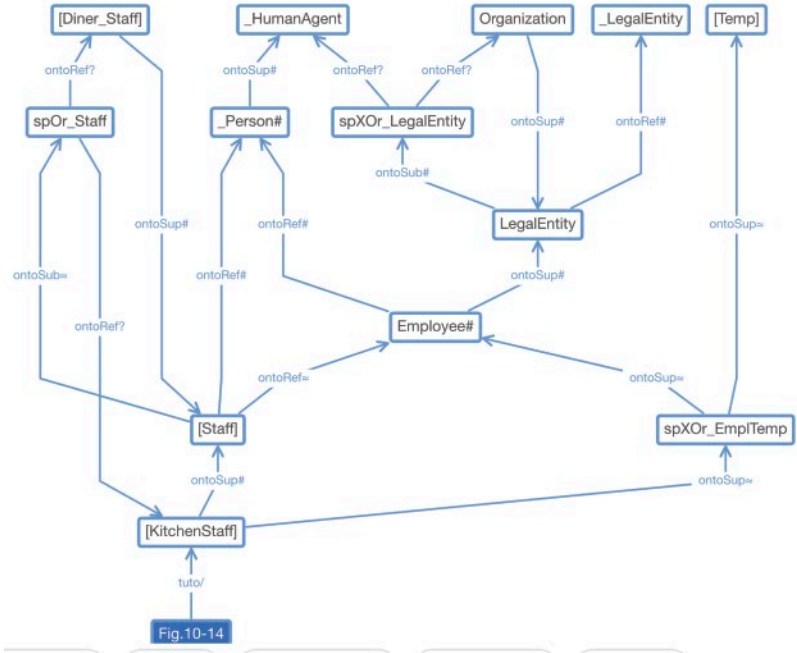


Figure 10.14 Crossing realms with structured connectors.

specifications can be integrated with OWL, the possibilities are limited, and their maintenance can be cumbersome. SHACL serves more as an alternative than as a complement to OWL, and the effectiveness of reasoners primarily depends on the consistency of the constraints and rules, as well as the consistency of tool integration. To ensure its ecumenism, the primary objective of the KEOPS Kernel is to allow alignment with predicate logic.

10.2.3.1 Constraints Through Connectors

With the KEOPS kernel, constraints can be expressed through ontological connectors, expressions, or both. Connectors allow for a direct representation of constraints in terms of intrinsic (#), mandatory (!), antinomic (‡), and optional (?) connectors. For instance, an antinomic connector (ontoRel‡) can prevent AI agents from holding organizational roles, while a mandatory connector (ontoRel!) can force judgments to be assigned to legal entities. This includes judgment roles and cognitive ability requirements (Figure 10.15).

Along with connectors, structures can also be used to implement constraints. For instance, in the case of Party, one could use a mandatory connector to LegalEntity or a spXO_ construct pointing to Person and Organization.

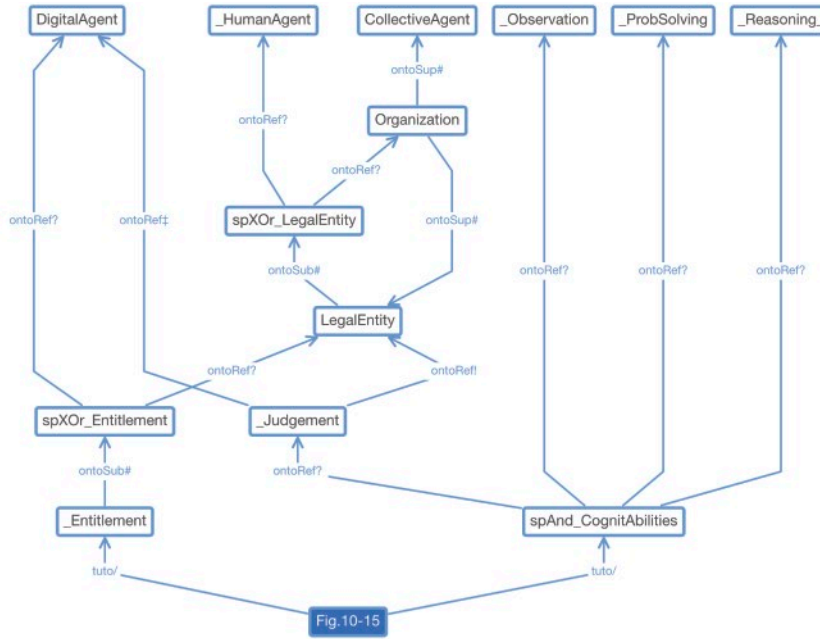


Figure 10.15 Constraints through connectors.

10.2.3.2 Nominal Footprints and Semantic Layers

Semantic layers can be defined in relation to the consistency of domain-specific thesauri and/or in terms of the consistency between thesauri and taxonomies. The former approach remains focused on the alignment of domain semantics, while the latter considers the mapping of the respective semantic spaces of thesauri, pertaining to words, and taxonomies, pertaining to things. Such a mapping can take advantage of the distinction between the nominal, formal, empirical, and ontological bases of semantic layers introduced in Chapter 6.

For example, consider a business (deontic) rule such as "The head waiter must have excellent English." This rule could be represented as a nominal footprint combined with a §HumanResources SemanticSpace, where semantic spaces serve as the thesaurus counterpart of business domains. Based on thesauri, §SemRel_ connectors could be used to organize nominal items into semantic graphs, which can then be further associated with relevant entities through ontological connectors (Figure 10.16).

Alternatively, constraints and rules can also be represented by formal expressions.

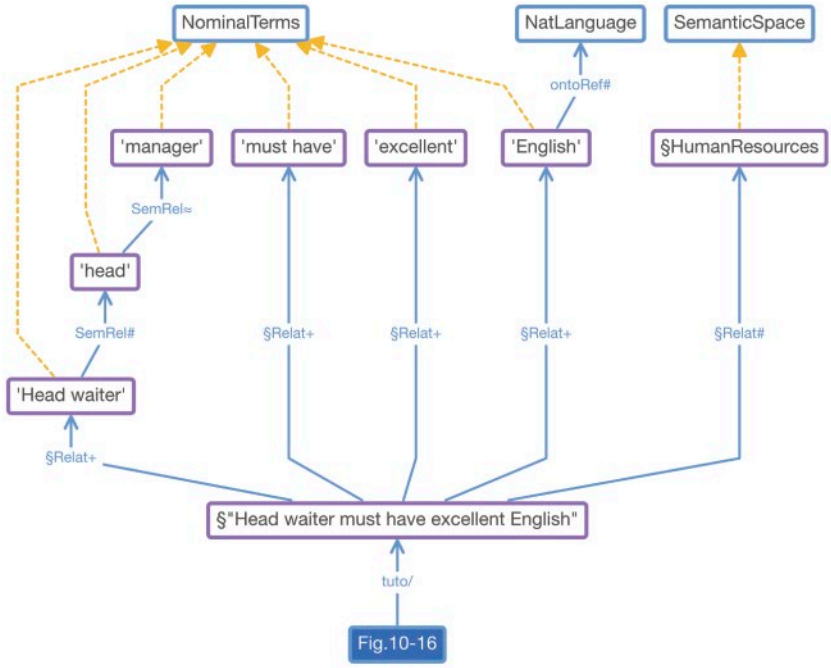


Figure 10.16 Nominal footprints and semantic spaces.

10.2.3.3 Constraints Through Expressions

Expressions, commonly referred to as methods in modeling terminology, are represented as aspects intended to be instantiated through their association with nodes. The KEOPS kernel distinguishes between numerical, logical, and business-specific expressions. Numerical expressions are supposed to be supported by dedicated tools, while logical expressions are intended to align with the CLIF and predicate logic. Regarding business expressions, the objective is to enable interoperability between ontological prisms and relational databases, with expressions mirroring database schemas.

Complex constraints can thus be represented by combining expressions and ontological connectors. In our example, the rule "Head waiter must have excellent English" is associated with a formal expression (languageLevel) intended to be aligned with a relational scheme or a Prolog predicate, using a prefix '§' for controlled arguments. In that example, the controlled arguments allow for a joint with the partition of staff according to their language level (LangLevel2) (Figure 10.17).

The constraint can be verified for the controlled arguments §Staff and §Lang.

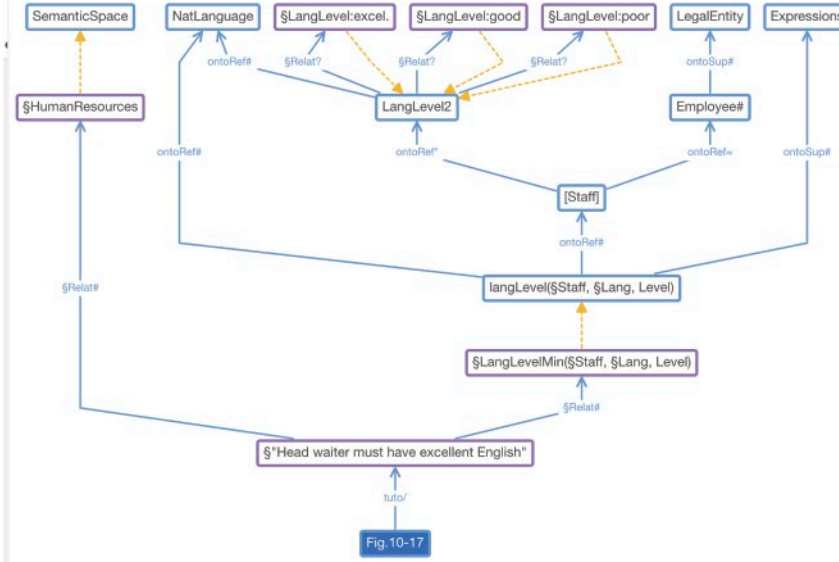


Figure 10.17 Constraints through expressions.

10.2.3.4 Inference Rules

The KEOPS kernel uses five dummy inference connectors to illustrate dependencies of truth values: antinomy (logic‡), necessity (logicΞ), congruence (logic>=), and divergence (logic>≠). The constraint example above can thus be expressed using the logicΞ connectors between the head waiter position and the English language condition. The inference connectors can also be applied to positive (logic>=) and negative (logic>≠) numerical changes. For instance, dish prices and orders are expected to move in opposite directions, while ice cream orders are expected to follow changes in weather temperature (Figure 10.18).

While these are makeshift options intended to be replaced by established reasoners, a more generic approach can rely on the KEOPS kernel's ontological modalities (see 10.4.1): deontic (ΔPermissible, ΔImpermissible, ΔObligatory, ΔOptional) or empirical (ΔPossible, ΔImpossible, ΔNecessary, ΔContingent).

10.3 Representing Intents

This segment is focused on the representation of agents, goals, and projects. Variants are detailed separately at the end of the segment.

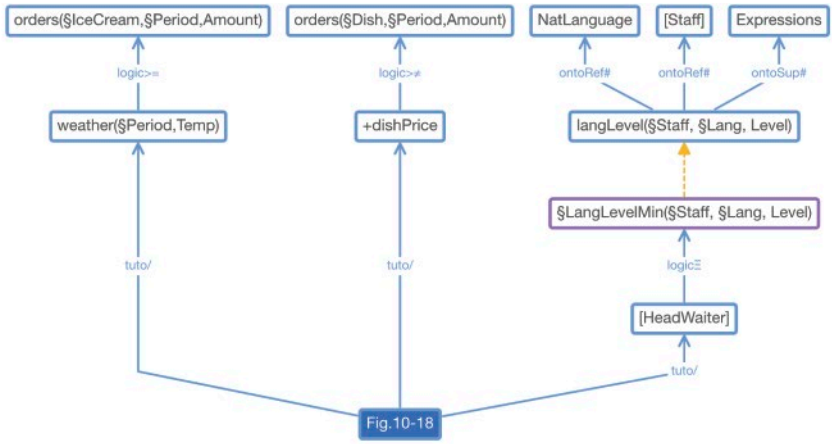


Figure 10.18 Inference connectors.

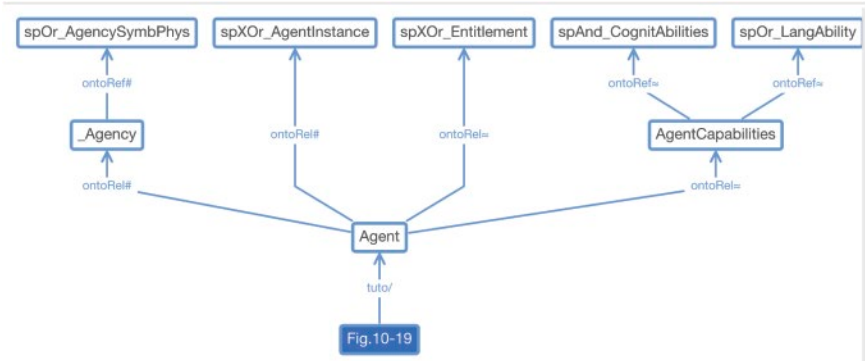


Figure 10.19 Agents: embodiments, entitlements, capabilities.

10.3.1 Agents and Goals

The development of agentic AI architectures and the role of ontologies supporting them shed light on the representation of actors, goals, plans, and decision-making.

10.3.1.1 Agents, Actors, Roles

The representation of agents should address embodiments, abilities, and entitlements. As already mentioned, agents can be embodied as mechanical, digital, biological, or collective entities. Regarding entitlements, the representation should distinguish between legal and digital entities (Figure 10.19).

Taking Agentic AI into consideration, agents' cognitive and communication capabilities should also be addressed, the former for observation, reasoning,

judgment, and problem-solving; the latter for natural, symbolic, digital, and physical communication. Agents could also be characterized by their environment agency (symbolic or physical) and entitlements.

Agents represent active individuals, actors are agents in bounded contexts, roles describe what actors can do in organizational contexts, and actors describe how they do that in actual environments.

10.3.1.2 Goals

Goals can be defined by environments, parties involved, and outcomes. The representation below illustrates three different ways to use structures (Figure 10.20).

- [Outcome] is characterized by a functional nonexclusive partition (spOr_) targeting tangible and intangible subtypes (ontoSub≈).
- [Environment] is defined by intrinsic (ontoSub#) exclusive subtypes (spXOr_) targeting tangible and intangible subtypes (ontoSub#).
- [Horizon] directly combines nonexclusive functional operational, tactical, and strategic subtypes (ontoSub≈) referencing data, information, and knowledge, respectively.

10.3.2 Projects

10.3.2.1 Projects

Projects are, by definition, set across facts (current situation), categories (realization), and concepts (intents). These perspectives can be explicitly represented using the KEOPS kernel (Figure 10.21).

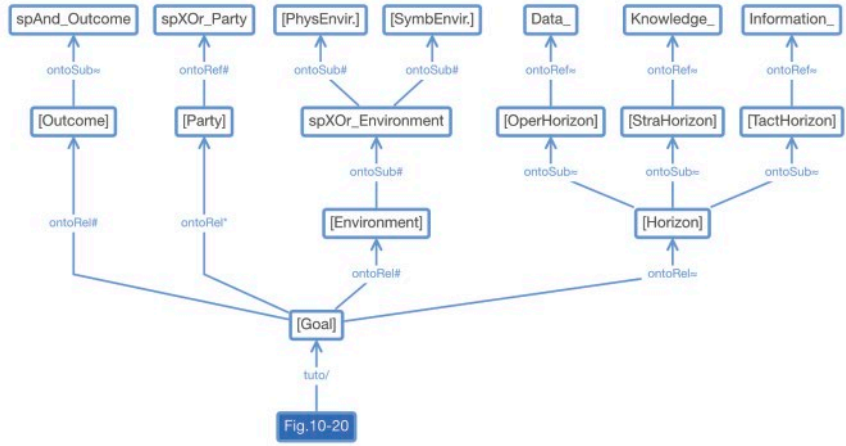


Figure 10.20 Goals.

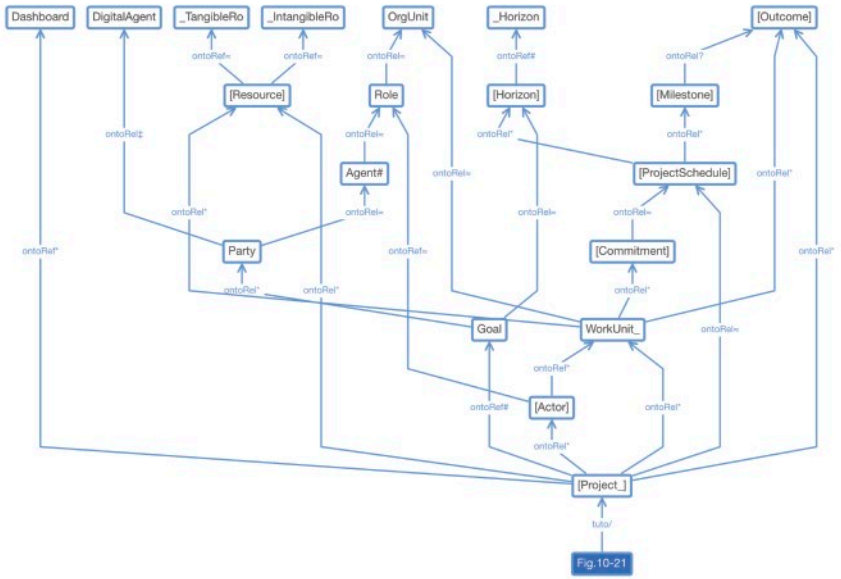


Figure 10.21 Projects.

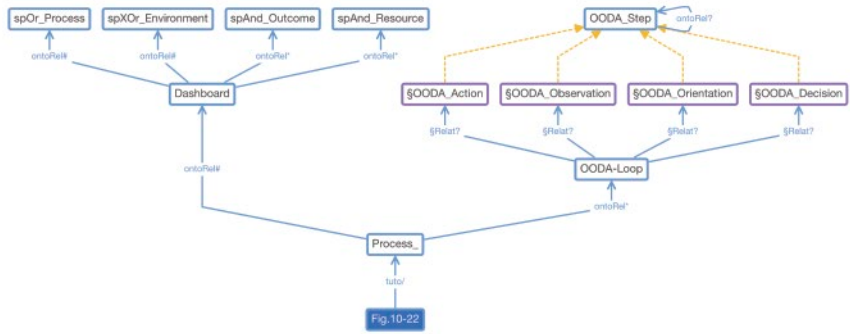


Figure 10.22 Process.

10.3.2.2 Process and Work Units

Taking the Observation–Orientation–Decision–Action (OODA) loop as a reference for the decision-making process, a project would iterate through four kinds of work units (OODA_Step), with advances monitored in terms of state of processes, environment, outcome, and resources (Figure 10.22).

10.3.3 A Sample of Variants

So far, the representation of variants and levels of abstraction has been based on an ontological extension of established constructs for composition, alternatives, and subtypes. Before presenting how abstraction levels can be represented with the KEOPS kernel, we will briefly revisit some of the examples mentioned above.

Agents are first characterized by the nature of their instantiation and the scope of their agency. The former is exclusive; the option for the latter is discretionary (Figure 10.23).

Agentic AI concerns are intended to be addressed through cognitive abilities and entitlements. Cognitive abilities distinguish between observation, reasoning, problem-solving, and judgment. Entitlements are defined by the nature of agents and their roles (Figure 10.24).

Primary variants for roles pertain to the respective entitlements of digital agents and legal entities, individuals, or organizations. Conversely, roles can also be characterized by their scope: organizational, functional, or operational. The former is exclusive, the latter is not.

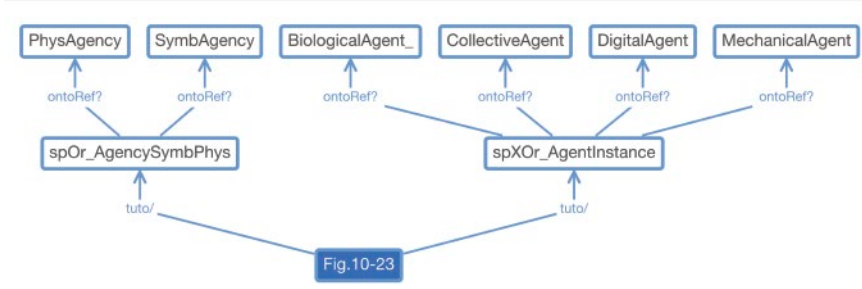


Figure 10.23 Agents and agency.

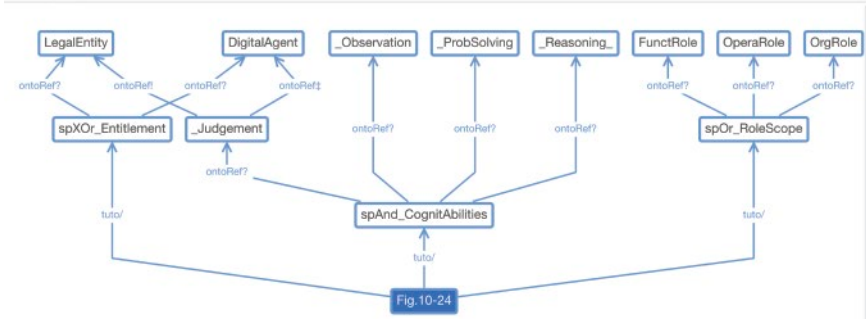


Figure 10.24 Agent abilities.

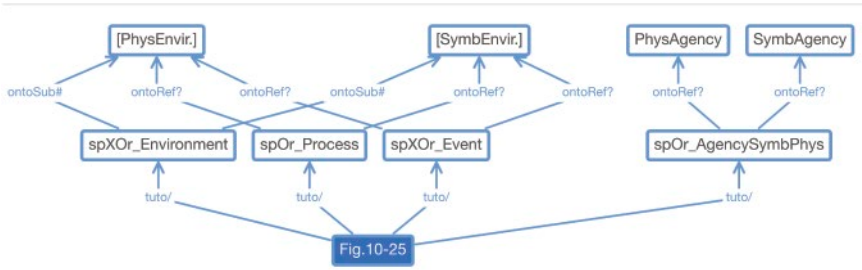


Figure 10.25 Agency, process, environment.

Finally, roles and agency must be congruent with processes' environments (Figure 10.25).

Time is addressed through events.

As illustrated by this sample, an open-ended range of objects and phenomena is bound to be characterized by a limited number of intrinsic variants, making the case for principled abstraction levels.

10.4 Representing Abstractions

Representing abstractions is arguably the defining dilemma of modeling, especially when addressing heterogeneous ontologies. As explained in our book, the core of the matter is akin to a sleight of hand: one hand is meant to separate domain-specific concerns from ecumenical constructs, while the other hand tries to ensure their seamless integration. Customary solutions such as metamodels and upper ontologies fall short of establishing reliable and effective mappings. The former struggles due to the overlaps between syntax and semantics, while the latter is hindered by the cumbersome conceptual twists required to align upper and lower ontologies. Ontological modalities borrow from both approaches: stereotypes and patterns from the former, and intrinsic and functional qualities from the latter.

10.4.1 Modalities

Modalities (explained in Chapter 9) can be viewed as ontological stereotypes that provide a declarative alternative to imperative meta-models. The kernel includes KEOPS' seven modalities for temporality, instantiation, identification, agency,

communication, structure, and veracity. It must be noted that the list can be extended and the options modified, provided they do not overlap.

10.4.1.1 Ontological Modalities

Ontological modalities address intrinsic and functional qualities regardless of realms. As such, they ensure the consistency of abstraction levels without forcing discretionary and unwieldy options. In our example, a _LegalEntity has an elementary structure modality while Organization has a collective structure one, with an intrinsic lineage between them. Regardless of business-related explanations of the discrepancy, it can be explained by the different abstraction semantics applied to concepts and categories, which is materialized by the use of an ontoRef# connector (Figure 10.26).

10.4.1.2 Grammatical Modalities

Modalities can also be applied to the parsing of texts (Figure 10.27).

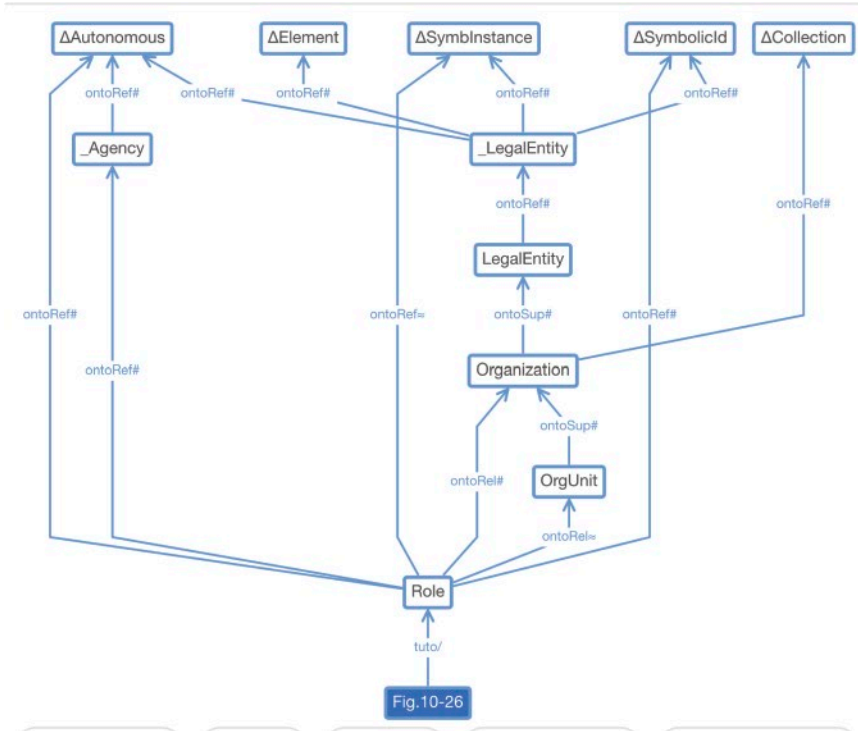


Figure 10.26 Ontological modalities.

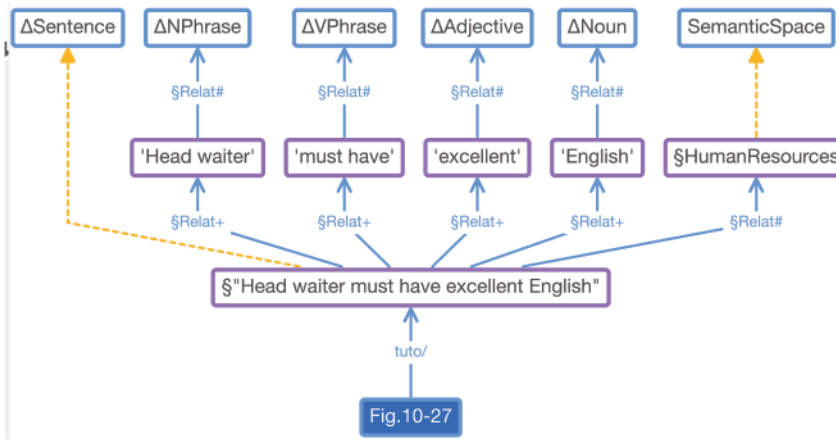


Figure 10.27 Grammatical modalities.

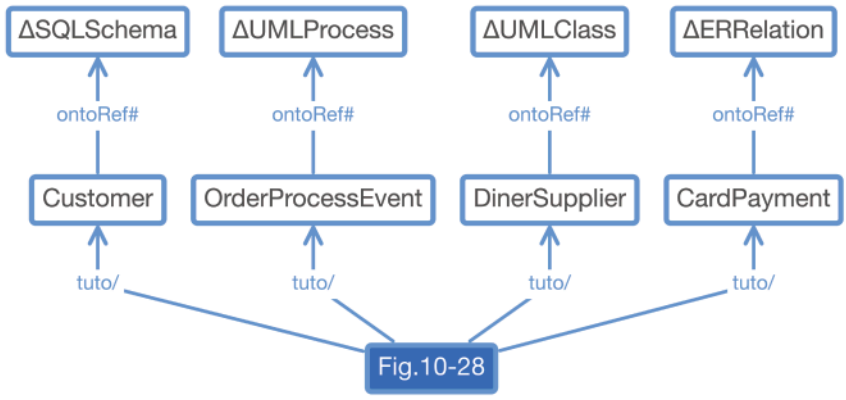


Figure 10.28 Representation modalities.

10.4.1.3 Representation Modalities

Finally, modalities can be used to ensure the interoperability of representations between ontologies and tools such as relational databases or Unified Model Language models (Figure 10.28).

10.4.2 Patterns

Patterns (prefix Δ2) are shared representations designed for reuse across different domains and realms. They are built on modalities, allowing for a shortcut free of the semantic discrepancies between representation levels. Patterns are rooted in realms and intended to be applied across realms; they can be defined directly from well-proven entities and modalities or by specializations of existing patterns. Additionally, they can combine different patterns.

10.4.2.1 Blueprints (Concepts)

Blueprints are conceptual patterns. In our example, a Δ2PhysRole blueprint can be introduced as a specialization of Δ2Role because the instantiation modality of the latter is defined as functional and can thus be reset to intrinsic. The reverse would not be allowed (Figure 10.29).

Blueprints can be composite or be mapped to profiles.

10.4.2.2 Profiles (Domains)

Profiles are domain patterns; they can be defined directly or based on blueprints, for instance, a Δ2CookingControl profile may reuse Δ2PhysRole and D2Equipment blueprints (Figure 10.30).

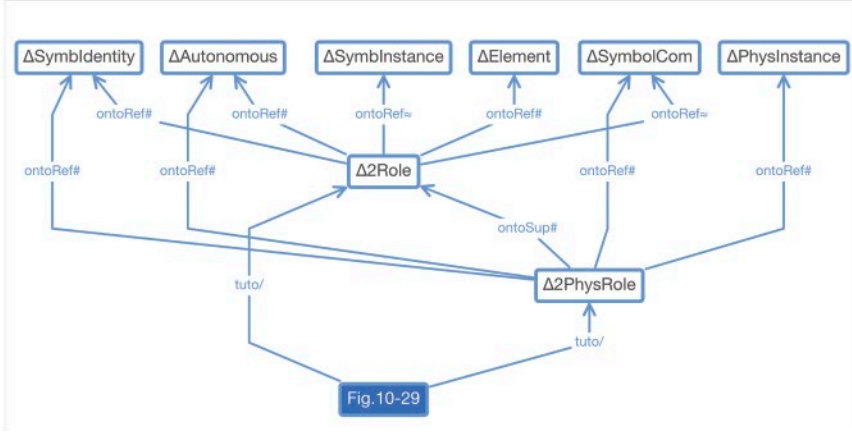


Figure 10.29 Blueprint.

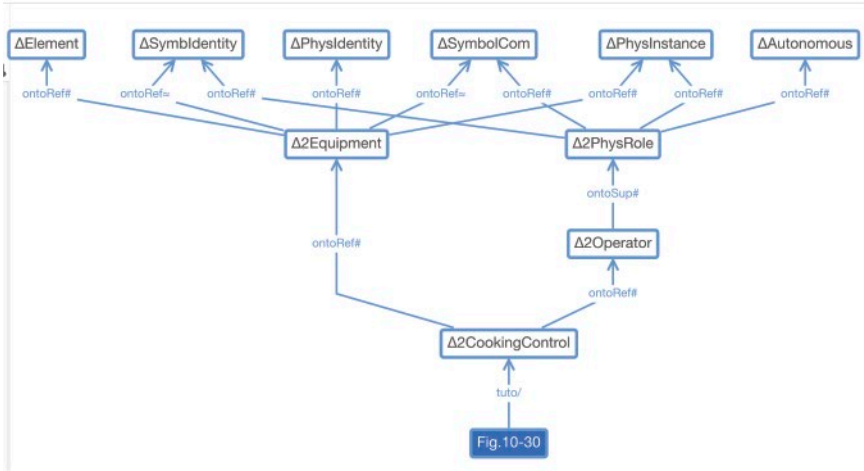


Figure 10.30 Profiles.

10.4.2.3 Templates (Models)

Templates are realization patterns. As such, they are best defined in relation to representation modalities.

Key Points

As noted before, this chapter is not supposed to serve as an OWL manual or to present reference models. Instead, to serve two purposes: first, to establish, through the kernel, the feasibility, practicality, and consistency of the KEOPS framework; then, to demonstrate the versatility of the OWL language, well beyond the common view that restricts it to web semantics.

Conclusion

This book introduces ontological prisms, a paradigm shift that unfolds the fabric of cognition, untangling taxing knots in knowledge engineering and artificial intelligence (AI). These can be considered from engineering, architectural, and cognitive perspectives.

From an engineering perspective, the primary and pragmatic objective of ontological prisms is to span the abstraction gap between foundational and domain-specific ontologies. This allows for an ecumenical and declarative integration of upper and lower ontologies without imposing unwieldy inheritance lineages. Furthermore, the epistemic distinction between facts, categories, and concepts, combined with the interoperability of their heterogeneous representation semantics, enables a diachronic management of knowledge engineering processes, regardless of overlapping life cycles. These objectives can be achieved through the Knowledge Engineering with Ontological Prisms (KEOPS) framework.

From a knowledge architectural perspective, the dual framework established by facts, categories, and concepts on one hand, and data, information, and knowledge on the other, ensures the integration of individual (agents) and collective (organizations) intelligence. Moreover, individual and collective layers can be crossed with digital and human tiers, agentic AI and systems for the former, persons and organizational units for the latter.

From a cognition perspective, establishing a principled and actionable distinction between communication (language) and representation (knowledge) ontological prisms provide a blueprint for an agentic operating system, first with regard to a collaboration layer combining conversation and context, second with regard to agents' cognitive capabilities regarding observation, reasoning, and judgment.

The seamless integration of knowledge and collaboration architectures on the one hand, agentic cognition on the other, paves the way to a holistic approach to collective learning, weaving together individuals, organizations, and systems.

Annexes

Annex 1

Summary of Notations

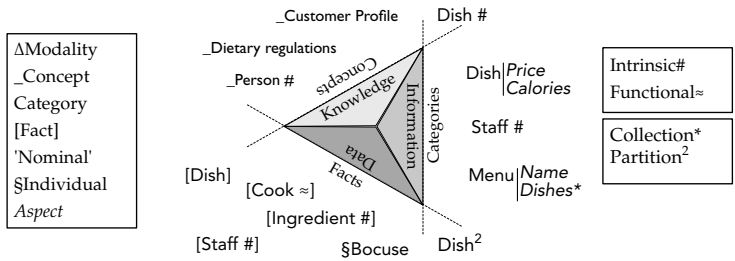


Figure A1.1 Nodes.

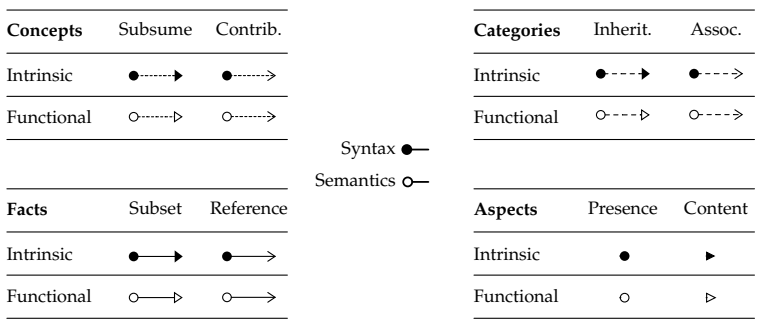


Figure A1.2 Connectors.

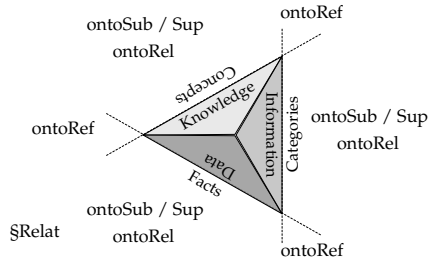


Figure A1.3 Connectors suffixes.

Annex 2

Acronyms and Glossary

The aim of this glossary is not to engage in semantic debates but to adopt Ockham's principle of parsimony and produce a definite set of concise definitions that are effective, purposeful, and consistent.

- Effective: The definitions must yield clear and unambiguous answers.
- Purposeful: The definitions must directly address the issues considered in this book.
- Consistent: The definitions should not entail circular or conflicting reasoning.

It follows that this glossary is, by design, doubly limited in both the range of terms considered and the scope of the definitions themselves.

Acronyms

- AI: Artificial Intelligence
- BFO: Basic Formal Ontology
- CCO: Common Core Ontology
- CLIF: Common Logic Interchange Format
- DAG: Directed Acyclic Graph
- DDD: Domain-driven Design
- DMS: Document Management System
- DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering
- DSL: Domain-specific language
- DSS: Decision Support System
- EA: Enterprise Architecture
- ER: Entity Relationship
- FDA: US Food and Drug Administration
- GDPR: EU General Data Protection Regulation
- GenAI: Generative Artificial Intelligence

- GenLM: Generative Language Model
- KEOPS: Knowledge Engineering with Ontological Prisms
- KG: Knowledge Graph
- LCM: Large Concept Model
- LLM: Large Language Model
- LPG: Labeled Property Graph
- M*R/K: Multilayer Metamodel for Representation and Knowledge
- MBSE: Model-based Systems Engineering
- MDA: Model-driven Architecture
- MLM: Multimodal Language Model
- NIEM: National Information Exchange Model
- OMG: Object Management Group
- OO: Object-oriented
- OODA: Observation, Orientation, Decision, Action
- ORM: Object-role Modeling
- OWL: Web Ontology Language
- RAG: Retrieval-augmented generation
- RDF: Resource Description Framework
- SHACL: Shapes Constraint Language
- SKOS: Simple Knowledge Organization System
- SQL: Structured Query Language
- UFO: Unified Foundational Ontology
- UML: Unified Modeling Language
- VSM: Viable System Model
- W3C: World Wide Web Consortium
- XML: Extensible Markup Language

Agent

Individual with agency.

Agency

Ability to change facts.

Aggregation

Set of instances that are functionally bound together by a root instance, while keeping their own separate identities.

Aspect

Type which cannot be instantiated on its own.

Category

Shared symbolic representation of facts or concepts.

Causality

Dependency between facts (extensional or empirical causality) or between concepts (intensional or deontic causality).

Class

Name of symbolic representations with OWL. Encompasses ontological prisms representation of aspects, facts, concepts, and categories.

Cognition

Ability of agents to process facts (observation), concepts (judgment) and categories (reasoning).

Collection

Set of identified individuals

Commitment

A pledge by an agent to ensure the realization of a fact.

Communication

Exchange of data between agents.

Complexity

Assuming full explainability (i.e., zero entropy), extensional complexity is the ratio between the number of explanatory variables (as defined by taxonomies) and the number of relevant facts (as referenced in thesauri). Assuming all terms can be mapped to concepts, intensional complexity is the ratio of the number of relevant facts that cannot be addressed by concepts to the total number of relevant facts.

Composition

Set of instances whose identity is bound to a root instance.

Comprehension

The holistic understanding by an agent of the different components of an explanation.

Concept

The mental representation of physical or symbolic objects or phenomena detached from actual observations.

Connector

A symbolic construct used to associate nodes in graphs.

Container

Vessel used to manage sets of unidentified individuals or uncountable content.

Context

Agent's physical or symbolic environment.

Conversation

Communication with shared semantics.

Correlation

Observed conjunction of facts regardless of possible explanations.

Data

Observations, datasets, or documents, regardless of meaning.

Dataset

Container of labeled digital content.

Decision

Commitment with regard to an explicit context and objectives.

Deduction

Assertion of hypothetical facts regardless of observations.

Deontic

Of something established by social consent.

Digital

A format

Document

Structured digital object.

Domain

A set of identified homogeneous elements (facts, concepts, or categories) governed by the same authority.

Edge

A graph construct connecting nodes.

Embedding

A self-contained component in a language model with defined structure and semantics.

Endogenous

A quality solely determined within a bounded context.

Entitlement

What agents have the right to do.

Entropy

Operational entropy is the ratio of the number of relevant facts that cannot be explained by explanatory variables (as defined by taxonomies) to the total number of relevant facts (as referenced in thesauri). Semantic entropy is the ratio of the number of terms used to label the relevant facts that cannot be aligned with concepts to the total number of terms used to label the relevant facts.

Environment

The set of objects and phenomena, physical or symbolic, that can be directly observed by agents.

Epistemic

Pertains to the nature, sources, validity, and boundaries of knowledge. Akin to meta-ontology.

Event

Changes in the state of identified objects, activities, or agents' internal representation of such states.

Experience

Learning from the use of categories to operate on facts.

Explainability

The ability to trace back the symbolic path that connects facts, reasoning, and judgments to the outcome in question.

Extensional

Pertains to the environment of the system considered.

Fact

Whatever can be directly observed: objects and phenomena, numbers, or documents.

Foundational (Ontology)

Generic (or upper-level) ontology designed to serve as a reference to domain-specific (or lower-level) ones.

Functional

Aspects (properties or behaviors) of individuals which can change without affecting their identity.

Generative

The ability of artificial intelligence to generate facts regardless of actual observations.

Grammar

Set of constructs and rules governing the agencing of communication components.

Hallucination

Nonfactual or nonsensical outcomes from generative language models.

Heterogenous

Of objects or phenomena defined, frames, and/or represented using different symbolic models.

Homogenous

Of objects or phenomena defined, frames, and/or represented using different symbolic models.

Identity Principle

Modus operandi used to uniquely distinguish individuals.

Induction

Hypothetical or second-degree deductions, as they utilize functional knowledge of facts and categories to expand functional knowledge of facts.

Inference

Knowledge obtained from deduction, induction, or abduction.

Information

What is obtained by applying structures and semantics to data.

Inheritance

Abstraction constructs of Object-oriented representation that allow aspects (properties and behaviors) of classes at higher levels to apply to classes at lower levels.

Interpretation

Relates to the intentional, which can operate across three tiers: individual agents harboring their own objectives and interpretations, cooperating agents sharing objectives and interpretations, and adversarial agents speculating about their mutually concealed objectives and interpretations.

Instantiation

Creation of individuals according to types, categories, or concepts.

Intensional

Pertains to mental representations.

Intrinsic

Aspects of individuals that cannot change without affecting their identity.

Intuition

Additional knowledge obtained without explicit symbolic processing.

Judgment

Blank commitment to specific knowledge elements made by agents, regardless of supporting arguments.

Knowledge

Intent supported by data and/or information.

Knowledge Graph

Symbolic construct made of nodes and edges, which can be used to represent an almost unlimited range of issues.

Language

Symbolic combination of grammar and semantics supporting communication between agents.

Language Model

Symbolic construct supporting language performance.

Literal

Sign without symbolic reference.

Logic

Grammar dedicated to truth-preserving processing.

Modality (Ontological)

Stereotype addressing the intrinsic and functional qualities of things.

Nominal

Of a thing whose reality is reduced to its name or identity.

Observation

Materialization of a fact. Ontological prisms consider three kinds of observations: individuals, datasets, and documents.

Ontology

As a branch of philosophy, ontology deals with the nature of the existence of things; as commonly understood, ontologies are upgraded conceptual models. In between definitions vary according to beliefs about the nature of things.

Orientation

Mapping observed facts to mental and/or collective representations.

Parthood

Also known as Mereology, Parthood is the theory of parts and relationships. Ontological prisms rely on parthood models to describe facts and concepts.

Partition

Grouping of individuals according to the shared value of selected properties.

Physical

What can be perceived through the senses.

Pragmatics

What can be understood and/or performed without relying on symbolic representations.

Predicate

Statement pertaining to individuals and symbols.

Projection

Mapping of singular mental representations into collective ones.

Property

The characteristic of an object or phenomenon.

Proposition

Statement pertaining to individuals.

Realm

Environment (facts), mental representations (concepts), collective representations (categories).

Reason

Ability to process symbolic representations.

Semantics

Association of meanings to symbols.

Semiotics

Association of signs to symbols.

Sign

A marker that can be directly and immediately associated with facts.

Symbol

A marker that can be associated with a representation regardless of facts.

Symbolic

Of a representation detached from facts.

Syntax

An agreed-upon organization of symbols, independent of what they represent.

Taxonomy

Hierarchical classification of labelled facts.

Text

Structured sequence of symbols.

Thesaurus

Networked organization of meanings.

Time

Time is what happens between events.

Time Frame

Root event and sequencing scale.

Transparency

Visibility on facts

Traceability

Visibility on events

Vector Space

All points that can be defined by the values of a finite set of parameters.

Veracity

Qualified truth value of a fact.

Annex 3

Bibliography

Aggarwal, T., Salatino, A., Osborne, F., and Motta, E. (2026). *Large Language Models for Scholarly Ontology Generation: An Extensive Analysis in the Engineering Field*. Elsevier.

Allemang, D., Hendler, J., and Gandon, F. (2020). *Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL*, 3rd edn. ACM Books.

Almeida, J.P.A., Falbo, R.A., Guizzardi, G., and Sales, T.P. (2019) gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO). Available from http://purl.org/nemo/gufo#.

Augusto L.M. (2022). Categories and foundational ontology: A medieval tutorial. *Journal of Knowledge Structures and Systems* 3(1):1–56.

Arp, R., Smith, B., and Spear, A. (2015). *Building Ontologies with Basic Formal Ontology*. Cambridge: MIT press.

Barrasa, J., and Webber, J. (2023). *Building Knowledge Graphs: A Practitioner's Guide*, 1st edn. O'Reilly.

Bazerman, M.H., and Moore, D.A. (2012). *Judgment in Managerial Decision Making*, 8th edn. Wiley.

Beer, S. (1985). *Diagnosing the System for Organizations*. Wiley.

Beer, S. (1995). *Brain of the Firm*, 2nd edn. Wiley.

Bergman, M.K. (2018). *A Knowledge Representation Practionary: Guidelines Based on Charles Sanders Peirce*. Cham: Springer.

Bertolini, D., Perini, A., Susi, A., and Mouratidis, H. (2005). The Tropos visual modeling language. A MOF 1.4 compliant meta-model. In: *Contribution for the AOSE TFG meeting*, February 28. Ljubljana, Slovenia, collocated with the Second AgentLink III Technical Forum (AL3-TF2).

Borgo, S., Ferrario, R., Gangemi, A., Guarino, N., Masolo, C., Porello, D., Sanfilippo, E.M., and Vieu, L. (2023). *DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering*. IOS Press.

Braun, G., Fillottrani, P., and Keet, M. (2022). A framework for interoperability with hybrid tools. *Journal of Intelligent Information Systems* 60(2):437–462.

Charniak, E. (2024). *AI & I: An Intellectual History of Artificial Intelligence*. The MIT Press.

Clayton, A. (2022). *The Bernoulli Fallacy: Statistical Illogic and the Crisis of Modern Science*. New York: Columbia University Press.

Coase, R.H. (1990). *The Firm, the Market, and the Law*. University of Chicago Press.

Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Professional.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley.

Collège de France (2025). *Formes de L'intelligence*. Colloque de rentrée.

CUBRC (2019). An Overview of the Common Core Ontologies, National Institute of Standards and Technology. Available from: https://www.nist.gov/system/files/documents/2021/10/14/nist-ai-rfi-cubrc_inc_004.pdf

Davis, R., Shrobe, H., and Szolovitz, P. (1993). What is a knowledge representation? *AI Magazine* 14(1):17–33.

Deacon, T.W. (1997). *The Symbolic Species, The Co-evolution of Language and the Brain*. W.W. Norton.

Dehbozorgi, A. (2025). OntoLog: The Unified Intelligence Paradigm That Experts and Entrepreneurs Can't Ignore. Available from https://medium.com/@alirezadehbozorgi83/ontolog-the-unified-intelligence-paradigm-that-experts-and-entrepreneurs-cant-ignore-8695b10b6949

Dunbar, R. (2014). *Thinking Big: How the Evolution of Social Life Shaped the Human Mind*. Thames and Hudson.

Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional.

Fannader, R. (2021). *Enterprise Architecture Fundamentals Using the Pagoda Blueprint*. Izzard Ink.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Fodor, J.A. (1992). *A Theory of Content and Other Essays* (Representation and Mind series). Bradford Books.

Gorichanaz, T. (2017). *Applied Epistemology and Understanding in Information Studies*. University of Boras: Sweden.

Guizzardi, G., Gerd, W., Andrade, J., and Guizzardi, R. (2015). Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Applied Ontology* 10: 3–4.

Guizzardi, G., and Guarino, N. (2024). Explanation, semantics, and ontology. In: *Data & Knowledge Engineering*. Amsterdam: Elsevier.

Halpin T., and Morgan, T. (2008). *Information Modeling & Relational Databases*, 2d edn, Morgan Kaufman.

Hedden, H. (2022). *The Accidental Taxonomist*, 3rd edn. Oxon: Information Today.

Hofstadter, D.R. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books.

Huyen, C. (2024). *AI Engineering: Building Applications with Foundation Models*, 1st edn., O'Reilly.

Hurwitz, J.S. and Thompson, J.K. (2024). *Causal Artificial Intelligence, The Next Step in Effective Business AI*. Wiley.

Jablonka, E., and Lamb, M.J. (2014). *Evolution in Four Dimensions,* Revised edn: *Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life*. Bradford Books.

Jacobson, I. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.

Jensen, M., De Colle, G., Kindya, S., More, C., Cox, A., and Beverley, J. (2019). Common Core Ontologies. Available from: https://arxiv.org/pdf/2404.17758.

Johnson, M. (1987). *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago Press.

Jurisica, I., Mylopoulos, J., and Yu, E. (2004). Ontologies for knowledge management: An information systems perspective. In: *Knowledge and Information Systems*. London: Springer-Verlag.

Kahneman, D. (2013). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Keet, M. (2018). *An Introduction to Ontology Engineering*. Rickmansworth: College Publications.

Keet, M. (2023). *The What and How of Modelling Information and Knowledge. From Mind Maps to Ontologies*. London: Springer.

Kotis, K.I., Vouros, G.A., and Spiliotopoulos, D. (2020). *Ontology Engineering Methodologies for the Evolution of Living and Reused Ontologies: Status, Trends, Findings and Recommendations*. Cambridge University Press.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things, What Categories Reveal About the Mind*. University of Chicago Press.

Lario, R., Kawamoto, K., Sottara, D., Eilbeck, K., Huff, S., Del Fiol, G., Soley, R., and Middleton, B. (2023). A method for structuring complex clinical knowledge and its representational formalisms to support composite knowledge interoperability in healthcare. *Journal of Biomedical Informatics* 137: 104251.

Lenat, D.B., and Guha, R.V. (1989). *Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.

Liskov, B. (1993). *A New Definition of the Subtype Relation*, B00FNBIFAI.

Liszka, J.J. (1996). *A General Introduction to the Semiotic of Charles Sanders Peirce*. Indiana University Press.

Merrihew, R. (2024). *What is, and What is in Itself: A Systematic Ontology*. Oxford: Oxford University Press.

Miller, A., Davis, M., and Dana Neacsu, D. (2023). *Intellectual Property, Patents, Trademarks, and Copyright in a Nutshell*, 7th edn. West Academic Publishing.

NFDI. MatWerk Ontology (MWO) (2025). Available from: https://github.com/ISE-FIZKarlsruhe/mwo

Noyes, J.L. (2007). *Artificial Intelligence with Common Lisp: Fundamentals of Symbolic and Numeric Processing*, Boston: Houghton Mifflin College Div.

Ogden, C.K.I.A., and Richards, A. (1923). *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Magdalene College, University of Cambridge.

Olesen-Bagneux, O. (2025). *Fundamentals of Metadata Management: Uncover the Meta Grid and Unlock IT, Data, Information, and Knowledge Management*. O'Reilly.

Ostler, N. (2005). *Empires of the Word, A Language History of the World*. HarperCollins Publishers.

Ozdemir, S. (2024). *Quick Start Guide to Large Language Models: Strategies and Best Practices for ChatGPT, Embeddings, Fine-Tuning, and Multimodal AI*. Addison-Wesley.

Pavlyshyn, V. (2025). Semantic Spacetime: Understanding Graph Relationships in Knowledge Representation. Available from: https://ai.plainenglish.io/semantic-spacetime-understanding-graph-relationships-in-knowledge-representation-1175cab8282d

Pearl, J., and Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. New York: Basic Books.

Porello, D., Vieu, L., Terkaj, W., Borgo, S., Compagno, S., and Sanfilippo, E.M. (2024). *DOLCE in OWL: The Core Theory*. University of Twente.

Provost, F., and Fawcett, T. (2013). *Data Science for Business*. O'Reilly.

Prudhomme, T., De Colle, G., Liebers, A., Sculley, A., Xie, P.K., Cohen, S., and Beverley, J. (2025). A semantic approach to mapping the provenance ontology to basic formal ontology. *Scientific Data* 12: 282.

Raschka, S. (2024). *Build a Large Language Model (From Scratch)*. Manning.

Riedl, M.O. (2023). A Very Gentle Introduction to Large Language Models Without the Hype. Available from: https://mark-riedl.medium.com/a-very-gentle-introduction-to-large-language-models-without-the-hype-5f67941fa59e.

Riedl, M.O., and Desai, D.R. (2025). AI Agents and the Law. Available from: https://arxiv.org/pdf/2508.08544

Schalley, A.C. (2019). Ontologies and ontological methods in linguistics. *Compass Journal* https://doi.org/10.1111/lnc3.12356

Sharp, B. (2020). *The EMS OODA Loop*. Independently Published.

Singh, A. (2022). *Graph Database Modeling*, 2nd edn. Independently Published.

Sowa, J.F. (2019). *Natural Logic is Diagrammatic Reasoning About Mental Models*. Procedia.

Sowa, J.F. (2010). Ontology. Available from: https://www.jfsowa.com/ontology/

Sowa, J.F. (2020). Knowledge Graphs for Language, Logic, Data, Reasoning. Available from: https://jfsowa.com/talks/kng.pdf

Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Glove: Brooks/Cole.

Stanford Encyclopedia of Philosophy (2021). *Deontic Logic*., Stanford University.

Sterling, L., and Shapiro, E., (1986). *The Art of Prolog: Advanced Programming Techniques*, Cambridge: MIT Press.

Taleb, N.T. (2007). *The Black Swan: The Impact of the Highly Improbable*. Random House.

The Art of Service (2020). *Zachman Framework A Complete Guide*. Zachman Framework Publishing.

Von Fint, K., and Heim, I. (2011). *Intensional Semantics*. MIT Press.

Wegerif, R., and Casebourne, I. (2025). *A Dialogic Theoretical Foundation for Integrating Generative AI Into Pedagogical Design*. British Journal of Educational Technology.

Williamson, T. (2023). *The Patterns of Reality*. Aeon Essays.

Winograd, T. (1983). *Language as a Cognitive Process: Syntax*. Boston: Addison-Wesley.

Zins, C. (2007). Conceptual approaches for defining data, information, and knowledge. *Journal of the American Society for Information Science and Technology* 58(4): 479–493.

Index

a

accountability 3.2.4.4
agency 2.4.2.2.2
agent 1.2; 2.4; 5.2; 7.5; 8.5; 10.3.3
  behavior 1.2.1.3.3; 2.4.2.2.2
agentic 1.1.1.1; 1.2.1.4.1; 5.2.3; 5.2.3.1; 5.2.3.3; 5.3.1; 5.3.3.3; 5.3.4.2; 6.3.2.1; 6.5.3.1; 7.5; 10.3.1.1
  architecture 5.2
  collaboration 5.3.1; 7.5
alignment 1.3.1; 2.2.3; 2.3.1.1; 2.4.2; 3.1; 3.2; 3.3; 4.1.2.2; 4.2.2.1; 4.3.2.2; 4.4.1.1; 4.4.1.2; 4.4.2.1; 4.4.2.2; 4.4.3.2; 5.2.1; 6.2.1.3; 6.3.1.2; 6.4.1.1; 7.2.2.2; 7.3.2; 7.4.2.2; 7.5.5.1; 8.2.3.1; 8.4.2.2; 8.4.3.1; 9.1.3.1; 9.1.3.2.1; 9.1.3.2.2; 9.1.3.2.3; 9.1.4; 9.2.1
  conceptual 9.1.3.2
  modalities 9.1.3.2.1
  semantic 8.4.3.3.1; 9.1.3.1
anchors 2.3.1.1; 2.3.2.2; 3.1.2.1.1; 3.1.2.1.3; 6.3.1.2; 6.4.1.2; 8.1.2.3; 8.2.3.2; 8.3.1; 8.3.2; 8.4.2; 8.4.3.2; 8.5.2.1; 9.1.1.1; 9.2.1.1; 9.2.2.2
architecture
  collaborative 5.2.3
  enterprise 8.1.1; 8.5.2.2; 9.2.2.2
  knowledge 1.2.2.1
  knowledge-driven 5.2.3.3
Aristotle 2.2.1
  3 unities rule 8.4.3.1
artificial intelligence (AI) 1.1
  generative AI 1.1.1.5; 4.2.3.1; 5.2.2.2; 5.3.4.1; 6.2.2; 7.3.1.2; 7.4.1.3; 8.5.2.4
  neural networks 1.1.1; 5.1.3; 5.3
aspect 2.3.2; 3.1.2.1.2; 6.2.1.3; 6.4.1; 9.1.4.2; 10.1.2.3

b

blockchain 4.2.2.3.2

c

capture 2.1.2.1; 4.2.1.2; 5.3.4.2; 7.4.2.2; 8.3.3; 8.4.1.1
causal chain. *see* causality
causality 3.3.2.1.2; 4.4.3.3; 6.2.2.2
  analytics 6.4.1
  bayes 6.2.2.1; 6.4.2.2.1; 6.5.2.2; 7.5.3.1.2
  directed acyclic graphs (DAG) 6.4.1
  explanatory variable 6.4.1
code 8.5.2.3; 8.5.2.3.3; 8.5.4; 8.5.4.1; 9.1.2.1.3; 9.1.2.1.2
  coding 8.5.1.2; 8.5.2.4; 9.3.2.1.2
cognition 1.1.1.2; 2.1.1.3; 6.2.2.4
  beliefs 6.4.2.1.3; 2.1.2.2; 5.3.2
  judgement 1.1.2.3; 1.3.2; 4.1.3; 5.2.3.2; 5.3.3.3; 6.3.2.1; 7.3.1; 7.4.1; 7.5.4; 8.1.1; 9.2.1.4; 10.3.3

commitment 4.2.2.1; 7.1.1; 7.3; 7.4.2; 7.5.4. *see also* decision-making
  conceptual 4.2.2.1.1; 9.2.1
  ontological 2.4.2; 2.4.2.3; 3.1.2.1.1; 3.2.4.1; 9.1; 9.2.1; 9.3.2
  semantic 3.2.2
  space 7.1.2
communication 5.3.1
  sign 2.1.1.2; 5.1.2
  symbol 5.1.2
complexity 4.4
  extensional (2D) 4.4.1.1
  intensional (3D) 4.4.1.2
  management 4.4.2.1
computation paradigm 5.3.2
connector 2.1.1.3
  cardinality 2.1.1.3
  conceptual 2.2.3.2
  contribution 2.1.1.3; 2.2.3.2; 3.1.1.1; 3.2.3.2.2; 3.2.3.3; 3.3.2.1.1
  functional 2.1.1.3
  intrinsic 2.1.1.3
  kernel 10.1.2.2
  logical (association) 2.3.1.2
  relationship 2.2.3.1
  semantic 2.1.1.3
  syntactic 2.1.1.3
consistency 1.2.1.2; 2.3.2.3; 3.1.2.2
context 1.1.1.5; 1.2.1.4.2; 1.1.1.5; 2.2.1; 3.1.1.1; 3.1.1.2; 3.1.2.2; 3.2.4; 3.2.4.4; 4.1.1; 4.1.3.1; 4.4.1.1; 4.4.1.2; 4.4.2.2; 5.1.1; 5.1.2; 5.2.1; 5.2.3.1; 5.2.3.3; 5.3.3.2; 5.3.3.3; 5.3.4.1; 6.2.2.3; 6.3.1.2; 6.3.2.1; 6.5.2.3.1; 6.5.2.3.2; 7.1;7.3.1.1; 7.3.1.4; 7.4.2.3; 8.1.2.3; 8.4.2; 8.5.4.1; 9.1.1; 9.1.2.3.1; 9.1.3.1.2; 9.2.1.3
  bounded 1.1.1.5; 3.2.4.4; 4.2.2.2.2; 8.2.2.2.1; 8.5.2.1; 8.5.2.1.1; 8.5.2.3.1; 8.5.2.4; 8.5.4; 9.2; 10.3.1.1
correlation 6.2.2.2
cybernetics 4.4.2.2; 7.2.1.1; 7.5.6.1; 8.5.4.3
  entropy 4.4.2.2
  homeostasis 3.2.3.4; 3.3; 4.4.2.2; 7.5.6.1
  macrostate 7.2.1.1
  microstate 7.2.1.1
  osmosis 3.2.3.4; 3.3; 4.3.1.3; 4.4.2.2; 7.5.6.1

d

data product 4.2.1.3.3
decision 7.2.2. *see also* time frame
  operational 4.4.2.1; 5.2.1; 5.3.1; 7.3.2; 7.5.4.1; 7.5.5.1; 8.5.3.1
  strategic 4.4.2.1; 7.3.2; 7.5.4.1
  tactical 4.4.2.1; 7.3.2; 7.5.4.1
decision-making 7. *see also* process
  accountability 7.3.1.2.2
  action 7.5.5
  adversarial 7.4.1.3
  agentic 7.5
  assessment 7.1.1; 7.5.6
  collaborative 7.4.1.2
  decision 7.5.4
  entitlement 7.2.1.2; 7.3.1.2.1
  explainability 7.3.1.2.2
  horizon 7.3.2
  knowledge-based (KBDM) 7.5.1
  observation 7.5.2
  observation-orientation-decision-action (OODA) 7.4.2.2
  one-sided 7.4.1; 7.4.1.1
  orientation 7.5.3
  party 7.3.1.2
  process 7.4.2
  risks 7.2.2.2; 7.3.2; 7.5.4.1
  template 7.2.2
  traceability 1.1.1.5; 3.2.4.4; 4.2.2.2; 6.2.2; 7.3.1

transparency 1.1.1.5; 3.2.4.4; 4.2.2.2; 6.2.2; 7.3.1
decision support system (DSS) 7.1.2
decoding 5.2.1
definitions 9.1.4.1.1
digital twins 3.2.3.4
domain 3.1.2.1.3; 3.1.2.2; 4.4.2; 6.4.2.1.3; 7.2.1.2; 8.4.2.2; 9.1.1; 9.3.2.3
domain-driven design 8.5.2.1
domain-specific language (DSL) 8.5.2.1.2; 8.5.2.2.1. *see also* model

e

earmarking 8.4.1.2
elementary meaningful unit (EMU) 6.1.2; 6.2.1.1
elicitation 3.3.1.1; 7.4.2.2; 8.3.3; 8.4.1.2
emerging 5.3.4.2
empirical 1.2.2; 1.2.3; 2.3; 2.4.2; 1.3.5; 3.1.2.2; 5.3.3; 6.5.2.2; 7.2.2.2; 7.3.2; 7.4.1.3; 8.3.2.4
empirical reasoning 5.3.3; 6.1.1; 6.2.2.2; 6.4; 6.4.2.1; 6.5.1.2; 6.5.2.2
encoding 5.2.1
engineering
  digital 1.1.1.1; 1.2.2.1; 2.1.2.1; 2.3.2.2; 2.4.2.2.2; 3.2.3.4; 4.2.1.1; 4.2.2.3.2; 4.3.2; 7.4.2.2; 7.5.5.1; 9.1.3.2.1; 9.2.2.1; 10.3.1.1
  environment 1.2.1.4.1; 1.2.1.4.2; 2.1.1.1; 2.4.2.2.2; 3.1.2.2; 3.2.3.3; 3.2.3.4; 3.2.4.3; 3.3.1.1; 4.2.1.3.3; 4.2.2.1; 4.2.2.3.1; 4.2.2.3.2; 4.3.2.1; 4.4.1.1; 4.4.1.2; 4.4.2.1; 4.4.3; 4.4.3.2; 6.3.2.2; 6.4.2.1.1; 7.1.2; 7.3.1.4; 7.4.1.1; 7.4.1.2; 7.4.1.3; 7.4.2.1; 7.5.6.1; 7.5.6.3; 8.1.2.1; 8.1.2.3; 8.2.1; 8.2.2; 8.2.2.3
  flow-based 8.5.1.1
  granularity 8.4.3
  physical 1.2.1.4.1; 1.2.1.4.2; 1.2.1.4.3; 2.1.1.1; 2.1.3.1; 2.2.3; 2.4.2.1.3; 3.2.3.4; 4.2.1.2; 4.3.2.2; 7.5.6.3; 8.1.2.1; 8.2.1; 8.4.3.1; 8.4.3.3.1; 8.5.2.2.1; 9.1.2.1.2; 9.1.2.1.5; 9.1.2.2; 9.1.2.3.1; 9.1.3.2.1; 9.2.1.4; 9.2.2.1
engineering knowledge 9
engineering system 8
environment
  business 4.2.2.1; 4.4.2.1; 8.3.1.2
  institutional 4.1.1; 4.2.2.3.1; 4.3.1.2. *see also* environment, statutory
  statutory 4.1.1; 4.2.1.3.2; 4.2.2.3.1; 4.3.1.2; 7.3.1.2.1; 7.5.6.4; 9.1.1.1
epistemic 1.2.1.4; 1.2.2.2; 2.2.2; 2.2.3.3; 3.1.3.2; 3.2.2.1; 6.3.2.1; 7.5.3.1.2
evolution
  epigenetics 4.4.3.1
  genetics 4.4.3.**1**
  natural evolution 4.4.3.1
  organizations 4.4.3.2
  resilience 4.4.3.3
  sustainability 4.4.3.3
  systems 4.4.3

f

footprint 1.1.1.5; 2.2.1; 2.3.2.2; 2.4.2.1.3; 3.2.4.4; 4.1.1; 6.2.2.3; 6.3.2.2; 6.5.1.1; 7.2.2.1; 7.4.1; 7.4.1.2; 7.4.2; 8.4.3.1; 8.4.3.2; 9.1.1; 9.1.3.1.1; 9.3.1.2.1; 9.3.1.2.2; 10.2.3.2

g

generative concept model 5.3.4.2
generative language model 5.2; 2.1.3.2
  embedding 5.2.3.1
  hallucinations 6.3.2.1
  RAG 5.2.3.1
  tokens 5.2.3.1
  training 5.2.3.1
  tuning 5.2.3.1

graph 9.2
  conceptual 2.2.3.2
  hypergraph 9.1.2.4.1
  knowledge 2.2.3.3
  semantic graph 3.1.1.2; 6.4.1.2; 10.2.3.2

h

holistic 3.3.2; 4.2.2.1; 4.3.2.1; 4.4.2.2; 6.2.2; 8.1.2.1
horizon 3.2.4.3; 7.3.2. *see also* visibility

i

integration 1.2.2.1
  functional 4.1.2.2
  shallow 2.3.1.1; 3.1.1.1; 4.1.2.1
  technical 4.1.2.1
intellectual property 4.3
  copyright 4.3.1.2
  creative work 4.3.1.2
  GDPR 4.3.1.2
  patent 4.3.2.2
intelligence
  artificial 1.1.1; 1.1.2; 3.2.3.4
  business 4.1.2.2; 7.4.1.3
  collective 3.3.1.2; 3.3.2
  operational 9.3.2.2.1
interoperability 1.2.2.1; 2.3.1; 3.2

k

KEOPS 9
kernel 2.2.3.3; 9.3.2.3; 10.1.1.1; 10.1.2; 10.2.2.1; 10.2.2.2; 10.2.3.4; 10.3.2.1; 10.3.3
knowledge engineering 1.2.3; 3.2
  abstraction 1.2.1.3.3; 1.2; 1.3.5; 2.1.1.3; 2.2.2; 2.2.3.2; 2.3.1.1; 2.4.1.2; 3.2; 8.5.3.2; 9.2.2.2; 9.3; 10.3; 10.4
  bounded cases 9.3.1.2.1
  expert system 1.1.3; 7.1.2.3
  explanatory cases 9.3.2.2.1
  functional cases 9.3.2.2.2
  reuse 1.2.2.2; 3.2.1
  unbounded cases 9.3.1.2.2

l

language 5.1.3
  formal 1.2.1.2; 1.2.1.3.2; 3.1.2.2; 3.2.3.1.2; 5.1; 5.3.3.3; 6.1.2; 6.2; 10.2.3
  grammar 5.1.3
  lexicon 5.1.3
  logographic 5.3.1
  metaphor 5.1.3; 6.4.2.2.2; 6.5.2.3.2
  metonymy 6.4.2.2.2; 6.5.2.3.1
  phoneme 5.3.1
  pragmatics 1.2.2.1; 5.1.3; 5.3.3
  semantics 2.1.1.1; 2.1.1.3; 2.2.3.2; 5.1.3; 10.4
  semantic space 5.3.4.1. *see also* vector
  semiotics 1.3.1; 6.3.2.2; 6.5.2.3
  syntax 2.1.1.1; 2.1.1.3; 2.2.3.2; 2.3.2.2; 2.4.1.2; 6.2.1.1; 8.4.3.2; 10.4
  term 9.1.3.1.1
  ubiquitous 8.5.2.1
  vocabulary 5.3.2.1; 6.3.1; 6.4.1.2; 6.4.2.1.3
learning 1.3.2. *see also* evolution
  collective learning 4.1.3.3; 6.5.3.2
  deep learning 5.2.1; 5.2.2
  learning curve 4.2.2.1.3; 6.4.2.2.1; 7.5.5.1; 7.5.6.2
  machine learning 1.1.1.2; 1.1.1.4;
logic 3.1.2.2. *see also* reason
  abduction 6.5.2.1
  clause 6.2.2.2
  CLIF 1.2.1.2; 4.1.2.1; 10.2.3.3
  deduction 6.5.2.1
  formal 1.2.1.2; 1.2.1.3.2; 1.2.2.2; 2.3.2.4; 3.1.1.1; 3.1.2.2; 3.2.3.1.2; 3.3.2.1.2; 4.1.2.1; 4.3.1; 5.1; 5.3.3.3; 6.1.2; 6.2; 6.3.2.2; 8.3.2.4; 9.1.2.4.1; 10.2.3
  fuzzy 6.3.1
  induction 6.5.2.1
  inference 6.5.2.1

intuition 6.5.2.1
nominal 6.2.1.1; 6.3.2
predicate 6.1.2; 6.2.1; 6.3.1; 10.2.3.3
Prolog 6.2.1.2; 9.1.2.4.2; 4.1.2.1
proposition 6.1.2; 6.2.1; 6.3.1

m

maturity 3.3.2.1.3
meaning 5.3.4.1
metadata 4.2.1.3; 4.2.2.3; 6.4.2.1.1; 7.5.2.1
modality 1.2.2.2; 2.1.2.2; 2.2.2; 2.4; 9.1.4.2
agency 2.4.2.2.2; 9.1.2.1.4
communication 1.1.1.2; 2.4.2.2.2; 3.1.2.2; 3.2.3.4; 4.1.3.1; 4.4.2.2; 4.4.3.2; 5.1.1; 5.1.2; 5.1.3; 5.2.3.3; 5.3.1; 5.3.3.3; 5.3.4.2; 6.3.2.1; 6.5.3.1; 8.1.1.3; 8.2; 8.2.2.3; 8.3.2; 8.4.3.1; 8.5.2.1.1; 9.1.2.1.5; 9.1.2.3; 9.1.4.2.2; 9.2.1.4; 10.3.1.1
identification 1.2.1.1; 1.2.1.4.1; 1.3.1.2; 2.4.2.1.4; 3.2.2.1; 4.3.2.1; 7.5.3.2; 8.5.2.2.1; 9.1.2.1.3; 9.1.2.3.1; 9.1.3.1.1; 9.1.3.2; 9.1.4.2.2; 9.2.2.2
instantiation 1.2.1.1; 2.4.2.1.2; 3.2.2.2; 9.1.2; 9.1.3.2; 9.1.4.2.2; 10.1.2.1; 10.3.3
structure 2.4.2.2.1; 9.1.2.1.6
temporality 2.4.2.1.1; 6.4.2.2.1; 9.1.2.1.1; 9.1.2.3.1; 9.1.3; 9.1.4.2.2
veracity 2.4.2.1.5; 9.1.2.1.7; 9.1.4.2.2; 9.2.1.5
model
actionable 6.5.1.1; 8.1.1.1; 8.5.2.3.1; 9.1.1.1; 9.1.1.2; 9.1.2.3; 9.3; 9.3.2.2.1
conceptual 2.2.3
data 2.1.3.1
entity-relationship (ER) 2.2.3.1
executable 8.5.2.3.1
logical 2.1.3.1; 2.2.3
object-role (ORM) 2.3.2.4; 3.2.3.1.2
physical 1.2.1.3.2; 1.2.1.4; 2.1.1.2; 2.1.3
relational model 2.3.2.3; 2.3.2.4; 3.2.3.1.2; 5.2.3.2; 8.4.3.2
UML 2.2.1; 3.2.3.1.1

n

National Information Exchange Model (NIEM) 3.1.2.1.1

o

Occam's razor 8.5.1.1.2; 9.1.2.1; 9.1.4.1.1
ontological prisms (OP) 2
actual facts 2.1.2.2
asserted facts 2.1.2.2
assessed facts 2.1.2.2
categories 2.3
concepts 2.2
concepts modalities 2.2.2
databases 2.1.2.1
datasets 2.1.2.1
documents 2.1.1
experience 3.3.1.3; 9.1.4
facts 2.1
perceptions 3.3.1.1; 9.1.3
projections 3.3.1.2;
virtual facts 2.1.2.2
ontological timescale 9.2.2.1
ontology 2.1.1
basic formal ontology (BFO) 1.2.1.3.2
common core (CCO) 1.2.1.3.4
continuant (endurant) 1.2.1
continuant 1.2.1
DOLCE 1.2.1.2
foundational 1.2.1
heterogeneous 3.1.2
homogeneous 3.1.1
lightweight 6.3.1.1
M*R/K reference framework 1.2.2.1
moment 1.2.1
names 2.1.1.2
occurent (perdurant) 1.2.1

ontology (*Cont.*)
ontoUML 1.2.1.1
perdurant 1.2.1
substance 1.2.1
tropos 1.2.1.3.3
unified formal ontology (UFO) 1.2.1.1
organizational 1.2.1.3.3; 1.2.1.4.2; 1.2.2.1; 1.3.4; 3.1.2.1.3; 3.2.4; 4.1.3; 4.4.2; 6.5.3; 7.2.2.2; 7.3.1.2.1; 7.3.2; 7.4.1.1; 7.5.6.4; 8.1.2.3; 8.2.2.2.2; 8.2.3.1; 8.3.1.2; 8.3.2; 8.4.2.1; 8.5.1; 9.1.2.1.3; 9.1.2.3; 9.2.1.4; 9.2.2; 10.2.3.1; 10.3.1.1; 10.3.3
organizational unit 3.2.4; 8.5.1.1
OWL 10.1.1
annotation property 10.1.1
data property 10.1.1
hierarchy 10.1.1
individuals 10.1.1
object property 10.1.1

p

parthood (mereology), 2.1.1.3. *see also* structure
partition 9.1.4.2.1. *see also* variant
pattern 9.1.2.3. *see also* process
blueprint 9.1.2.3.1
iterative 8.5.1.1
phased 8.5.1.1
profile 9.1.2.3.2
template 9.1.2.3.3
philosophy 1.3.2
process 9.1.2.2
collaborative 7.4.1.2. *see also* decision-making
declarative 2.4.2.3; 6.2.1.2; 6.5.1.1; 9.1.2
imperative 9.1.2
incremental 2.2.2; 3.2.2.2; 8.2.2; 8.3.1; 9
iterative 2.2.2; 9.1.2; 3.2.2.2; 7.4.2.2; 8.1.2.3; 8.2.2; 8.3.1; 9
phased 7.4.2.1; 8.1.2.3
responsibility 8.2.3.2. *see also* decision-making
sequential process 7.4.2.1; 7.4.2.2
project 9.1.2.3; 9.2.2.4; 10.3.2.2

q

quality 4.2.2.1.3; 8.2.3.2; 8.5.4; 8.5.4.1

r

realm
extensional 2.1
intensional 2.2
logical 2.3
reason 6.1. *see also* logic
causality 6.2.2.2
correlation 6.2.2.1
explanation 6.2.2.4
interpretation 6.2.2.3
regulation 4.1.1. *see also* rule
compliance 4.2.2.2.1; 4.3; 6.2.2; 7.2.1.2; 7.2.2; 9.1.1.1; 9.2.1.1
privacy 4.2.2.2.1; 4.3
relational schema 9.1.4.3
representation 5.1.3
agnostic 9.1.1; 9.2
inheritance 2.1.1.3; 2.3.1.2
mental 1.2.1.4; 5.3.2; 6.1.1; 6.2.2.3; 6.4.2.1.3; 7.5.6.1
surrogate 3.2.3.2.2
requirements 8.3.2.4; 8.4.2.2; 8.4.3.1; 8.4.3.2
root 8.5.2.1; 9.2.1.4
rule 6.2.1.2; 6.5.1.1; 6.5.1.3;
constraint 2.5.2.3
contingency 6.5.1.2
deontic 6.5.2.4; 7.2.2.2
elective 6.5.1.2; 8.4.3.3.1; 9.1.2.3.1
mandatory 2.3.2.3; 6.5.1.2; 8.4.3.1; 10.1.2.2

s

security 4.2.1.3.2; 4.3.1.3; 6.2.2; 8.5.4.1; 9.1.2.1; 9.1.2.2
  breach 8.5.4.2
semantics 5.3.3
  extensional 1.2.1.4; 5.3.3.1
  intensional 1.2.1.3.3; 1.2.1.4; 5.3.3.2
  language semantics, layers 5.3.3.3
SHACL 4.1.2.1
SKOS 1.2.3; 2.2.3.3; 6.3.1.1; 6.5.1.3
structure 2.4.2.2.1; 4.3.1.1; 7.4; 8.5.2.1.1; 9.1.2.1.6; 9.2.1.1; 10.4.1.1. *see also* parthood
subsumption 2.1.1.3; 2.2.3.2; 3.1.2.1.1; 10.2.2.1.2
system 4.4.3
  expert 1.1.3; 7.1.2.3
  legacy 3.2.3.2.1
  plasticity 3.2.3.4; 4.4.3.2
  versatility 3.2.3.4; 4.4.3.2
  viable 3.2.3.4
system engineering 8
  business cases 8.2.2.1
  cases 8.2.2
  function cases 8.2.2.2.3
  model based system engineering (MBSE) 3.2.3.3; 8.5.1; 8.5.2
  model driven architecture (MDA) 8.5.2.2
  pagoda blueprint 8.2.1
  system cases 8.2.2.2.3
  use cases 8.2.2.2.1
  user stories 8.2.2.1

t

taxonomy 1.3.1; 1.3.4.3; 3.1.2.1.2; 3.1.2.2; 4.2.1.1; 4.3.2.1; 4.4.2.1; 6.1.1; 6.2.2.1; 6.3.1.1; 7.2.1.2; 8.4.2.2; 9.1.1.2; 9.1.2.2; 9.1.4; 9.3.2.3; 10.2.3.2. *see also* variant
tests 8.5.4.1; 8.5.4.2; 8.5.4.3
thesauri 1.3.1; 1.3.4.3; 3.1.2.1.1; 3.1.2.2; 4.2.1.1; 4.3.2.1; 4.4.2.1; 5.1.3; 5.3.3.1; 6.1.1; 6.1.2; 6.2.2.1; 6.3.1; 6.3.2.2; 6.4.1.2; 7.2.1.2; 8.4.2.2; 9.1.1.2; 9.1.2.1; 9.1.2.2; 9.1.4; 9.3.2.3; 10.2.3.2
things 1.2.1.1; 1.2.1.4; 1.3.1; 1.3.3; 2.1.1; 2.1.2.1; 2.2.1; 2.2.2; 2.3.1.; 2.4.2; 3.2.2.1; 3.2.3.3; 4.3.2; 5.1.1; 6.2.1.3; 6.5.1.2; 8.5.4; 9.1.1; 9.1.2.1; 10.1.1
  artifact 1.2.1.4.1; 3.2
  designed 1.2.1.4.1;
  symbolic artifact 4.2.2.3.2; 4.3.2.2; 5.1.2; 8.5.1.2; 9.3.1.1
time 3.2.3.2.1; 3.2.3.3; 3.2.3.4
time frame 3.2.4.3. *see also* horizon
  operational (deterministic) 3.2.4.3; 4.4.1.2; 7.2.2; 7.4.1.1; 7.5.4.1
  planning 3.2.4.3; 3.3.1.2; 7.5.3.1.1; 7.5.4.1
  strategic (non stochastic) 3.2.4.3; 4.4.1.2; 7.2.2; 7.4.1.3; 7.5.4.1
  tactical (stochastic) 3.2.4.3; 4.4.1.2; 7.2.2; 7.4.1.2; 7.5.4.1; 8.5.4.1
truth 1.1.1.5; 1.1.2.2.3; 1.2.2.2; 2.4.2.3; 6.1;6.2.1; 6.5.1.2; 9.1.2.1.7
truth model 9.2
truth value 10.2.3.4
type 1.2.1.1; 1.2.2.1; 2.3.1.1; 4.3.2.1; 6.2.1. *see also* facts
  powertype 2.3.1.3; 8.4.3.2; 9.1.4.2. *see also* variant
  subtype 2.3.1.2; 3.1.2.1.2. *see also* variant

v

validation 8.5.2.3.2; 8.5.3; 8.5.3.2.1; 8.5.4
value chain 4.2.2
variant 2.3.2.4; 8.4.3; 10.2.2; 10.3.3

vector 5.2.1; 9.1.3.2
  semantic 6.3.2.2; 9.1.2.4.1
  space 9.1.2.4.1
verification 5.2.2.1; 8.5.3.2.3
visibility 7.3
vulnerabilities 8.4.1; 8.4.2

W

workflow 8.2.3

X

XML 2.1.1.1; 4.1.2.1

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.