UNIVERSITÄT HOHENHEIM

Theory-based Knowledge Acquisition

for Ontology Development

A dissertation

submitted to the Faculty of Business, Economics and Social Sciences of the

University of Hohenheim

in partial fulfillment of the requirements for the degree of

Doctor of Economics (Dr. oec.)

by

Andreas Scheuermann

July 2016

Principal advisor:	Prof. Dr. Stefan Kirn
Co-advisor:	Prof. Dr. Ralph Bergmann
Chair of the examination committee:	Prof. Dr. Christian Ernst
Dean:	Prof. Dr. Dirk Hachmeister
Date of oral examination:	7. July 2016

Abstract

This thesis concerns the problem of knowledge acquisition in ontology development. Knowledge acquisition is essential for developing useful ontologies but it is a complex and error-prone task. When capturing specific knowledge about a particular domain of interest, the problem of knowledge acquisition occurs due to linguistic, cognitive, modelling, and methodical difficulties. For overcoming these four difficulties, this research proposes a theory-based knowledge acquisition method.

By studying the knowledge base, basic terms and concepts in the areas of ontology, ontology development, and knowledge acquisition are defined. A theoretical analysis of knowledge acquisition identifies linguistic, cognitive, modelling, and methodical difficulties, for which a survey of 15 domain ontologies provides further empirical evidence. A review of existing knowledge acquisition approaches shows their insufficiencies for reducing the problem of knowledge acquisition.

As the underpinning example, a description of the domain of transport chains is provided. Correspondingly, a theory in business economics, i.e. the Contingency Approach, is selected. This theory provides the key constructs, relationships, and dependencies that can guide knowledge acquisition in the business domain and, thus, theoretically substantiate knowledge acquisition.

Method construction uses an approach from the field of Method Engineering, which defines how to develop a tailored method with respect to specific requirements on method design, functionality, components, and the underlying assumptions. The development of the method for theory-based knowledge acquisition covers the specification of the (method and outcome) metamodel, activity model, outcomes, roles, and techniques.

The evaluation comprises two descriptive approaches to demonstrate the proposed method's utility. First, a criteria-based approach evaluates the method with respect to design-related, functional, and component-related requirements. Second, a scenario-based evaluation applies the method within a scenario from the domain of intermodal transport chains for acquiring knowledge to build a domain ontology.

The contribution of this research is a theory-based knowledge acquisition method for ontology development. The application and usefulness of this method is demonstrated for a particular domain (transport chains) and uses a particular theory of business economics (the Contingency Approach).

Acknowledgements

In the first place, I would like to thank Prof. Dr. Stefan Kirn as my supervisor and as a first reviewer. Prof. Kirn offered me the opportunity to work for a doctoral degree after my graduation. He provided the framework conditions and gave me the necessary freedom to carry out my research. I am very grateful for that.

I would also like to thank Prof. Dr. Ralph Bergmann from the University of Trier who, without hesitation, accepted the request to serve on the examination committee as a second reviewer. Many thanks to Prof. Dr. Christian Ernst from the University of Hohenheim who served on the examination committee as chairman.

Additionally, I am grateful to all the members of the Department of Information Systems 2 for the excellent and friendly working atmosphere. Particularly, I would like to thank Dr. Joerg Leukel for his support and insightful comments and feedback on my thesis. Many thanks also to my former colleagues for the lively discussions on research topics and beyond. Also, many thanks to the project management and technical team members for their support.

Similarly, I would like to thank all project[1] partners and collaborators. In particular, I would like to mention my colleagues from the University of Leipzig and the Karlsruhe Institute of Technology.

During my doctoral studies, I spend three months at the Knowledge Media Institute (KMi) at the Open University and was privileged to work with Prof. Dr. Enrico Motta and Dr. Mathieu d'Aquin. Many thanks for this collaboration, which was very valuable for me and my research.

Last but not least, I would express my great thanks and gratitude to my family. My parents Elisabeth and Paul as well as my brother Stefan have supported me at all time in every respect. Mostly, I am eternally grateful to my wife Kathrin.

[1] InterLogGrid – Intermodal Logistics and IT Services (funding code: 01IG09010E)

Dedicated to Kathrin

Outline

Table of Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ATHENA	Advanced Technologies for Interoperability of Heterogeneous Networks and their Application
CRLM	Configurable Role-Limiting Methods
CSCMP	Council of Supply Chain Management Professionals
CYC	Cyc Method
DKAM	Designed Knowledge Acquisition Method
DL	Description Logics
DOGMA	Developing Ontology-Grounded Methods and Applications
EO	Enterprise Ontology
ERM	Entity-Relationship Model
FOL	First-Order Logic
GenCLOn	Ontology for City Logistics
GFM	Grüninger and Fox’s Method
GODF	A Generic Ontology Development Framework
GSCF	Global Supply Chain Forum
GT	Generic Tasks
HSM	A Helix-Spindle Model for Ontology Development
IDEF1X	Integration DEFinition for Information Modeling
IS	Information Systems
IT	Information Technology
ITC	Intermodal Transport Chain
KACTUS	Modelling Knowledge About Complex Technical systems for multiple USe
KADS	Knowledge Acquisition and Documentation Structuring
KIF	Knowledge Interchange Format
LO	Logistics Ontology
LOPLHL	Logistics Ontology for Production Logistics and Hospital Logistics
MCO	Mass Customisation Ontology
MET	METHONTOLOGY
MIKE	Model-based and Incremental Knowledge Engineering
NeOn	Network Ontologies
NMM	Noy and McGuiness Method

OCML	Operational Conceptual Modelling Language
ODE	Ontology Development Environment
ODSCSM	Ontology for Distributed Supply Chain Simulation and Modelling
OFSCO	OWL-Formalisation of Supply Chain Operations
OLSP	Ontology for Logistics Service Provision
OSCM	Ontology for Supply Chain Management
OSCPMA	Ontology for Supply Chain Process Modelling and Analysis
OSPR	Ontological Knowledge Model for Supply Partner Relationships
OTK	On-To-Knowledge
OWL	Web Ontology Language
PSM	Problem-Solving Method
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
RLM	Role-Limiting Methods
SC	Supply Chain
SCO	Supply Chain Ontology
SCOPO	Supply Chain Organisation and Problem Ontology
SCOR Model	Supply Chain Operations Reference Model
SCSO	Supply Chain Simulation Ontology
TOVE	Toronto Virtual Enterprise
UKM	Uschold and King’s Method
UML	Unified Modelling Language
UPON	Unified Process for Ontology Building
VEO	Virtual Enterprise Ontology
W3C	World Wide Web Consortium
WebODE	Web Ontology Development Environment
XML	Extensible Markup Language

1 Introduction

1.1 Motivation

Research in ontologies has attracted increasing attention through the vision of the Semantic Web (Berners-Lee et al. 2001) and a demand for intelligent business applications (Fensel 2004). Ontologies provide knowledge representation and reasoning capabilities for enhancing knowledge sharing and reuse to support (semi-)automated semantic integration, semantic interoperability, knowledge management, and intelligent decision support. As such, ontologies provide their potential and benefits in distributed and heterogeneous business environments, in which knowledge intensity, dynamicity, complexity, and the need for flexibility challenge decision-making (Breslin et al. 2010; Rai et al. 2006; Singh 2003).

The development of ontologies as engineering artefacts is critical for delivering useful ontologies. Developing such ontologies is typically a cumbersome, time-consuming, and error-prone process, which exhibits a structural and logical complexity comparable to the production of large-scale software artefacts (Staab and Studer 2009). Particularly, the activity of *knowledge acquisition* plays a central role as it accounts for the highest impact on the total efforts in ontology development compared to the activities of implementation, evaluation, and documentation (Simperl et al. 2010, pp. 55-56).

Despite the importance of knowledge acquisition, research in ontology development does not sufficiently provide means for reducing the problem of knowledge acquisition through adequately considering the specificity and complexity of domain knowledge (Cardoso 2007; Simperl et al. 2010). For instance, empirical results show that there is a lack of dedicated methods, domain-specific best practices, guidelines and techniques for supporting knowledge acquisition in ontology development. Based on that, Simperl et al. (2010) articulate the need for increased research on more specific, tailored, and substantiated methods for advancing knowledge acquisition (Simperl et al. 2010, pp. 54-56, 60).

Little attention has been paid to the various difficulties of acquiring specific knowledge about a particular domain. That is, the knowledge base does not inform sufficiently about the problem of knowledge acquisition. Especially, there is little known about how to reduce the difficulties of knowledge acquisition. As such, it remains yet an insufficiently solved problem in ontology development, which not only prevents the whole field of ontology engineering fully turning from an art into a mature engineering discipline but also restricts the quality and usefulness of ontologies (Gómez-Pérez et al. 2004).

1.2 Research Approach

This thesis is concerned with *knowledge acquisition in ontology development*. Ontology development deals with the construction of useful ontologies (Gómez-Pérez et al. 2004, p. 5), which represent formal, explicit specifications of a shared conceptualisation of a domain of interest (Studer et al. 1998, pp. 185-187). Within ontology development, knowledge acquisition is a particular activity that deals with the identification (and elicitation) of data, the interpretation of this data (information), and the structuring and interlinking of this information (knowledge). As such, knowledge acquisition presupposes a preceding activity for defining the ontology's scope and purpose as well as a succeeding activity for formalisation and/or implementation.

The *problem of knowledge acquisition* occurs when acquiring specific knowledge about a particular domain of interest. This problem consists of linguistics, cognitive, modelling, and methodical difficulties. Linguistic and cognitive difficulties concern the communication and understanding about the domain of interest, whereas modelling and methodical difficulties pertain to the creation of corresponding knowledge models and the associated activities (d'Aquin et al. 2008, pp. 21-23; Motta 2013; Musen 1993, pp. 406-409; Simperl et al. 2010).

For reducing the problem of knowledge acquisition, this thesis proposes *theory-based knowledge acquisition* for ontology development. Theory-based knowledge acquisition suggests guiding knowledge acquisition by theories used in business economics. These theories capture the specific knowledge with regard to the key constructs, relationships, and dependencies inherent to the particular domain of interest. Such theories might substantiate knowledge acquisition in terms of its theoretical underpinning. That is, theory-based knowledge acquisition makes use of such theories for reducing the linguistic, cognitive, modelling, and methodical difficulties when acquiring specific domain knowledge.

The contribution is a *theory-based knowledge acquisition method for ontology development*. For building and evaluating the knowledge acquisition method, this thesis studies a particular domain of interest, i.e. transport chains, and uses a particular theory of business economics, i.e. Contingency Approach (Kieser and Kubicek 1992). The theory has been frequently used in Information Systems (IS) research (IS Theory 2012) for, among others, design of information systems (Zhu 2002) and organisational knowledge management (Becerra-Fernandez and Sabherwal 2001), as well as assessment of Information Technology (IT) appropriateness (Khazanchi 2005).

1.3 Methodology

This research is conducted by following the *design science paradigm*. In contrast to behavioural science, design science originates in engineering and the sciences of the artificial (Simon 1996). It is a problem-solving paradigm, which builds and evaluates useful IT artefacts to solve identified organisational problems. Design science in IS research consists of two research processes – *build* and *evaluate* – and four design artefacts – *constructs*, *models*, *methods*, and *instantiations* (Hevner et al. 2004; March and Smith 1995; Peffers et al. 2008).

For conducting quality design science, seven guidelines have been proposed (Hevner et al. (2004, pp. 82-90). Table 1 summarises how this thesis adopts these guidelines.

Guideline	Description	Adoption
Design as an Artefact	Produce a viable artefact in terms of a construct, a model, a method, or an instantiation.	This thesis builds a method as the design artefact, i.e. a theory-based knowledge acqui-sition method for ontology development.
Problem Relevance	Develop technology-based so-lutions to important and rele-vant business problems.	Knowledge acquisition is an important task in ontology development and affects the usefulness of the produced ontologies.
Design Evalu-ation	Demonstrating the utility, qua-lity, and efficacy of a design artefact rigorously via well-executed evaluation methods.	This thesis demonstrates the utility of the design artefact through two descriptive evaluation approaches: criteria-based (informed argument) and scenario-based evaluation.
Research Contri-butions	Provide clear and verifiable contributions in the areas of the design artefact, foun-dations, and/or methodologies.	The contribution is the knowledge acquisition method (design artefact) that mitigates the problem of knowledge acquisition and advances research in the area of the design artefact.
Research Rigor	Applying rigorous methods in both the construction and eva-luation of the design artefact.	This research applies principles of deduction, theories in business economics and Artificial Intelligence, method engineering, and evaluation methods.
Design as a Search Process	Utilising available means to reach desired ends while satis-fying laws in the problem en-vironment.	The search included studying the problem of knowledge acquisition in ontology development and analysing the limitations of extant approa-ches as the basis for deducing requirements to build and evaluate the method.
Commu-nication of Research	Presenting effectively both to technology-oriented as well as management-oriented audiences.	The work on this thesis has led to journal articles (Scheuermann and Leukel 2013, 2014), confe-rence (Scheuermann et al. 2013; Scheuermann and Hoxha 2012) and workshop papers (Hoxha et al. 2010; Scheuermann and Obermann 2014), posters (Scheuermann 2011 a, b), as well as technical reports.

Table 1: Adoption of Guidelines for Design Science in IS research

1.4 Thesis Structure

Figure 1 depicts the structure of this thesis by referring to the main components of the conceptual framework of IS research, which are the environment, the knowledge base, as well as the design and evaluation activities (Hevner et al. 2004, pp. 78-61)

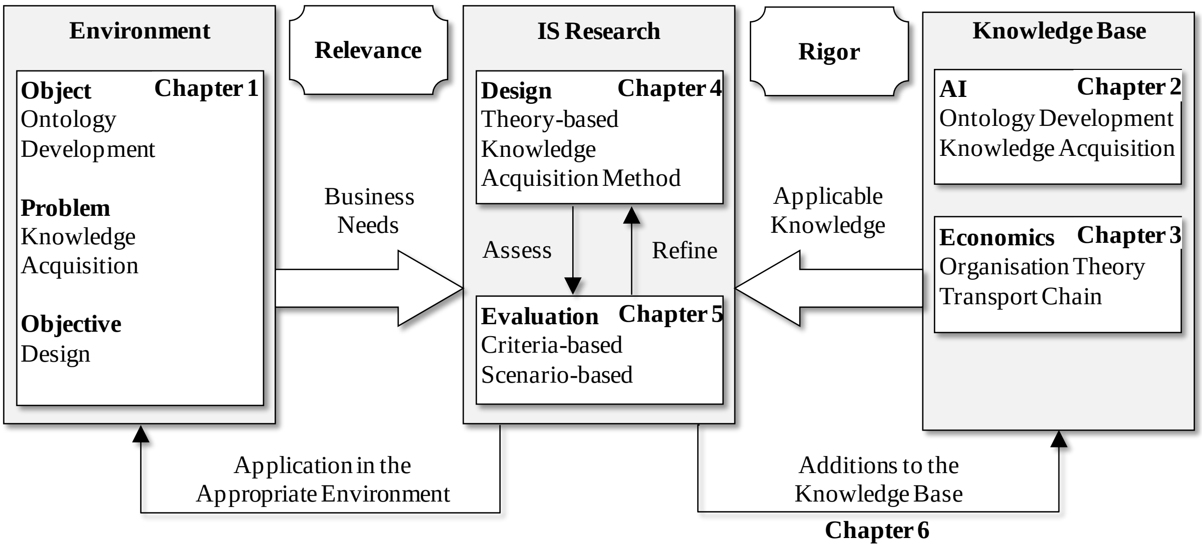

Figure 1: Structure of the Thesis

2 State-of-the-Art

This chapter presents the analysis of the state-of-the-art of knowledge acquisition in ontology development. First, the basic terms and concepts in the areas of ontology, ontology development, and knowledge acquisition are defined. Second, the problem of knowledge acquisition is analyzed and its significance is shown by results of an empirical survey of 15 domain ontologies. Third, existing approaches to knowledge acquisition are reviewed to identify the gap in the extant literature.

2.1 Definitions and Assumptions

2.1.1 Ontology

2.1.1.1 Definition of Ontology

The term ontology originates from philosophy. It represents the branch of philosophy that studies the nature of being or the kinds of existence for organising the things in the world (Brockhaus 2005, p. 4531; Gilchrist 2003, p. 7). The Greek philosophers Socrates and Aristotle developed the foundations of ontology. Socrates provided the concept of abstract classes, the hierarchical relations among them, and class-instance relations. Aristotle added the logical foundation. The result corresponds to a model for describing knowledge about the real world (Smith 2003, pp. 155-156).

At the beginnings of the late 1980s and early 1990s, ontology became a topic in Computer Science and Artificial Intelligence (AI). AI research deals with the formal representation of models of real world phenomena and with the reasoning about these representations. AI research studied ontology specifically in the areas of knowledge engineering (Studer et al. 1998) and knowledge representation (Sowa 2000). In a literal sense, AI "borrowed" the term ontology from philosophy (Gruber 1995, p. 908) and equipped it with a computational meaning. As a result, AI coined the term formal ontology (or computational ontology) (Guarino 1995, Kishore et al. 2004a).

The increasing diffusion and adoption of ontology in AI was accompanied by multiple endeavours to define this computational meaning. An early definition of ontology stems from Gruber (1993, pp. 199-200; 1995, p. 908) who defines ontology as an "explicit specification of a conceptualization".

Borst (1997) takes up the definition of ontology provided by Gruber (1993, 1995) and extends it. Correspondingly, Borst (1997, pp. 11-12) proposes the specification to be formal and the conceptualisation to be shared. Augmenting the definition of ontology by the two characteristics formal and shared reduces possible ambiguous and mismatching interpretations. Thus, it contributes to a more concise and unambiguous understanding of the term ontology.

Moreover, Studer et al. (1998) provide both a concise and comprehensive definition of ontology through not only defining the term ontology but also explaining its key characteristics. In accordance to Studer et al. (1998, p. 185), "ontology is a formal, explicit specification of a shared conceptualisation of a domain of interest". In this context, *conceptualisation* depicts an abstract representation of some (real world) phenomenon by having determined its relevant concepts, relations, and axioms. *Explicit* denotes the symbolic representation and the explicit (not implicit) definition of the types of concepts and relations in addition to the constraints that hold on their use. *Formal* indicates that an ontology should be readable and interpretable by machines; thus, formal excludes the use of natural language. At last, *shared* reflects that an ontology captures consensual knowledge that is not private to an individual person but commonly agreed by a group of individuals.

The common characteristic of these three definitions is their high degree of abstraction. This degree of abstraction allows for covering different types of ontologies independent of a particular ontology language and the associated formalism for knowledge representation.

Due to its general applicability, conciseness, and comprehensiveness, the definition by Studer et al. (1998) establishes the basic understanding of ontology for this thesis. Correspondingly, this thesis defines ontology as follows:

> *An ontology is a formal, explicit specification of a shared conceptualisation of a domain of interest.*

Based on this definition, the Semiotic Triangle (Ogden and Richards 1923, pp. 9-12) allows for characterising the role of ontologies for the formal representation of models of real world phenomena (Figure 2). The starting point of the Semiotic Triangle constitutes a *symbol* (lower left corner), which exhibits a specific form (e.g. word). A symbol has a relationship to an object in the real world, which corresponds to the term *referent* (lower right corner). A direct relationship (dashed line) holds between symbol and referent because everyone can use a symbol to substitute a referent. This means that a symbol stands for a real world object. To complete the Semiotic Triangle, Ogden and Richards (1923) introduce the third element *thought* or *reference* (upper corner). A thought or reference evolves in a human's system of thought. Thus, it corresponds to a mental representation, i.e. to a knowledge element. A thought or reference refers to a *concept*. On the one hand, a symbol literally symbolises a concept, whereas, on the other hand, a concept refers to a real world object. These relation-ships imply that a concept serves as a mediator and (indirectly) relates a symbol to a referent.

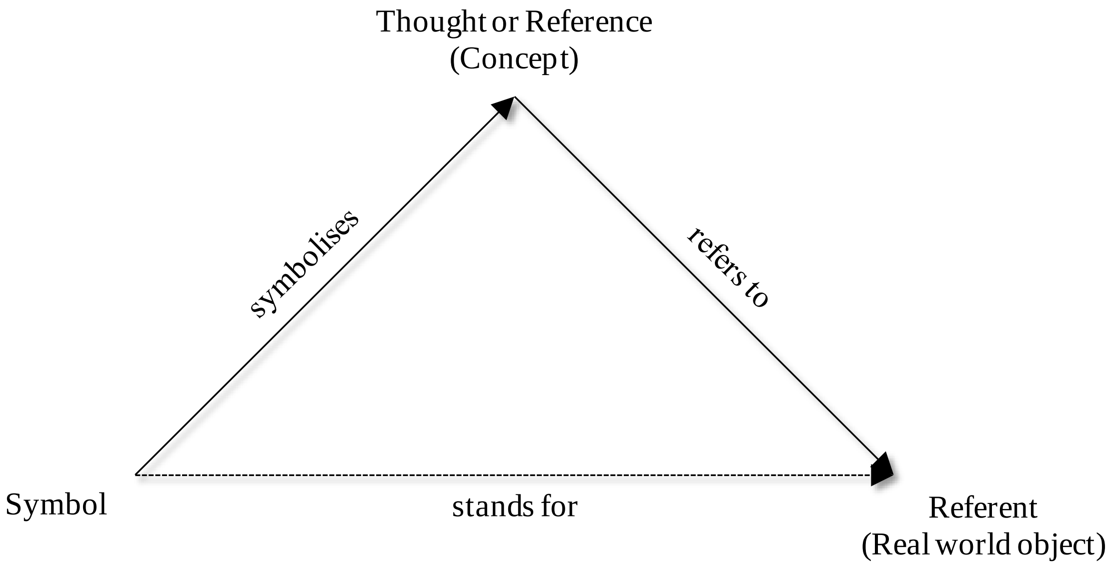

Figure 2: Semiotic Triangle (Ogden and Richards 1923, p. 11)

In accordance to the Semiotic Triangle, an ontology formally and explicitly specifies the concept to restrict the set of possible interpretations of symbols and their corresponding objects of the real world (referents). As a result, ontologies provide consensual and well-defined semantics (meaning) to reduce the number of possible relationships between symbols and referents (Guarino 1995, pp. 632-634; Guarino et al. 2009, pp. 14-16).

Furthermore, a mature body of knowledge discusses the capabilities of ontologies and the associated benefits of their use (Chandrasekaran et al. 1999; Grüninger and Lee 2002; Uschold and Grüninger 1996). Reviewing this body of knowledge and resolving identified ambiguities and redundancies shows that ontologies enable the structuring and interlinking of information (knowledge), the formal and explicit representation of this knowledge, and the provision of a common terminology, i.e. an interlingua. These capabilities enhance know-ledge sharing and reuse as well as reasoning for enabling and supporting semantic integration, semantic interoperability, knowledge management, and intelligent decision support.

2.1.1.2 Components of Ontologies

The formal and explicit representation of consensual knowledge about a particular domain of interest requires a set of modelling primitives. These modelling primitives refer to as the components of an ontology (Gruber 1993; Sharman et al. 2004, pp. 187-190):

– *Classes* are used in a broad sense. They can be either abstract (e.g. intentions, beliefs), concrete (e.g. people, trees), elementary, or composite. Classes correspond to types of anything (e.g. real, fictitious) of which it is possible to make statements about. Classes typically follow a hierarchical organisation, which allows for applying inheritance mechanisms.

– *Relations* define the type of associations between classes of the domain of interest. There are two basic types of relations: unary relations and binary relations.

– *Axioms* depict true statements. Ontologies contain axioms to constrain the knowledge, to verify the correctness of the knowledge in terms of consistency and coherence, as well as to deduce new knowledge.

– *Instances* represent elements of a specific class. Facts depict the relation between these elements. Both instances and facts, i.e. any element of the domain of interest that is not a class refers to as individuals.

Mutual connections between the constituent components of ontologies, the ontology language, and the knowledge representation paradigm influence the use of the respective terms and definitions. For instance, the term concept corresponds to the term class in frame-based languages and to the term unary predicate in First-Order Logic (FOL).

Various knowledge representation paradigms competed to provide ontology languages. An ontology language builds on a knowledge representation paradigm to define the language constructs for the formal specification of ontologies. For instance, Genesereth and Fikes (1992) propose the Knowledge Interchange Format (KIF), which is based on FOL. In contrast, F-Logic (Kifer et al. 1995), Ontolingua (Farquahr et al. 1997), and the Operational Conceptual Modelling Language (OCML) (Motta 1999) build on Frames combined with FOL. LOOM (MacGregor 1991) uses Description Logics (DL) as its underpinning knowledge representation paradigm.

In 2004, the World Wide Web Consortium (W3C) officially recommended the Web Ontology Language (OWL) (McGuiness and van Harmelen 2004) and, in 2009, subsequently recommended OWL 2 as a revision and an extension of OWL 1.1 (Grau et al 2008; W3C OWL Working Group 2012). Both OWL 1.1 and OWL 2 build on DL as their knowledge representation paradigm, i.e. DL establishes the logical foundation of OWL 1.1 and OWL 2.

With respect to the adoption and diffusion of OWL, Cardoso (2007, pp. 85-86) reports on a survey, which involved 627 participants from academia and industrial research. The results show that OWL 1.1 is the most frequently applied ontology language. Thus, OWL 1.1 and presumably OWL 2 refer to as a kind of de facto standard ontology languages.

The previous explanations indicate that both literature (Gruber 1993; Sharman et al. 2004) and particular ontology languages (e.g. OWL 1.1 and OWL 2) use different terms and definitions to depict the constituent components of an ontology (Table 2).

Ontology Components based on Gruber (1993)	**Ontology Components based on OWL 1.1/2**	**Ontology Components based on DL**
Classes	Classes	Concepts
Relations	Properties	Roles
Axioms	Axioms	Axioms
Instances	Individuals	Individuals

Table 2: Different Terms for Ontology Components

To avoid misunderstandings and obscurities, subsequently, this thesis relies on the terms and definitions that OWL 1.1 and OWL 2 provide.

2.1.1.3 Classification of Ontologies

Various classification schemes to categorise different types of ontologies were proposed. These schemes respectively stress and integrate different viewpoints with regard to the purpose, content, and application of ontologies and may have an overlapping coverage.

Mizoguchi et al. (1995, pp. 51-52) study ontologies and specifically task ontologies for reusing problem-solving knowledge. As part of their study, the authors develop a classification that contains four categories: content ontology, communication ontology, indexing ontology, and meta-ontology. First, *content ontology* primarily aims at reusing knowledge and, therefore, contains three sub-categories: domain ontology, task ontology, and general (common) ontology. Second, *communication ontology* is also called tell and ask ontology. This type of ontology is supposed to foster knowledge sharing. Third, *indexing ontology* aims at supporting case retrieval. Fourth, *meta-ontology* incorporates the modelling primitives for representing knowledge in accordance to specific knowledge representation paradigms. However, this classification scheme mixes up various properties and viewpoints of ontologies that concern the content, purpose, and application of ontologies.

Van Heijst et al. (1997b, pp. 191-194) address the use of ontologies with regard to the development of knowledge-based systems. Therefore, the authors propose several approaches to construct and use ontologies for enhancing the process of building knowledge-based systems.

Their classification consists of two orthogonal dimensions: the amount and type of structure as well as the subject of conceptualisation. The first dimension includes a classification that distinguishes between three types of ontologies: *terminological ontology,* which specifies the terms to represent knowledge about the domain of interest (e.g. lexicon), *information ontology*, which specifies the structure of databases (e.g. conceptual database scheme), and *knowledge modelling ontology*, which specifies the conceptualisation about the particular domain of interest. The second dimension differentiates four types: representation ontology, generic ontology, domain ontology, and application ontology. First, r*epresentation ontology* specifies the conceptualisation that underlies a knowledge representation paradigm. Thus, this type of ontology provides the modelling primitives for the other three types of ontologies. Representation ontology corresponds to meta-ontology as proposed by Mizoguchi et al. (1995). Second, g*eneric ontology* represents knowledge that is applicable across several domains. This type of ontology includes classes such as state, event, and process. Generic ontology is comparable to general (common) ontology as introduced by Mizoguchi et al. (1995). Third, *domain ontology* is similar to the definition of domain ontology as presented by Mizoguchi et al. (1995) since it specifies the conceptualisation of a particular domain of interest. As such, it typically specialises a generic ontology. Fourth, similar to Tu et al. (1995), *application ontology* specifies the knowledge that is necessary for a particular application. Despite some similarities between the two previously introduced classification schemes, van Heijst et al. (1997b) separate two main categories of ontologies. This categorisation is comprehensive and detailed but, in contrast to Mizoguchi et al. (1995), it does not capture ontologies for modelling tasks and problem-solving behaviour.

Guarino (1997, pp. 144-145; 1998, pp. 9-10) classifies different types of ontologies in accordance to the level of generality, i.e. the degree of dependence on a particular type of task or on a specific point of view (Figure 3).

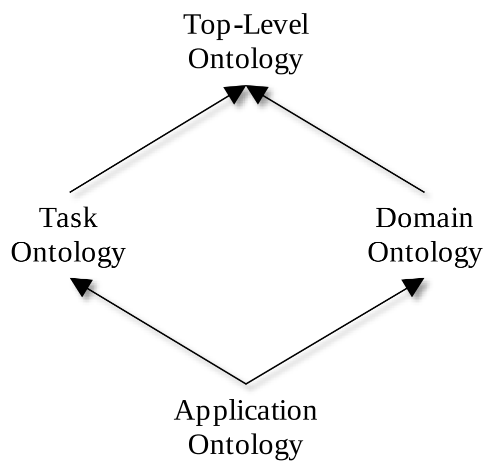

Figure 3: Ontology Classification (Guarino 1997, p. 145)

As shown in Figure 3, Guarino (1997, 1998) distinguishes between four different types of ontologies and, in addition to that, structures these types by means of generalisation-/ specialisation relationships: top-level ontology, task ontology, domain ontology, and application ontology. First, *top-level ontology* specifies a conceptualisation that is independent of a specific domain of interest such as space, time, object, or event. There are several synonymous expressions for the term top-level in use: generic (van Heijst et al. 1997b), general, common (Mizoguchi et al. 1995), upper-level, and foundational. Second, *task ontology* specialises a top-level ontology with respect to a particular type of task (e.g. planning, configuration, scheduling). This type of ontology specifies the task knowledge that is required for solving a particular task type. Third, *domain ontology* specialises a top-level ontology with respect to a particular domain of interest such as healthcare, manufacturing, or transport chains. Thus, this type of ontology specifies the knowledge that is inherent to a particular domain. Fourth, *application ontology* specialises both a task and domain ontology. That is, an application ontology depends on both a particular task and a particular domain for specifying the knowledge of a certain application.

This classification of ontology largely overlaps with the second dimension proposed by van Heijst et al. (1997b) and with the category of content ontology introduced by Mizoguchi et al. (1995). In contrast to Mizoguchi et al. (1995) and van Heijst et al. (1997b), Guarino (1997, 1998) differentiates between various types of ontologies and categorises them consistently according to their level of generality.

In addition to Guarino's (1997, 1998) classification, McGuiness (2003, pp. 173-177) and Uschold and Grüninger (2004, pp. 59-60) introduce a complementary viewpoint, which is denoted as semantic spectrum. The *semantic spectrum* considers the degree of formal semantics, i.e. the richness of the internal structure of ontologies. This spectrum covers categories that range from simple and less expressive to complex and highly expressive types of ontologies (Figure 4).

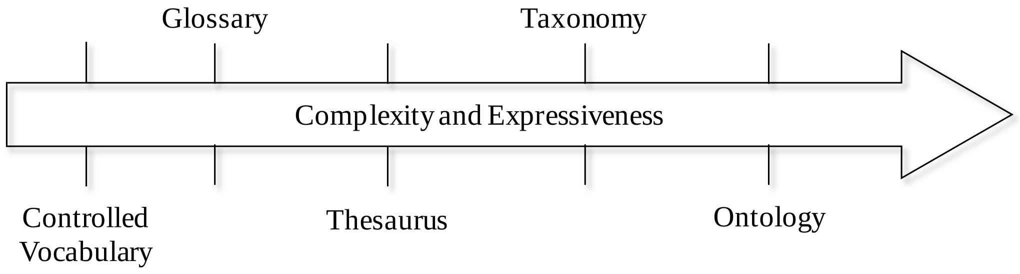

Figure 4: Semantic Spectrum (McGuiness 2003, p. 175)

The main categories of the semantic spectrum are as follows: controlled vocabulary, glossary, thesaurus, taxonomy, and ontology. First, a *controlled vocabulary* provides a finite list of terms (e.g. catalogue). Second, a *glossary* is a list of terms with their meanings specified as natural language statements. Third, a *thesaurus* extends the glossary by incorporating relationships between the terms (e.g. synonym or antonym relationships but no explicit hierarchy). Fourth, a *taxonomy* or formal is-a hierarchy organises the terms hierarchically by generalisation-/specialisation relationships. Fifth, an *ontology* allows for all possible axioms. As indicated above, the semantic spectrum may distinguish between further categories of ontologies that incorporate slightly different or additional characteristics, e.g. based on specific properties of the underlying knowledge representation paradigm.

To categorise different types of ontologies, this thesis relies on both Guarino's (1997, 1998) classification and the semantic spectrum (McGuiness 2003; Uschold and Grüninger 2004) by referring to their level of generality and the richness of their internal structure respectively.

2.1.1.4 Issues Related to Ontologies

The adoption of ontologies may result in some confusion with other existing forms of formal (knowledge) representation. These confusions predominantly occur between ontology and conceptual database schema, Extensible Markup Language (XML) schema, and knowledge base.

Gruber (1993, p. 203) raises questions on similarities and differences between an ontology and a *conceptual database schema*. First, the main difference concerns the respective purpose. An ontology defines the meaning of classes, relations, and properties in a particular domain of interest to represent knowledge, whereas, a conceptual database schema models some data. Second, a conceptual database schema does not necessarily attach explicit and formal semantics to the data. Third, a conceptual database schema usually does not reuse and extend other schemes since these are restricted to a specific and integrated application system. Fourth, developing a conceptual database schema may conventionally be a centralised process, whereas the development of ontologies may be perceived as rather decentralised and collaborative. Fifth, an ontology incorporates a richer internal structure than a conceptual database schema. This difference is due to the typically larger number of modelling primitives in ontology languages. Sixth, modelling a conceptual database schema distinguishes between schema and instances. In the case of ontologies, this could be blurred depending on the use of certain ontology languages and knowledge representation paradigms. Seventh, a conceptual database schema relies on the closed world assumption (i.e. lack of knowledge implies falsity), whereas an ontology takes the stance of the open world assumption (i.e. the lack of knowledge does not imply falsity) (Gruber 1993, p. 203; 2009, pp. 1963-1965; Noy and Klein 2004, pp. 430-432; Uschold and Grüninger 2004, pp. 60-61).

There are three reasons why ontology is different from *XML schema*. At first, an XML schema explicitly defines a specific representational syntax for a certain domain of interest but falls short in specifying the corresponding semantics inherent to this domain. Second, an XML schema specifies the sequence and the hierarchical ordering of elements in a (valid) document instance. Again, such a specification disregards the semantics of the orderings, e.g. it lacks semantics of nested elements. Third, in contrast to ontology, the goal of an XML schema is not to model reusable and context-independent categories of things of the real world. For instance, an XML schema does not aim at modelling whether a data element denoted as logistics service provider refers to the company or the role of being a provider of logistics services within transport chains (Hepp 2007, p. 7; Hitzler et al. 2012, p. 6).

An ontology and a *knowledge base* are sometimes confused with each other. This confusion occurs because the same languages, technical infrastructures, and tools enable the construction of an ontology and knowledge base. Nevertheless, there exists a clear distinction. Ontology provides an explicit terminology and a formal specification that allows for expressing knowledge bases. Referring to this, one of the motivations that leverage the adoption of ontology concerns advanced interoperability between multiple knowledge bases (Guarino and Giaretta 1995, p. 25; Mizoguchi 2003, pp. 375-376; Noy and Klein 2004, p. 428).

To prevent from misunderstandings and obscurities, it is necessary to take into account the respective similarities and differences between the three previously enumerated representational forms and ontology.

2.1.2 Ontology Development

2.1.2.1 Method, Methodology, and Technique

The terms technique, method, and methodology are associated with heterogeneous definitions and, thus, ambiguous understandings within and across several (scientific) disciplines are predominant. Despite this pluralism, studying knowledge acquisition methods in ontology development requires a concise understanding of these terms. Since the term method plays a central role, an inquiry of the term method precedes the terms methodology and technique.

Method originates in the Greek word méthodos, which delineates a systematic process to accomplish a specific goal (Brockhaus 2005, p. 4028). Similar to this definition, Lorenz (1995, pp. 876-879) characterises a method as a planned and systematic process with respect to its means and purpose for solving theoretical or practical tasks. Correspondingly, a method incorporates two characteristics: goal orientation and systematic process (multiple ordered and coherent activities).

From a complementary viewpoint, a method corresponds to a process, which determines how to systematically and based on certain principles (or a combination of these principles) accomplish a specific goal (Stahlknecht and Hasenkamp 2005, p. 212). This definition introduces principles as a third characteristic of method. Such principles refer to fundamental procedure models in terms of general guidelines (e.g. top-down, bottom-up) that serve as templates for activities.

Moreover, Balzert (2000, pp. 36-39, 54-55) defines a method as a systematically applied and substantiated process to accomplish specific goals in the context of defined principles. In compliance with Zelewski (2008), Balzert (2000) explicitly points out that a method should be intersubjective repeatable. Thus, intersubjective repeatability constitutes another characteristic of method.

Based on that, a method includes the characteristics of goal orientation, systematic process, principles, and intersubjective repeatability, which constitute the definition of a method:

> *A method is a systematic and intersubjective repeatable process based on certain principles to accomplish a specific goal.*

For a better understanding, it is necessary to determine a method's constituent components. Therefore, an analysis of various viewpoints on the constituent parts of methods to obtain generally applicable components shows the following result (Gutzwiller 1994, pp. 11-15):

- The *metamodels* (Kühne 2006, pp. 377-382) depict the conceptual foundation, i.e. the language constructs and rules for specifying the method (method metamodel) and its outcomes (outcome metamodel).
- The *activity model* denotes the process (multiple ordered and coherent activities) that transforms inputs into outcomes.
- The *outcomes* present the results from performing the activities of the activity model.
- The *roles* show the competences and responsibilities of the actors involved in performing the activities of the activity model.
- The *techniques* correspond to procedures, instructions, and/or guidelines, which support carrying out the activities of the activity model.
- The *tool* depicts means (e.g. software), which allow for supporting both the activities of the activity model and the use of techniques.

A part of these components forms the description of a method. However, a lack of clarity typically leads to a heterogeneous picture of the adoption of various method descriptions, which aggravates finding consensus on the constituent components of a method. Instead of arguing whether one, two, or certain combinations of multiple components are constituent for a method, it is reasonable to elaborate on the minimal number of constituent components. With regard to this minimal number, there is a broad consensus that a method should at least consist of an activity model (Braun et al. 2005, pp. 1297-1298). Consequently, a method is defined in terms of its constituent components as follows:

> *A method consists of an activity model. Moreover, a method could further contain a description of the metamodels, outcomes, roles, techniques, and tools.*

Such a definition further allows for distinguishing between method and the terms methodology and technique. In the following, this thesis studies the term methodology and, then, examines the term technique.

Methodology is defined as the theory of methods, which are applied in and across different scientific disciplines (Brockhaus 2005, p. 4028). Accordingly, a methodology resides on a superordinate level compared to a method and covers a multiplicity of methods.

In the work on methodologies for building knowledge-based systems, de Hoog (1998, pp. 1.2–1.4) particularly studies the relationship between methodologies and methods. It is argued here that a methodology conforms to "knowledge about methods" (de Hoog 1998, p. 1.2), which depicts that a methodology and a method are not the same. In essence, this definition supports the understanding of a methodology as depicted above.

Moreover, methodology is denoted as “a collection of problem-solving methods governed by a set of principles and a common philosophy for solving targeted problems” (Kettinger et al. 1997, pp. 56-58). As such, a methodology conforms to an accumulation of methods in terms of a conceptualisation on their highest level of abstraction.

In comparison to the definition of method, the term methodology is defined as follows:

> *A methodology represents knowledge about methods and, thus, covers multiple methods. It conforms to an abstract concept that resides on superordinate level in comparison to method.*

Technique is defined as a particular type of proceeding or the execution of an activity (Brockhaus 2005, pp. 6276-6282). This definition indicates that a technique is subordinate to a method and various techniques can be associated to a method with regard to carrying out particular activities.

Further, Gutzwiller (1994, p. 14) points out that a technique describes the way of creating a specific result. Again, this definition highlights the subordinate nature of a technique in comparison to a method.

Similarly, the IEEE (1990, p. 74) defines technique as a “technical or managerial procedure used to achieve a given objective”. It is explicitly added that a method has associated techniques. This understanding of technique is consistent with the two previous definitions.

With reference to the terms method and methodology, subsequently, the definition of the term technique is presented:

> *A technique is a particular type of proceeding (procedure) to create a (pre-)specified outcome of an activity. A technique resides on a subordinate level in comparison to method.*

In this generally applicable understanding, a technique has a means-end relationship to an activity as an integral part of a method’s activity model. Thus, the definition of technique is complementary to the understanding of technique as a constituent component of a method.

2.1.2.2 Definition of Ontology Development Method

Ontology engineering deals with the “activities that concern the ontology development process, the ontology life cycle, and the methods, tools, and languages for building ontologies” (Gómez-

Pérez et al. 2004, p. 5). Since knowledge acquisition in ontology development is the object of research, the particular focus is on the process and the methods for developing ontologies.

Fernandéz-López et al. (1997, pp. 33-34) consider the ontology development process as the set of activities that need to be carried out for building ontologies. Thereto, Fernandéz-López et al. (1997, pp. 33-34) identify and characterise nine activities that constitute the ontology development process: plan, specify, acquire knowledge, conceptualise, formalise, integrate, implement, evaluate, and maintain.

The IEEE standard for Software Engineering (IEEE 1997) motivated Gómez-Pérez et al. (2004a, pp. 109-111) to revise the activities of the ontology development process initially introduced by Fernandéz-López et al. (1997, pp. 33-34). As a result, Gómez-Pérez et al. (2004a, pp. 109-111) propose 18 activities and classify them in three main categories: management, development oriented, and support (Figure 5).

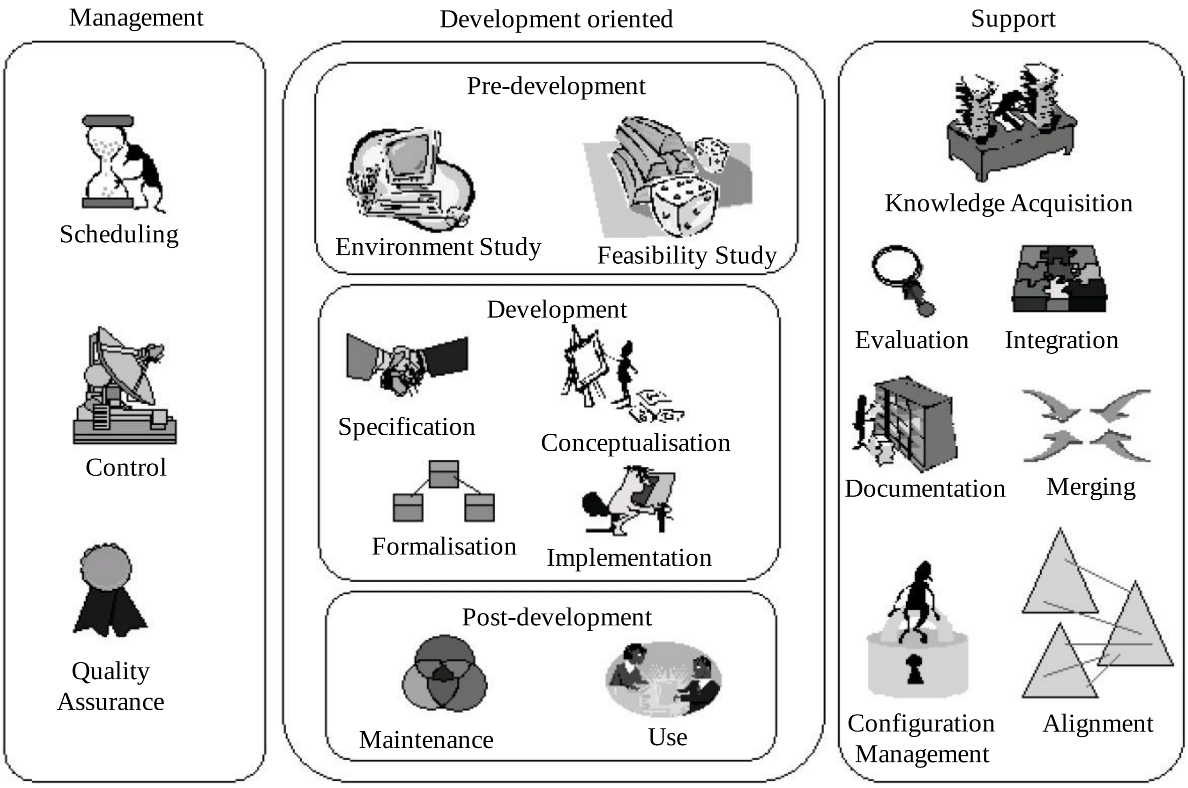

Figure 5: Ontology Development Process (cf. Gómez-Pérez et al. 2004, p. 110)

First, *management* comprises the activities of scheduling, control, and quality assurance for ensuring the effective and efficient development of ontologies. In this context, *scheduling* aims at identifying the activities to be performed, their arrangement, and the definition of the respective time and resources that are needed for their completion. *Control* is supposed to guarantee the completion of the scheduled activities, whereas *quality assurance* is in charge to assure the quality of the results from the development and support activities.

Second, *development oriented* activities cover three sub-categories: pre-development, development, and post-development. *Pre-development* activities contain an *environmental study* to obtain information about the application area and a study to assess the feasibility of ontology development (*feasibility study*). *Development* activities comprise the *specification* of the purpose and scope of the ontology, whereas the *conceptualisation* structures and interlinks the knowledge based on the principle of the Knowledge Level (Newell 1982). *Formalisation* transforms the conceptualisation into a formal model and *implementation* builds a machine-processible model by using an ontology language. *Post-development* includes *maintenance* in terms of updating and refining the ontology as well as the actual (*re-)use* of the ontology.

Third, *support* contains seven activities that proceed in parallel to the development-oriented activities. These support activities are equally essential for ontology development and particularly concern: knowledge acquisition, evaluation, integration, merging, alignment, documentation, and configuration management. The activity of *knowledge acquisition* deals with acquiring knowledge of the particular domain, e.g. from human experts or other knowledge sources (e.g. documents). *Evaluation* incorporates, inter alia, an assessment of the ontology's technical and application-oriented properties. The reuse of existing ontologies to construct a new ontology requires the activity of *integration*. Similarly, *merging* deals with obtaining a new ontology by unifying several ontologies of the same domain of interest, whereas *alignment* establishes different kinds of mappings between the ontologies involved. *Documentation* and *configuration management* record the results of the respective activities within ontology development, e.g. handling different versions of an implemented ontology.

As shown, the ontology development process defines the activities required for ontology development but does not make explicit statements about their order of execution, which the term (ontology development) process might technically indicate. In contrast, an *ontology development method* defines both the exact order of execution and the actual design of the activities of the ontology development process. Therefore, an ontology development method can be defined as follows:

> *An ontology development method is a systematic and intersubjective repeatable process based on certain principles to develop ontologies.*

2.1.2.3 Classification of Ontology Development Methods

The area of ontology development accommodates various types of methods for constructing ontologies. These method types build on specific approaches. The characteristics of these approaches allow for classifying ontology development methods as follows: methods for ontology alignment and merging, methods for re-engineering of existing ontologies, methods for ontology learning, and methods for ontology new development (from scratch) (Corcho et al. 2003, pp. 44-47; Gómez-Pérez et al. 2004, pp. 111-178).

Both *ontology alignment (matching) methods* and *ontology merging methods* generally aim at conflating existing ontologies. Specifically, methods for ontology alignment establish various kinds of mappings between the ontologies and, thus, preserve the original ontologies. In contrast, methods for ontology merging generate a unified ontology from the original ontologies but do not preserve the original ontologies (Euzenat and Shvaiko 2013, pp. 25-55; Shvaiko and Euzenat 2013). That is, the methods for ontology alignment and merging presuppose the a priori acquisition, formalisation, implementation, and provisioning of the relevant knowledge in terms of the availability of useful ontologies.

Ontology re-engineering methods retrieve the conceptualisation of an ontology implementation, transform this conceptualisation according to the given requirements, and implement the (re-engineered) ontology (Gómez-Pérez and Rojas-Amaya 1999; Swartout et al. 1997). As a result, such methods rely on the same prerequisites as methods for ontology alignment and merging in terms of the availability of the corresponding ontologies.

Ontology learning methods aim at enriching or populating existing taxonomies or ontologies with respect to specific ontology components (e.g. classes, properties). Therefore, these methods combine several approaches such as linguistics, statistics, heuristic and pattern matching, machine learning, or data mining for applying them on different types of knowledge sources (e.g. structured, semi-structured, unstructured documents) (Hazman et al. 2011; Shamsfard and Barforoush 2003). Methods for ontology learning typically require the existence and availability of an initial conceptualisation that covers (parts of) the domain of interest. Similarly, these methods rely on fulfilling considerable preconditions (e.g. existing taxonomies or ontologies), which are comparable to the prerequisites for ontology alignment, merging, and re-engineering.

Ontology new development methods are designed for developing ontologies from scratch, i.e. for developing new ontologies. Despite this category of methods explicitly includes ontology reuse (Gómez-Pérez et al. 2004, pp. 111-113), their key characteristic constitutes the important role of knowledge acquisition in response to the adverse effects of the difficulties due to acquiring specific knowledge about a particular domain of interest. This importance points out the shortcomings concerning the usefulness, quality, and availability of existing ontologies, which impede their (re-)use by employing one of the other three types of ontology development methods.

Moreover, Simperl et al. (2010, pp. 49-50) discuss methods for centralised and decentralised (distributed) ontology development. *Centralised ontology development* circumscribes the issue that the ontology developers are at the same geographical location, whereas *decentralised ontology development* in terms of distributed ontology development deals with geographically dispersed ontology developers. In contrast, *collaborative ontology development* addresses issues of building and reaching consensus between the involved actors in ontology development. Consequently, collaborative ontology development primarily concerns methods (e.g. protocols) for reaching an agreement on issues in ontology development such as the inclusion or exclusion of particular classes or properties.

Against this background, the focus is on ontology development methods that pertain to the fourth category while explicitly excluding issues of centralised, decentralised, and collaborative ontology development. Setting this focus is due to the importance of knowledge acquisition inherent to this category of methods and the general (practical) relevance of ontology new development. This relevance becomes evident when considering the empirical finding that 60% of ontologies are newly developed. With regard to reused ontologies, it has been shown that they account for about 95% of the ultimately built ontology (Simperl et al. 2010, pp. 54-56, 60). Presumably, the remaining 5% are required for customisation, which might not require dedicated methods for ontology alignment, merging, and learning.

2.1.3 Knowledge Acquisition in Ontology Development

2.1.3.1 Data, Information, and Knowledge

The term knowledge lacks a commonly agreed and broadly adopted definition. Extant definitions not only discuss knowledge with reference to data and information but also distinguish between different knowledge types. Correspondingly, this thesis defines knowledge with regard to data and information prior to a close inspection of various types of knowledge.

Defining data, information, and knowledge comprises their respective characteristics and points out their mutual relationships (Aamodt and Nygard 1995, pp. 196-201; Alavi and Leidner 2001, p. 109):

- *Data* refers to the arrangement of symbols (e.g. characters, words) from a set of symbols in accordance to syntactical rules (syntax), e.g. 99%.
- *Information* conforms to data that is equipped with meaning. As such, it denotes that information is data in a specific context, i.e. a particular domain of interest such as transport chains, e.g. 99% delivery quality.
- *Knowledge* corresponds to interlinked information. The interlinking of information means to make explicit functional associations between items of information with respect to a specific purpose. For example, 99% delivery quality is related to a higher degree of delivery service (, which might indicate a very low likelihood of customer complaints).

This hierarchical classification of data, information, and knowledge allows for further examining the term knowledge from three predominant but different viewpoints.

The first viewpoint argues about the truth of knowledge in terms of plausible and justified statements based on three characteristics: the availability of knowledge in terms of statements, their justification, and the justifications have to withstand a test, which is acknowledged in its respective area (Heinrich et al. 2004, p. 720; Schreyögg und Geiger 2003, pp. 12-13). Similarly, Nonaka (1994, p. 15) characterises knowledge as justified true beliefs, whereas Talaulicar (2004, p. 1640) highlights the difference between knowledge, belief or faith, and opinion. As such, this viewpoint rather concerns philosophical issues, which are out of scope.

The second viewpoint is different from the previous one since it considers knowledge as an essential prerequisite for successful decision-making, acting, and problem-solving by individuals (von Krogh and Grand 2004, pp. 1648-1656; Probst et al. 2006, p. 22). It introduces several types of potential actions (e.g. planning, classification, diagnosis) and implicitly posits that knowledge is bound to individuals.

The third viewpoint characterises knowledge as networked information that serves a specific purpose (Rehäuser and Krcmar 1996, pp. 5-6; Wittmann 1959, p. 14). This understanding is rather abstract and generic but complies with the above definition and classification of data, information, and knowledge.

Reflecting upon the above definitions in terms of the least common denominator, knowledge can be defined as follows:

Knowledge is interlinked information with regard to a specific purpose.

In addition, the viewpoints on knowledge suggest that there are different types of knowledge. For instance, literature distinguishes between implicit and explicit knowledge (cf. Kuhlen 1995, pp. 34, 38, 42) and implicitly presupposes a distinction between individual and social knowledge (cf. Probst et al. 2006, p. 22).

From an ontology development point of view, literature differentiates between various knowledge types. These knowledge types typically form pairs in the sense of antipodes, e.g. explicit and implicit, individual and social, or declarative and procedural knowledge. Within such pairs, knowledge types mutually exclude each other. Subsequently, this thesis non-exhaustively defines relevant knowledge types (Alavi and Leidner 2001, pp. 110-113; Nonaka 1991, p. 98; 1994, pp. 15-17):

- *Explicit knowledge* refers to articulated and codified knowledge that allows for its sharing and reuse in a symbolic form and/or in (spoken) natural language.
- *Implicit (tacit) knowledge* is rooted in action, experience, and a specific context. It is difficult to articulate and formalise, which both aggravate knowledge sharing and reuse.
- *Individual (personal) knowledge* is created by and inherent to single actions of an individual person.
- *Social (consensual) knowledge* is created by and inherent to collective actions of a group, which at least consists of two individual persons.
- *Declarative (descriptive or propositional) knowledge* represents knowledge about facts. Knowledge about facts is static in nature and describes how things are, i.e. knowledge about things.
- *Procedural (imperative) knowledge* is knowledge about carrying out a specific task. It is dynamic and task-dependent. Procedural knowledge concerns the procedure to perform a task, i.e. how to obtain results by employing declarative knowledge.

For reasons of explanation, knowledge has either an explicit or an implicit form at a specific point in time, whereas knowledge could be explicit, declarative, and consensual at the same point in time. Moreover, it is possible to represent procedural knowledge declaratively, e.g. in terms of task ontologies.

Relating the different knowledge types to ontologies discloses that the types of explicit, consensual, and declarative knowledge are inherent to the definition of ontology. However, further studying knowledge acquisition in ontology development also includes the types of implicit, individual, and procedural knowledge.

2.1.3.2 Definition of Knowledge Acquisition

Knowledge acquisition constitutes an ontology development activity (Gómez-Pérez et al. 2004, pp. 109-111). It is of particular importance in methods for ontology new development, which do not presuppose the availability of existing ontologies. Knowledge acquisition has its origins in the beginning of AI research and, more specifically, in the discipline of knowledge engineering. Therefore, it is necessary to take into account this discipline to establish a thorough definition and understanding of knowledge acquisition in ontology development.

Knowledge Engineering arose from the late 1970's onward as a subdiscipline of AI. In the seminal AI work on themes and case studies in knowledge engineering, Feigenbaum (1977, p. 1017) defines knowledge engineering as the "art of building complex computer programs that represent and reason with knowledge of the world". This definition indicates that knowledge engineering is more generic than ontology engineering with its particular subfield of ontology development. In the early phases of knowledge engineering, the predominant understanding of knowledge acquisition complied with the direct transfer and transformation of human (expert) knowledge to a computer programme. This understanding assumes that the relevant knowledge already exists (typically in the minds of human experts) so that it merely needs to be mined or collected and, then, implemented. Consequently, knowledge acquisition is denoted by the so-called mining view or transfer view. The transfer view implies that knowledge acquisition was considered in a reductionist understanding, which corresponds to the mere (identification and) elicitation of knowledge (Hayes-Roth et al. 1983, p. 23; Schreiber et al. 2002, pp. 15-16).

The knowledge engineering discipline evolved from an art into an engineering discipline (Studer et al. 1998, pp. 161-163) and, thus, knowledge acquisition became subject of a paradigm shift. This paradigm shift changed the role and understanding of knowledge acquisition. This change not only replaced the transfer view on knowledge acquisition but also posited a new constructivist understanding. This constructivist understanding considers knowledge acquisition as a modelling activity, which results in abstract representations, i.e. conceptualisations (Morik 1991, pp. 144-154). The emphasis of *human interpretation*, i.e. the assignment of a particular meaning, as an essential prerequisite for conceptualisations within knowledge acquisition further demonstrated the renunciation of knowledge acquisition in terms of the mere elicitation of knowledge (Kidd 1987, p. 3).

On that basis, the understanding of knowledge acquisition not only consists of identification and elicitation but also, due to its constructivist nature, encompasses interpretation, structuring, and interlinking (Chorafas 1990; Morik 1991; Motta 2013).

The fact that ontology development constitutes a subfield of knowledge engineering allows for defining knowledge acquisition as a constituent activity of ontology development as follows:

> *Knowledge acquisition includes the identification (and elicitation) of data, the interpretation of this data (information), and the structuring and interlinking of this information (knowledge).*

Moreover, it is necessary to classify the activity of knowledge acquisition and clarify the role of methods for knowledge acquisition in ontology development. On the one hand, carrying out the activity of knowledge acquisition requires a preceding activity in terms of the specification of the ontology. Ontology specification concerns the definition of the ontology's purpose and scope. As such, it constitutes an essential prerequisite for knowledge acquisition. On the other hand, knowledge acquisition requires a succeeding activity with regard to formalisation and/or implementation. The activity of formalisation transforms the results of knowledge acquisition into a formal model prior to the activity of implementation, which applies an ontology language on this formal model to build a machine-processible model. Thereby, it is possible to have merely one succeeding activity of implementation, which then implicitly incorporates the activity of formalisation (Figure 6).

Figure 6: Knowledge Acquisition in Ontology Development

The constructivist understanding of knowledge acquisition and its classification within ontology development indicate similarities between the activities of knowledge acquisition and conceptualisation. For reasons of clarification, a distinction between these two activities is due to the understanding of knowledge acquisition that is restricted to the mere elicitation of knowledge (transfer view). That is, restricting knowledge acquisition to the transfer view (reductionist understanding) requires an additional activity of conceptualisation. In contrast, knowledge acquisition based on a constructivist understanding inherently includes the activity of conceptualisation.

With respect to developing ontologies, the definition of knowledge acquisition based on the constructivist understanding is depicted below:

> *In ontology development, knowledge acquisition includes the identification (and elicitation) of data, the interpretation of this data (information), and the structuring*

and interlinking of this information (knowledge). Knowledge acquisition incorporates an activity of conceptualisation and essentially requires both a preceding activity of ontology specification and a succeeding activity of formalisation and/or implementation.

Based on that, a method for knowledge acquisition for ontology development can be defined as follows:

A knowledge acquisition method is a systematic and intersubjective repeatable process based on certain principles to acquire knowledge for ontology development.

When acquiring specific knowledge about a particular domain of interest, ontology development faces the problem of knowledge acquisition.

2.2 Problem Analysis

2.2.1 The Problem of Knowledge Acquisition

The problem of knowledge acquisition is about the difficulties when acquiring specific knowledge about a particular domain of interest. It is relevant for business applications, which use representations of domain knowledge in terms of ontologies to provide capabilities for (semi-)automated semantic integration and interoperability to support knowledge management and decision-making. Particularly, Semantic Web enabled business applications are affected because they use diverse and heterogeneous knowledge sources with different levels of quality (d'Aquin et al. 2008). In addition, the development of ontologies is concerned since knowledge acquisition has the highest impact on the total development efforts in comparison to implementation, evaluation, and documentation. This impact becomes more significant when developing ontologies for specialised domains (e.g. transport chains) (Simperl et al. 2010, pp. 55-56). For instance, the acquisition of specific knowledge about a particular domain such as biomedicine or engineering is difficult due to the specificity (e.g. terminology) and complexity (e.g. number of concepts, relationships, and dependencies) of the respective domain knowledge (Li et al. 2009, pp. 39-40; Payne et al. 2007, pp. 582-585).

The problem of knowledge acquisition is still a major concern in ontology development research (d'Aquin et al. 2008, pp. 21-23; Motta 2013; Simperl et al. 2010). It has been studied from various perspectives, in particular linguistic theories, cognitive theories, knowledge engineering, and ontology engineering, with latter two being interdisciplinary approaches. Each field aimed at better understanding the problem of knowledge acquisition by identifying and explaining its main difficulties (Motta 2013; Musen 1993, pp. 406-409). According to the specific viewpoint of each discipline, the problem of knowledge acquisition in ontology development can be characterised based on four central difficulties (Figure 7).

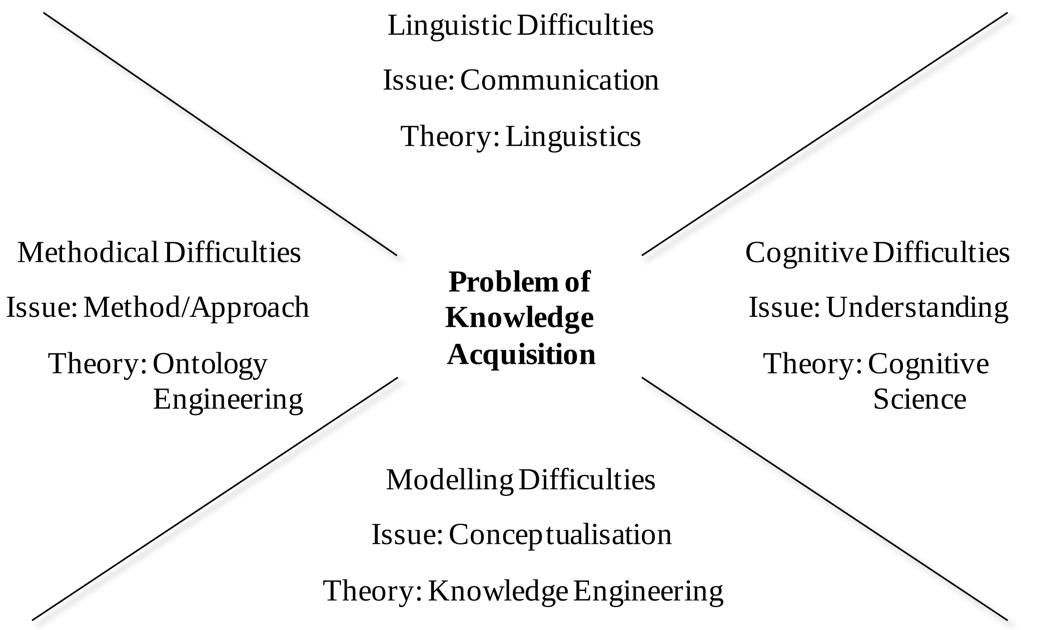

Figure 7: The Problem of Knowledge Acquisition

These difficulties are distinct because they originate from the respective viewpoints of different academic fields.

2.2.1.1 Linguistic Difficulties

Linguistic difficulties concern the communication about the domain of interest. This difficulty typically occurs between the ontology developer and domain expert because they rarely speak the same language. There is a lack of a common understanding about terms and definitions (Musen 1993, pp. 407-409). For instance, similar or the same words with different meanings are used as in the context of transport chains where loading unit represents a charge carrier and unit load the cargo to be charged. However, both the ontology developer and domain expert need to participate jointly in knowledge acquisition for developing useful ontologies.

Ontology developers might not expect domain experts to articulate their specialised knowledge in clear terms, which are easily usable for knowledge acquisition. In contrast, they are supposed to familiarise with the domain of interest and learn the relevant terms and definitions for being able to communicate in a clear and concise manner. With regard to that, psychological studies of comprehension (Dabrowska 2004; Whitaker and Stemmer 1997) indicate that the prior (background) knowledge about a particular domain is central for clearly communicating and providing understandable explanations so that obscurities and potential misunderstandings could be avoided. For example, ontology developers have knowledge in the field of computer science or related areas whereas experts in the domain of transport chains rather have a background in the area of business administration or engineering (Byrd et al. 1992, pp. 119-120; Winograd and Flores 1995).

2.2.1.2 Cognitive Difficulties

Cognitive difficulties are due to the nature of human knowledge. Human knowledge about a particular domain of interest normally relates to solving a specific type of task. The dependence of domain knowledge on solving a task means that large parts of human knowledge refer to the types of implicit and procedural knowledge (Musen 1993, pp. 406-407; Schreiber et al. 1994, pp. 31-34; Van Heijst et al. 1997a, pp. 311-313). This kind of knowledge is difficult to articulate and, thus, to conceptualise as domain experts generally have limited capabilities for reliable introspection (Payne 2007, pp. 583-584; Schreiber et al. 2002, pp. 190-191). For instance, knowledge about designing transport chains is rooted in the actions and experiences of the domain expert and it is bound to the respective context of transport chain design. In addition, the corresponding knowledge is closely linked to the task and procedures of how to design transport chains.

Moreover, cognitive difficulties arise from the complexity of a particular domain with regard to the number of constructs, relationships, and dependencies as well as the amount of knowledge that is initially necessary for understanding the domain. As a result, the ontology developer requires high efforts and subjective judgement to capture and understand the domain of interest for acquiring the relevant knowledge (Karbach and Linster 1990, pp. 2-3; Li 2009, p. 40). For instance, ontology developers will likely perceive the domain of pizza as less complex than transport chains.

2.2.1.3 Modelling Difficulties

Modelling difficulties result from the key characteristics inherent to models: representation, abstraction, and pragmatism. Correspondingly, a model is an image of some real world phenomena, which could be natural or artificial (representation). It captures knowledge only about the relevant characteristics (abstraction) and it is designed in terms of a substitute for serving a specific purpose (pragmatism) (Stachowiak 1973, pp. 131-133). These characteristics are also constituent for knowledge acquisition based on the constructivist understanding. This means that modelling in terms of constructing abstract and pragmatic representations, i.e. conceptualisations is not supposed to be limited to the mere transfer of knowledge (e.g. making implicit knowledge explicit) but rather suggested to be creative, approximate, and iterative (Musen 1993, pp. 410-411; Studer et al. 1998, p. 163). As a result, modelling accounts for difficulties with regard to the identification and elicitation of (data), the interpretation of this data (information), and the structuring and interlinking of this information (knowledge) (Karbach and Linster 1990, pp. 2-3; Chorafas 1990; Morik 1991). For instance, when a domain expert reports on 99% delivery quality in the context of logistics service provisioning, the ontology developer needs to identify and elicit the relevant data, i.e. '99%' and 'delivery quality'. Further, this data needs human interpretation by capturing the meaning of '99%', which represents a high degree of fulfilment, and of 'delivery quality', which corresponds to a key performance indicator. At least, this information requires interlinking with other information so that it could be associated to the likelihood of customer complaints.

Furthermore, modelling also relates to the use of ontology languages and formalisms for representing conceptualisations, which capture the constructs, relationships, and dependencies of the domain. Correspondingly, modelling difficulties depend on the characteristics of the ontology language and the knowledge representation formalism (e.g. expressiveness and computational decidability) but also on the ability and skills of the ontology developer in using them (Byrd et al. 1992, pp. 119-120; Musen 1993, pp. 409-410). For instance, logistics service provisioning involves temporal information such as pick-up date and delivery date. Depending on different ontology languages and formalisms, such temporal information could be modelled differently but equally contain different semantics (Scheuermann et al. 2013).

2.2.1.4 Methodical Difficulties

Methodical difficulties reflect the empirical finding that "knowledge acquisition support leaves room for improvement with respect to the level of detail of methods, availability of domain-specific best practices, guidelines [...] and the usage of techniques [...] (Simperl et al. 2010, p. 60).

Methodical difficulties result from the generic character of available methods for knowledge acquisition and the application of associated techniques. According to the definition of method (Section 2.1.2.1), ontology developers benefit from comprehensive method descriptions that allow for understanding and tailoring the method to the specific requirements of domain knowledge acquisition instead of following ad-hoc approaches. Such method descriptions include the application of techniques, which support the creation of outcomes by performing the knowledge acquisition activities. For instance, depending on the size of the domain to be modelled and having a comprehensive method description, ontology developers could appropriately adapt the available method. This adaptation might include merging or separating activities, defining new types of outcomes, involving additional roles, and integrating suitable techniques to acquire thoroughly the specific domain knowledge.

Because the knowledge acquisition methods described in the literature have been purposely designed for general applicability, i.e., independent from particular domains, these methods are less capable to integrate domain specific best practices and guidelines. With regard to acquiring knowledge about complex and specific domains, it could be useful to a priori enrich available methods with insights and established practices of the respective domain to ensure a targeted knowledge acquisition. For instance, for the acquisition of knowledge about the domain of transport chains, the ontology developer could use the Supply Chain Operations Reference (SCOR) Model (APICS Supply Chain Council 2010) as a domain-specific best practice to further detail the activities and select suitable techniques.

That is, available methodical approaches for knowledge acquisition are too generic and inadequately consider the specificity and complexity of the domain of interest. There is a need for providing more systematic and tailored knowledge acquisition methods (Li 2009, p. 39-40; Simperl et al. 2010, pp. 55-60).

2.2.2 Survey of Example Domain Ontologies

For providing empirical evidence for the problem of knowledge acquisition, a survey is conducted to study how existing ontologies in the domain of transport chains have been developed.

2.2.2.1 Rationale and Review Strategy

This review covers transport chain ontologies that make an original contribution to their respective field. It studies how and to which extent the development of transport chain ontologies has considered the problem of knowledge acquisition in terms of its particular difficulties. In other words, which means have been taken to reduce the linguistic, cognitive, modelling, and methodical difficulties to develop the respective transport chain ontologies. Further, this review includes the application of methods for ontology development because knowledge acquisition is typically embedded in an overarching ontology development method. In addition, the review takes into account if and how the proposed ontologies have been evaluated; this criterion is used as a proxy for ontology quality (assuming that knowledge acquisition affects ontology quality).

The survey comprises ontologies that represent knowledge about the domain of transport chains. Since the term transport chain is afflicted with ambiguities and used heterogeneously, ontology search not only considers this term but also “supply chain”, “supply network”, “value chain”, “virtual enterprise”, “networked enterprise”, and “extended enterprise”, “supply chain management”, “logistics”, and “logistics management”. It additionally focuses on domain ontologies that generally address transport chains without subscribing to any specific industry sector. This coverage is because transport is a cross-sectional function; thus, it occurs in wide array of industry sectors (Christopher 2005, pp. 2-6; Mentzer et al. 2001, pp. 3-5; Pfohl 2010, pp. 297-298). This framework excludes ontologies that can be attributed solely to the manufacturing domain such as the ontologies provided by Lin and Harding (2007) and Lin et al. (2004) or product design and development, e.g. Vegetti et al. (2011). Similarly, the term “ontology” has different underlying definitions in literature so that related terms, i.e. “data model”, “information model”, “meta model”, “reference model”, “semantic model”, “knowledge model”, “ontology model”, “domain model”, and “domain ontology” need to be considered as well. However, the review excludes conceptual models as proposed by Lu et al. (2010), Madni et al. (2001) or Xu et al. (2011) that are no ontology as defined by the semantic spectrum (Section 2.1.1.3)

Ontologies for the transport chain domain have been developed in various fields of research, and thus have been published in various outlets serving these fields. In an iterative way, the list of search results was reduced by adding constraints such as source type (journal and proceeding) as well as expanded by adding alternative terms as depicted above. This procedure led to a shorter list, which was then manually inspected by analysing the abstracts and skimming the content. The search yielded a total of 15 transport chain ontologies. These ontologies represent major contributions of ontology development in the domain of transport chains from the year 2000 onwards. Table 3 presents them according to their chronological order of development.

Transport Chain Ontology	Acronym	Author(s) and Year
Virtual Enterprise Ontology	VEO	Soares et al. 2000
Ontology for Supply Chain Management	OSCM	Ahmad et al. 2003
Logistics Ontology for Production Logistics and Hospital Logistics	LOPLHL	Wendt et al. 2003
Mass Customisation Ontology	MCO	Pawlaszczyk et al. 2004
Supply Chain Simulation Ontology	SCSO	Fayez et al. 2005
Supply Chain Organisation and Problem Ontology	SCOPO	Chandra and Tumayan 2007
Formal Approach Toward a Unified View of the Supply Chain	SCOntology	Gonnet et al. 2007
Logistics Ontology	LO	Leukel and Kirn 2008
Supply Chain Ontology	SCO	Ye et al. 2008
Ontological Knowledge Model for Supply Partner Relationships	OSPR	Chi 2010
Ontology for Supply Chain Process Modelling and Analysis	OSCPMA	Grubic et al. 2011
OWL-Formalisation of Supply Chain Operations	OFSCO	Zdravkovic et al. 2011
Ontology for City Logistics	GenCLOn	Anand et al. 2012
Ontology for Distributed Supply Chain Simulation and Modelling	ODSCSM	Lin et al. 2012
Ontology for Logistics Service Provision	OLSP	Scheuermann and Hoxha 2012

Table 3: Identified Transport Chain Ontologies

2.2.2.2 Ontology Descriptions

To provide a basis for empirically studying the problem of knowledge acquisition, each of the identified transport chain ontologies is briefly described. This description contains background information about ontology development as well as the purpose, scope, basic structure, key concepts, and example applications for each of the identified ontologies.

The Virtual Enterprise Ontology (VEO) was developed in the context of a trans-European project, which involves several academic institutions and industrial companies from the microelectronics industry. The project dealt with the phases of requirements analysis and system specification of an order promise module in a decision support system to foster the task of production planning and control in virtual enterprises. The purpose of the VEO is to enhance human communication with regard to requirements identification, requirements specification, and system design. The scope of the VEO covers manufacturing supply chains in the semiconductor industry. The VEO uses natural language statements to define its components, exploits object models as a means for visualisation, and covers the following three main sections: Networked/Extended Organisations, Plans and Planning, as well as Management of Orders (Soares et al. 2000).

The Ontology for Supply Chain Management (OSCM) was introduced by Ahmad et al. (2003). The purpose of the OSCM is to facilitate knowledge sharing and communication among the participants of supply chains. Since the OSCM is supposed to represent a general-purpose ontology for supply chain management, its scope covers supply chains independent of any specific industry sector. This general-purpose character necessitates the extension and refinement of the OSCM with respect to specific application scenarios. These scenarios primarily consider the areas of forecasting, aggregate planning, and supply chain decision making. The OSCM comprises four main sections: Supply Chain Stages (e.g. Manufacturer, Supplier, Customer), Supply Chain Functions (e.g. Operations, Distribution, Customer Service), Supply Chain Strategies (e.g. Response Time, Product Variation, Service Level), Supply Chain Performance (e.g. Inventory, Transportation, and Facilities) (Ahmad et al. 2003).

The Logistics Ontology for Production Logistics and Hospital Logistics (LOPLHL) was proposed in the context of the DFG-SPP 1083 Intelligent Software Agents and Business Application Scenarios project. The purpose of the LOPLHL is to establish a common basis for developing domain ontologies in the areas of both production logistics and hospital logistics to enable efficient communication processes. Consequently, the scope of the ontology covers production logistics and hospital logistics. The LOPLHL primarily specialises the Enterprise Ontology (EO) (Uschold et al. 1998), i.e. its classes Supply, Person, Resource, Capability, and Activity. Based on that, the Production Logistics Ontology contains classes such as Partner, Resource, QUA-Entity, and Activity, whereas the Hospital Logistics Ontology comprises classes such as Diet, Patient, Material, Role, and Act (Wendt et al. 2003).

The Mass Customisation Ontology (MCO) was suggested in the context of the research project EwoMacs, which was funded by the German Federal Ministry for Education and Research. The purpose of the MCO is to optimise inter-organisational and distributed cooperation. Its scope covers supply chains in the area of mass customisation. The MCO serves as an integral part of an agent-based simulation framework based on an example of mass customisation in the shoe industry. It reuses the EO and provides a generic and a specific part. The generic part refers to a kind of top-layer. This top-layer contains classes and properties that are supposed to be valid for nearly all conceivable mass customisation applications. The specific part complements the top layer and corresponds to a kind of bottom-layer. As such, it encompasses classes and properties that address specific mass customisation settings at an operational level (Pawlaszczyk et al. 2004).

The Supply Chain Simulation Ontology (SCSO) was proposed by Fayez et al. (2005). Its purpose is to integrate various supply chain views and models to capture the knowledge for constructing distributed simulation models of dynamic, information intensive, geographically dispersed, and heterogeneous supply chain environments. Thus, the scope of the SCSO comprises supply chains independent of any specific industry sector. The SCSO is organised in three layers and consists of three corresponding ontologies: a core ontology, a middle ontology, and a dynamic ontology. The dynamic ontology extends and constrains the core and middle ontology for circumscribing specific supply chains and their environments. In total, the SCSO contains 16 main classes, which, apart from their enumeration, neither have a definition nor provide any further explanations (Fayez et al. 2005).

The Supply Chain Organisation and Problem Ontology (SCOPO) was introduced as part of a framework for designing a knowledge-based information support system. This knowledge-based system aims at facilitating the handling of organisational dynamics, operational uncertainty, and process integration in supply chains. Against this background, the purpose of the SCOPO is to document shared knowledge about issues and problems in supply chains for enhancing information and process integration. The scope covers supply chains in the context of multi-staged steel manufacturing processes. Moreover, Chandra and Tumayan (2007) report on three exemplary applications of the SCOPO: as an explicit medium that allows knowledge workers to share their skills, as a specification for software engineers within the development of complex applications, and as a support for decision makers to understand decision-making in a multi-staged steel manufacturing process (Chandra and Tumayan 2007).

SCOntology was presented as a formal approach to provide a unified and integrated view, i.e. a global view on supply chains. SCOntology constitutes a framework to describe supply chains formally at various levels of abstraction. This description allows for the specification of information logistics processes, different metrics, and performance-related concepts to evaluate supply chains. The purpose of SCOntology is to provide an interlingua for the stakeholders that participate in supply chains for supporting the understanding of the multiplicity of supply chain interrelationships. The scope of SCOntology covers supply chains independent of any specific industry sector. SCOntology focuses on both process- and performance-related concepts (Gonnet et al. 2007).

The Logistics Ontology (LO) was developed in the context of the Business Objective Driven Reliable and Intelligent Grids for Business project. The purpose of the LO is to facilitate customers and suppliers in specifying and, thus, formally representing logistics systems for their particular application in information systems. The scope of the LO covers logistics systems independent of any specific industry sector. The LO is supposed to contribute to a research framework that focuses on logistics systems under customisation, i.e. logistics systems that deal with individual customer requirements. This framework exploits, inter alia, domain ontologies to make logistics massively customisable (Leukel and Kirn 2008).

The Supply Chain Ontology (SCO) was suggested by Ye et al. (2008). The purpose of the SCO is to provide an interlingua for enabling the semantic integration of heterogeneous application systems across supply chains. The scope of SCO concerns web-based enterprises, virtual enterprises, and supply chains independent of any specific industry sector. In contrast to closed supply chain systems, web-based or virtual enterprises incorporate supply chain partnerships that are dynamically and last only for a short time. Against this background, the SCO constitutes the backbone of a data integration framework (Ye et al. 2008). Therefore, SCO not only reuses the EO but also includes parts of the SCOR Model (APICS Supply Chain Council 2010).

The Ontological Knowledge Model of Supply Partner Relationships (OSPR) was presented for monitoring partners across supply networks. The purpose of the OSPR is to describe both supply partners and the relationships between them to enable partner tracing and finding by means of inferring implicit relationships among supply network participants. The scope of the OSPR covers supply networks independent of any specific industry sector. The ontological knowledge model for supply partner relationships consists of 16 classes, 18 object properties, and 20 datatype properties (Chi 2010).

The Ontology Model for Supporting Supply Chain Process Modelling and Analysis (OSCPMA) was proposed within the context of a larger research project. This project aims at the development of a business process model to represent dyadic or buyer-supplier relation-ships. The purpose of the OSCPMA is to enable and support supply chain process modelling and analysis. Its scope covers supply chain processes with regard to material flows and information flows in buyer-supplier relationships independent of any specific industry sector. The OSCPMA primarily relies on two abstract classes, i.e. GeneralView and SupplyChain-View. These abstract classes serve as an umbrella and anchor for the remaining 60 classes of the OSCPMA (Grubic et al. 2011).

The OWL-Formalisation of Supply Chain Operations (OFSCO) was introduced by Zdravkovic et al. (2011). The purpose of the OFSCO is to overcome semantic inconsistencies and incompleteness's of the SCOR Model. Based on that, the OFSCO contributes to a semantic infrastructure to improve the interoperability between information systems and to enable effective knowledge management in supply chains. Its scope covers supply chains independent of any specific industry sector. The OFSCO semantically enriches the SCOR Model with regard to different levels of expressiveness. Therefore, it provides three ontologies: SCOR-KOS OWL ontology (the term KOS stands for knowledge organisation system), SCOR-Cfg OWL ontology (the term Cfg stands for configuration), and the SCOR-FULL OWL ontology (Zdravkovic et al. 2011).

The Ontology for City Logistics (GenCLOn) was developed by Anand et al. (2012). The purpose of GenCLOn is to formalise the knowledge about the domain of city logistics for providing an interlingua. As such, this ontology fosters interoperability between models and their reusability for automated categorisation, query answering, as well as modelling and simulation. The ontology's scope encompasses city logistics and, in particular, urban freight transport. The GenCLOn covers a macro and a micro part. The macro part addresses social, political, and environmental issues whereas the micro part concerns supply-demand patterns between private actors (Anand et al. 2012).

The Ontology for Distributed Supply Chain Simulation and Modelling (ODSCSM) was proposed as part of an ontology-based framework. This framework fosters the annotation of supply chain process models for enabling their interoperability and reusability as well as for facilitating their modelling and simulation implementation. The purpose of the ODSCSM is to provide a standardised terminology, i.e. an interlingua of supply chains to describe supply chain process models based on formal semantics. Its scope covers supply chains independent of a specific industry sector. The ODSCSM relies on the SCOR Model and distinguishes between four main classes: SupplyChainProcess, SupplyChainProcessType, SupplyChain-Strategy, and SupplyChainEnterprise (Lin et al. 2012).

The Ontology for Logistics Service Provision (OLSP) was introduced as part of a larger approach, which combines Semantic Technology and Service-oriented Computing to enable the intelligent and flexible provision of logistics services in supply chains and customised logistics applications. The purpose of the OLSP is to capture, structure, and formalise the knowledge of the logistics domain for creating semantic descriptions of logistics services. Its scope covers supply chains independent of any specific industry sector. The OLSP has a modular organisation and consists of eight modules, which respectively represent separate logistics ontologies. These ontologies capture knowledge about logistics services, logistics processes, logistics objects, logistics actors, logistics roles, logistics locations, logistics resources, and logistics key performance indicators. In addition, the OLSP imports particular logistics ontologies that cover specific logistics areas such as hazardous cargo or airport codes (Scheuermann and Hoxha 2012).

2.2.3 Results from Example Domain Ontologies

The presentation of the survey results is divided into four parts. The first part deals with the usage of ontology development methods and ontology evaluation, whereas the remaining three parts focus on the problem of knowledge acquisition. Here, linguistic and cognitive difficulties are considered together, whereas modelling and methodical difficulties are treated separately.

2.2.3.1 Preliminary Remarks

Table 4 summarises the survey results with regard to the use of ontology development methods and ontology evaluation.

Transport Chain Ontology	Ontology Development Method	Ontology Evaluation
VEO	Uschold and King (1995)	Scenario
OSCM	Custom	Not reported
LOPLHL	Not reported	Not reported
MCO	Custom	Not reported
SCSO	Custom	Not reported
SCOPO	Custom	Scenario
SCOntology	Grüninger and Fox (1995)	Case study
LO	Ontologising	Two use cases
SCO	Uschold and King (1995)	Prototype system
OSPR	Custom	Case study
OSCPMA	Noy and McGuiness (2001)	Three case studies
OFSCO	Ontologising	Argumentation-based, scenario
GenCLOn	Custom	Data-driven, two case studies
ODSCSM	Not reported	Case study
OLSP	Uschold and Grüninger (1996)	Class room experiment, scenario

Table 4: Ontology Development and Evaluation in Transport Chain Ontologies

The results show that all 15 publications report on ontology new development, which highlights the significance of this type of ontology development and allows for distinguishing between the following four categories. The first category covers transport chain ontologies (VEO, SCOntology, SCO, OSCPMA, and OLSP) that rely on extant ontology development methods, e.g. proposed by Uschold and King (1995), Grüninger and Fox (1995), Noy and McGuiness (2001), and Uschold and Grüninger (1996). The second category represents transport chain ontologies (OSCM, MCO, SCSO, SCOPO, OSPR, and GenCLOn) that adopt custom ontology development methods. The term custom depicts that extant ontology development methods motivated this kind of methods. Insights and experiences gathered from prior ontology development projects further exert an influence on them. The third category comprises transport chain ontologies (LO and OFSCO) that result from applying an ontology language (e.g. OWL) on an existing informal or semi-formal body of knowledge (e.g. SCOR Model). In contrast, the fourth and last category encompasses transport chain ontologies (LOPLHL and ODSCSM) that do not explicitly report on the use of any method in general and, particularly, any ontology development methods.

Furthermore, the majority of the transport chain ontologies reports on ontology evaluation (VEO, SCOPO, SCOntology, LO, SCO, OSPR, OSCPMA, OFSCO, GenCLOn, ODSCSM, and OLSP). Evaluation covers a wide array of methods, which primarily centre on descriptive approaches in terms of use cases, case studies, and scenarios. The remaining transport chain ontologies (OSCM, LOPLHL, MCO, and SCSO) do not address ontology evaluation as the prerequisite for ensuring the quality and usefulness of the developed ontologies. In this context, merely the SCO provides a paper-based serialisation in OWL DL, whereas all the other transport chain ontologies fall short in providing machine-readable specifications.

These survey results demonstrate that about one third of the transport chain ontologies utilise established ontology development methods, whereas the remaining two thirds rely on custom methods or completely lack a methodical basis. The partial lack of ontology evaluation in combination with dominant usages of descriptive approaches as a rather weak form of evaluation might indicate a low degree of the perceived usefulness of the respective transport chain ontologies.

2.2.3.2 Results for Linguistic and Cognitive Difficulties

The survey results demonstrate that the development of the majority of transport chain ontologies is affected by linguistic and cognitive difficulties in knowledge acquisition. That is, five ontologies (VEO, OSCM, LOPLHL, SCOPO, and OSPR) fall short in reducing linguistic and cognitive difficulties for enhancing the communication about and the understanding of the respective domains of interest. The remaining ten ontologies (MCO, SCOntology, LO, SCO, OSCPMA, OFSCO, GenCLOn, ODSCSM, and OLSP) partly consider these two difficulties by means of using dedicated literature (e.g. mass customisation, logistics, city logistics, supply chain management) or standards (e.g. SCOR, Global Supply Chain Forum (GSCF)). However, the mere use of literature or standards as reported insufficiently supports familiarising with the respective domain and, thus, enhancing communication and understanding. Consequently, the rationale of decisions and underlying assumptions of knowledge acquisition that would provide explanations for the content and structure of the developed ontology remains unclear.

For demonstrating these insufficiencies, this thesis subsequently provides some selected examples. Knowledge acquisition within the development of the MCO lacks explanations for selecting and specialising particular classes from the domain of mass customisation (Pawlaszczyk et al. 2004). Despite Fayez et al. (2005, p. 2368) characterise the SCOR Model (APICS Supply Chain Council 2010) as “the only shared and broadly accepted [...] knowledge within the supply chain community”, it remains unclear how the content and structure of the SCSO matches the SCOR Model. Similar to that, Anand et al. (2012, pp. 11946-11952) rely on literature about the domain of city logistics. Despite, it remains unclear how this literature relates to the resultant ontology.

Correspondingly, it is unclear how the mere use of literature and standards for the development of the transport chain ontologies reduces linguistic and cognitive difficulties in knowledge acquisition.

2.2.3.3 Results for Modelling Difficulties

The development of transport chain ontologies insufficiently considers modelling difficulties in knowledge acquisition. The development of seven transport chain ontologies (LOPLHL, MCO, LO, OFSCO, GenCLOn, ODSCSM, and OLSP) disregards modelling difficulties when acquiring the specific knowledge about the particular domain. Three transport chain ontologies (SCSO, SCOPO, and SCOntology) only deal with identification and elicitation while four ontologies (VEO, OSCM, SCO, and OSPR) additionally cover the structuring of domain knowledge. Solely, the development of the OSCPMA deals with modelling difficulties in knowledge acquisition by considering the identification, structuring and interlinking when acquiring the relevant domain knowledge.

These results indicate that modelling in terms of constructing conceptualisations suffers from shortcomings, which could be illustrated by the following examples. The development of the LOPLHL (Wendt et al. 2003) and MCO (Pawlaszczyk et al. (2004) concerns integrating and specialising knowledge, whereas there are no explanations for the reasons underlying modelling.

When developing the SCO, Ye et al. (2008, p. 5) only substantiate the conceptualisation of the class *Supply Chain Structure*. Furthermore, the definition of classes and properties of the OSCPMA during the fifth step of the ontology development method remains vague. This is because Grubic et al. (2011, pp. 853-854) merely report that these classes and properties rely on domain knowledge and experience.

As a result, the development of the transport chain ontologies falls short in reducing modelling difficulties, i.e. the creation of conceptualisations capturing the specific domain knowledge.

2.2.3.4 Results for Methodical Difficulties

The survey results disclose methodical difficulties in knowledge acquisition within the development of the transport chain ontologies. The development of four ontologies (LOPLHL, MCO, LO, and ODSCSM) lacks any of the constituent components of a knowledge acquisition method as defined in Section 2.1.3.2. Five ontologies (VEO, SCSO, SCOntology, SCO, and OLSP) refer to one activity of knowledge acquisition, whereas three ontologies (OSPR, OFSCO, and GenCLOn) report on the use of principles for knowledge acquisition. The development of the remaining four ontologies (OSCM, SCOPO, OSPR, and OSCPMA) comprises between two and four knowledge acquisition activities, which are combined with principles.

These methodical insufficiencies become even more evident when reflecting upon the following examples. When developing the OFSCO, Zdravkovic et al. (2011, pp. 406-407) merely report on the principles of induction, inspiration, and synthesis as introduced by Holsapple and Joshi (2002, pp. 43-45) without giving any explanations about their application. Furthermore, Leukel and Kirn (2008, pp. 97-98) as well as Lin et al. (2013, pp. 228-232) apply an ontology language (OWL) on selected parts of an existing body of knowledge instead of adopting an extant ontology development method, which would allow for acquiring knowledge on a methodical basis.

With regard to the constituent components of methods for knowledge acquisition, the development of transport chain ontologies predominantly suffers from a too generic methodical basis, which aggravates dealing with the specificity and complexity of the domain of interest.

2.2.4 Interim Summary and Implications

Table 5 summarises the results from reviewing the development of existing transport chain ontologies with regard to the particular difficulties of knowledge acquisition.

	Linguistic Difficulties	Cognitive Difficulties	Modelling Difficulties	Methodical Difficulties
VEO	Not considered		Identification, structuring	1 activity
OSCM	Not considered		Identification, structuring	4 activities
LOPLHL	Not considered		Not considered	Not considered
MCO	Partly considered by literature (mass customisation)			Not considered
SCSO	Partly considered by literature (supply chain management)		Identification	1 activity
SCOPO	Not considered		Identification	3 activities
SCOntology	Partly considered by standard (SCOR)		Identification	1 activity
LO	Partly considered by standard (SCOR)		Not considered	Not considered
SCO	Partly considered by standard (SCOR)		Identification, structuring	1 activity
OSPR	Not considered		Identification, structuring	2 activities, induction
OSCPMA	Partly considered by standard (SCOR, GSCF)		Identification, structuring, interlinking	4 activities
OFSCO	Partly considered by standard (SCOR)		Not considered	Induction, Inspiration, synthesis
GenCLOn	Partly considered by literature (city logistics)		Not considered	Middle-out
ODSCSM	Partly considered by standard (SCOR)		Not considered	Not considered
OLSP	Partly considered by literature (logistics, supply chain management)			1 activity

Table 5: Problem of Knowledge Acquisition in Transport Chain Ontology Development

The results provide empirical evidence of the problem of knowledge acquisition when developing transport chain ontologies. Subsequently, it is demonstrated that the problem of knowledge acquisition is not restricted to ontologies of that particular domain but is equally valid for ontology development independent of any specific domain.

Cardoso (2007) carried out an empirical study to provide an account of the adoption and application of constructs, models, methods, and tools for ontology development in the area of the Semantic Web. This study includes 627 participants from academic and industrial research. It covers nine main areas with close ties to ontology development and ontology (re-) use. Particularly with regard to the adoption of ontology development methods, the empirical study shows that 60 percent of the participants do not use any method for constructing ontologies, whereas the remaining 40 percent distribute over more than ten different development methods (Figure 8).

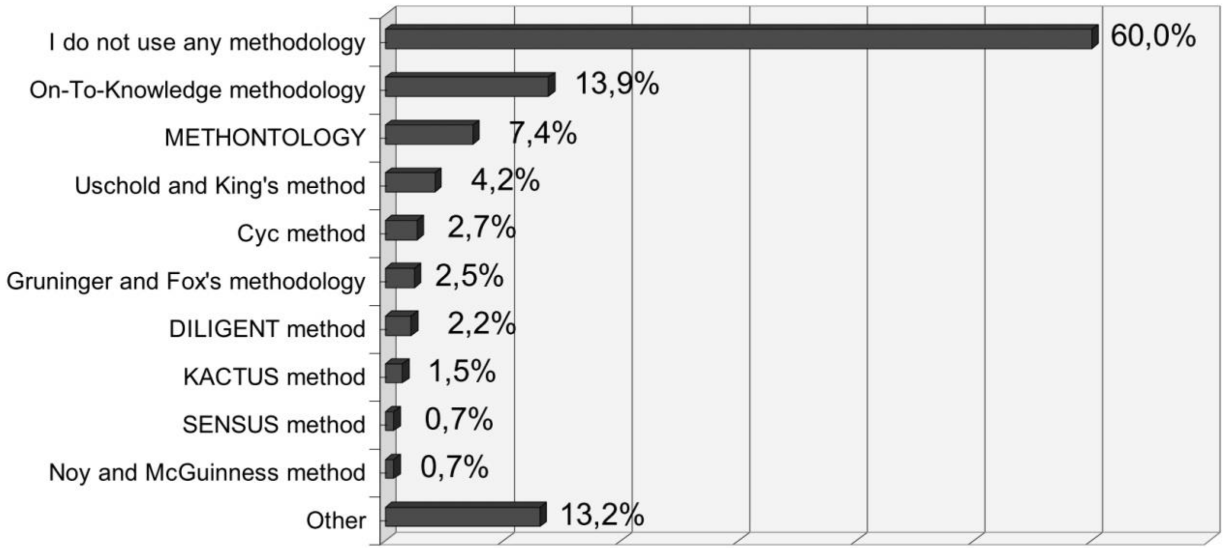

Figure 8: Adoption of Ontology Development Methods (Cardoso 2007, p. 87)

These findings suggest that if the construction of ontologies only pays little attention to the adoption of existing ontology development methods there could be an even lower interest in knowledge acquisition methods providing capabilities for reducing its particular difficulties.

With regard to that, Simperl et al. (2010) conducted a survey on current practices in ontology development with a particular focus on knowledge acquisition. This survey covers 148 ontology development projects from academia and industry. The results demonstrate that knowledge acquisition is of major importance for ontology development. This is because it accounts for the highest impact on the total ontology development efforts compared to the activities of implementation, evaluation, and documentation (Simperl et al. 2010, pp. 55-56). Moreover, there is further empirical evidence that "knowledge acquisition support leaves room for improvement with respect to the level of detail of methods, availability of domain-specific best practices, guidelines, ..." (Simperl et al. 2010, p. 60). For developing ontologies, exiting knowledge acquisition approaches are too generic and do not adequately take into account the specificity and complexity of the respective domain of interest. These shortcomings result in an articulated need for more specific and tailored knowledge acquisition methods (Simperl et al. 2010, pp. 55-56).

Both Cardoso (2007) and Simperl et al. (2010) provide empirical evidence for the problem of knowledge acquisition in ontology development being not restricted to transport chain ontologies. Based on that, existing methods for reducing the problem of knowledge acquisition in ontology development are considered.

2.3 Existing Knowledge Acquisition Methods

For defining the research gap, existing methods for knowledge acquisition in ontology development are reviewed.

2.3.1 Rationale and Review Strategy

This review covers knowledge acquisition methods that make an original contribution to their respective field. It studies to what extent these methods are able to reduce the problem of knowledge acquisition. The focus is on capabilities proposed by these methods to reduce the linguistic, cognitive, modelling, and methodical difficulties.

The analysis relies upon a unified graphical representations of each method. These representations use the Entity-Relationship Model (ERM) and its original notation (Chen 1976) and have been built from the individual representations that were found in the original sources. The ERM consists of entity types, relationship types, attribute types, and cardinality types for representing phenomena on the type level:

1. *Entity Types* correspond to categories of (real-world) objects, which could be abstract (e.g. intentions, beliefs), concrete (e.g. people, trees), elementary, or composite. In contrast, entities represent instances of entity types, i.e. an entity types could of one or more entity types.

2. *Relationship types* represent categories that describe different kinds of relationships between entity types. These relationship types can be abstract of concrete.

3. *Attributes* could be attached to both entity types and relationship types for providing additional information.

4. *Cardinality types* are applied on relationship types to constrain the number of entities participating in the respective relationship. Based on Chen (1976), cardinalities could be one or many.

According to the original Chen-Notation (Chen 1976), the syntactic format is associated to the following graphical notation (Table 6).

Language Construct	Notation	
Entity Type	Rectangle	Entity Type Name
Relationship Type	Rhombus (Diamond)	Relationship Type Name
Attributes	Ellipse	Attribute Name
Cardinality Type	Multiplicities	N 1

Table 6: Basic ERM Language Constructs and Notation Elements

Based on the ERM, a template is applied, which reflects the constituent components of a method (cf. Section 2.1.2.1) (Figure 9).

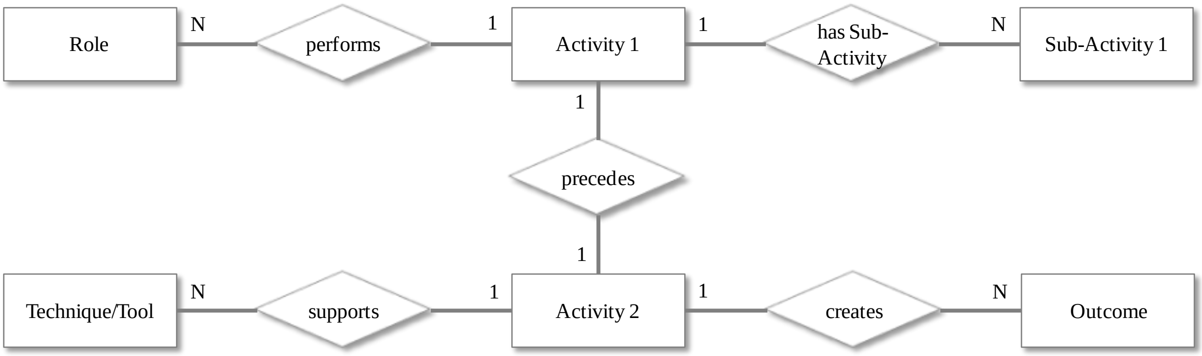

Figure 9: ERM-based Template for Representing Knowledge Acquisition Methods

When using this template, the original terms and definitions of the knowledge acquisition methods are retained as far as the analysis is not constrained.

While focusing on ontology new development, the review excludes methods for ontology alignment, ontology merging, ontology re-engineering, and ontology learning. Corresponding methods for these tasks as proposed by KACTUS (modelling Knowledge About Complex Technical systems for multiple USe) (Bernaras et al. 1996), SENSUS (Swartout et al. 1997), DOGMA (Developing Ontology-Grounded Methods and Applications) (Jarrar and Meersman 2002, 2008), and NeOn (Network Ontologies) (Suarez-Figuero 2010) must be excluded. Similarly, methods for the development of controlled vocabularies, glossaries, thesauri, and taxonomies (e.g. Nickerson et al. 2012) are not covered because these are no ontology as defined by the semantic spectrum (Section 2.1.1.3).

Methods for knowledge acquisition primarily originate from the fields of knowledge engineering and ontology engineering. An initial list of search results was reduced by adding constraints, e.g.

source type (journal and proceeding) as well as expanded by adding alternative terms as depicted above. This procedure led to a much shorter list, which was then manually inspected. The search yielded a total of nine ontology development methods, which represent the major contributions to the discipline of ontology engineering from the year 1990 onwards. Table 7 lists these methods according to their chronological order of publication.

Ontology Development Method	Acronym	Authors and Year
Cyc Method	CYC	Lenat and Guha 1990
Grüninger and Fox’s Method	GFM	Grüninger and Fox 1995
Uschold and King’s Method	UKM	Uschold and King 1995
METHONTOLOGY	MET	Fernandéz-López et al. 1997
Noy and McGuiness Method	NMM	Noy and McGuiness 2001
On-To-Knowledge	OTK	Staab et al. 2001
A Helix-Spindle Model for Ontology Development	HSM	Kishore et al. 2004b
Unified Process for Ontology Building	UPON	De Nicola et al. 2009
A Generic Ontology Development Framework	GODF	Rajpathak and Chougule 2011

Table 7: Identified Ontology Development Methods

In addition, the search identified three methods from the knowledge engineering literature (Table 8).

Knowledge Engineering Method	Acronym	Authors and Year
Knowledge Acquisition and Documentation Structuring	CommonKADS	Schreiber et al. 1994 (2002)
Protégé-II – Knowledge Engineering Environment	Protégé-II	Eriksson et al. 1995
Model-based and Incremental Knowledge Engineering	MIKE	Angele et al. 1998

Table 8: Identified Knowledge Engineering Methods

2.3.2 Results

2.3.2.1 The Cyc Method

The Cyc Method (CYC) for ontology development originates in the Cyc project (Lenat and Guha 1990), which started in the middle of the 1980s. The Cyc project aimed at capturing common sense knowledge on a large scale, i.e. it started from a basis of one million hand-entered statements. The Cyc project considered different so-called micro-theories that characterise knowledge of different domains of interest from various viewpoints. However, the application of CYC primarily refers to the Cyc project. Figure 10 provides an overview of CYC.

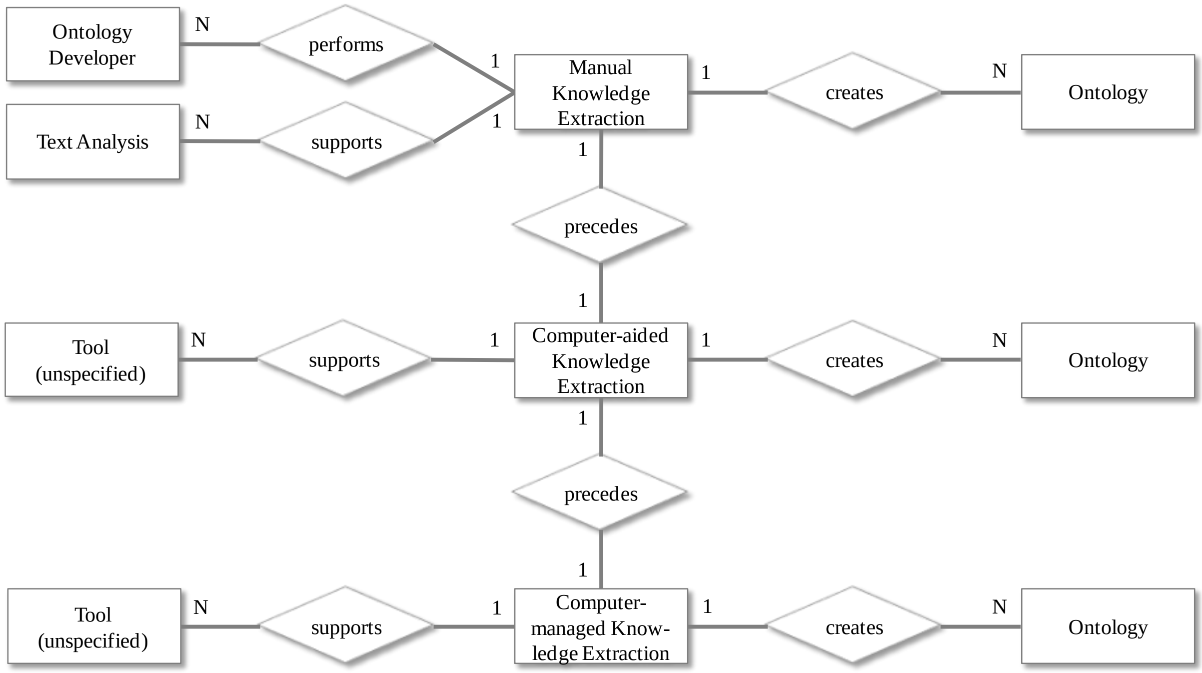

Figure 10: Method Metamodel of Cyc

The first activity concerns the *manual extraction of common sense knowledge* from various types of textual knowledge sources (e.g. books, newspapers). This activity includes three sub-activities: searching and representing common sense knowledge that underpins knowledge sources, i.e. the knowledge necessary to understand books or newspapers, examining the rationale behind curiosities or implausibilities in knowledge sources, and identifying questions about the extracted common sense knowledge (Figure 11).

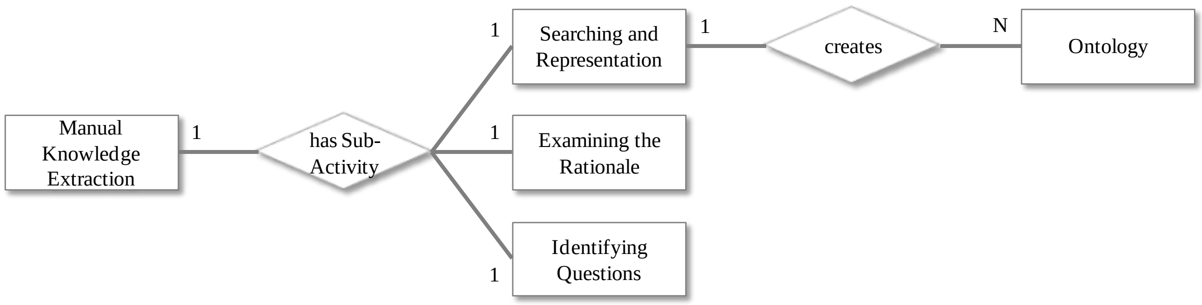

Figure 11: Method Metamodel of Manual Knowledge Extraction in CYC

The second activity is defined as *computer-supported extraction of common sense knowledge*, i.e. the (semi-)automatic extraction of common sense knowledge from various types of textual knowledge sources. Thereby, the computer-supported extraction of new common sense knowledge draws upon knowledge that the first activity already acquired.

The third activity is *computer-managed extraction of common sense knowledge*, i.e. the automatic extraction of common sense knowledge from various types of textual knowledge sources. In comparison to the previous activities, the third activity aims at the highest degree of automation.

CYC contains two cross-sectional activities, which extend over the previous three activities: (1) developing a top-level ontology that contains the most abstract classes and properties as well as (2) capturing the knowledge of different domains of interest to construct the micro-theories (domain ontologies).

It can be stated that the Cyc Method deals with knowledge acquisition based on a constructivist understanding. CYC does not pay attention to resolving linguistic and cognitive problems. It proposes three activities that primarily concern the identification and interpretation. Two activities superficially concern aspects of structuring and interlinking. Knowledge acquisition in CYC is based on five activities that produce the ontology (CYC knowledge base). The role of the ontology developer relies on the text analysis technique by following a bottom-up approach as well as some (unspecified) tool support. CYC lacks explicit information about a method and outcome metamodel.

2.3.2.2 Grüninger and Fox's Method

Grüninger and Fox (1995) proposed an ontology development method based on the experiences during the TOVE (Toronto Virtual Enterprise) project and the associated development of the TOVE Ontologies (Fadel et al. 1994; Fox and Grüninger 1998; Fox 1992; Fox et al. 1993, 1996, 1998; Grüninger and Pinto 1995; Kim and Fox 1995; Kim et al. 1999; Lin et al. 1996; Tham et al. 1994; TOVE 2002). Due to its origins, Grüninger and Fox's (1995) Method (GFM) is also known as the TOVE Method. Thereby, not only the TOVE project but also the adoption of First-Order Logic for developing knowledge-based systems influenced the design of GFM. The method contains six subsequent activities: motivating scenarios, informal competency questions, formal terminology, formal competency questions, formal axioms, and completeness theorems (Figure 12).

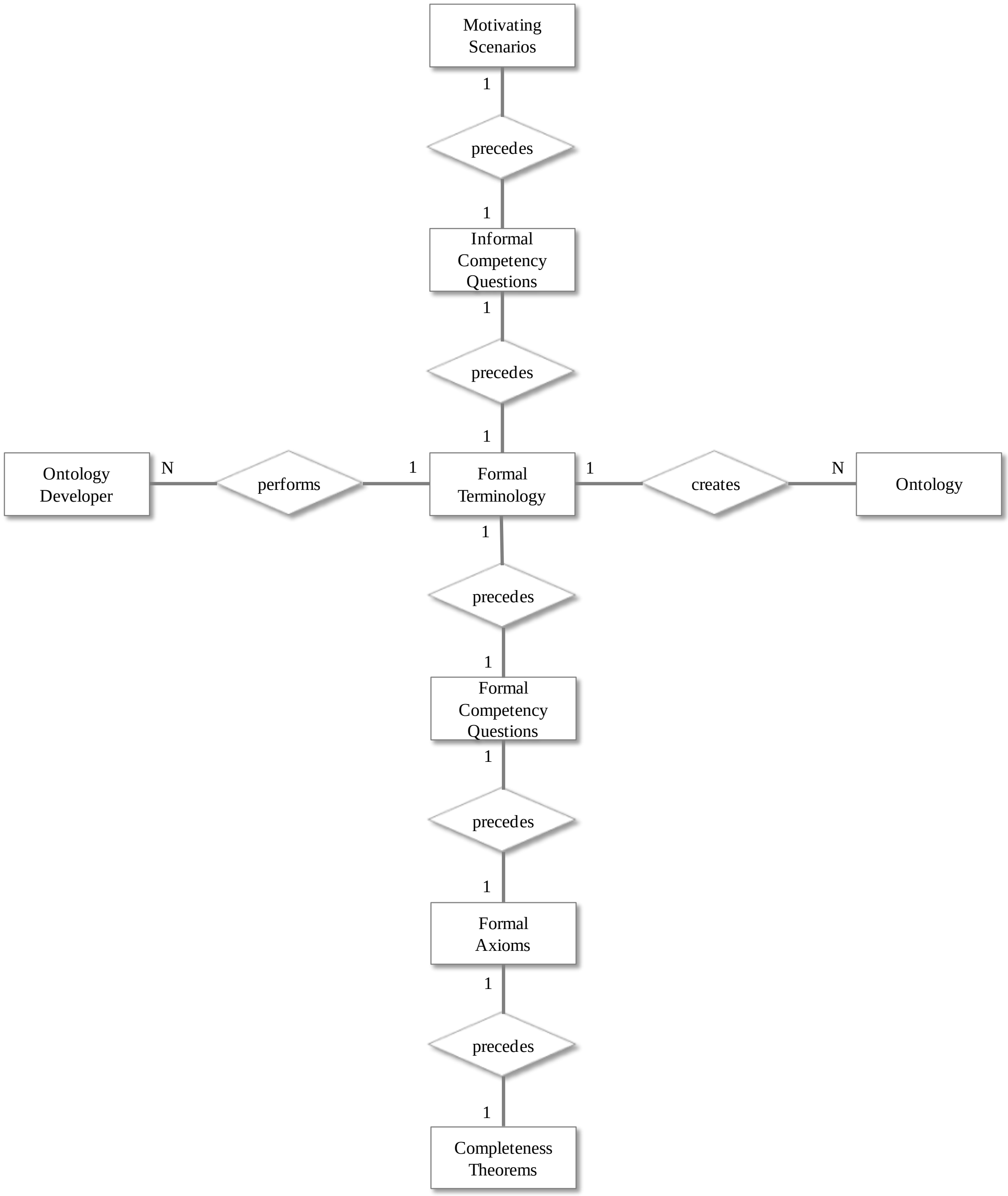

Figure 12: Method Metamodel of GFM

The activity *motivating scenarios* copes with intuitively identifying potential applications. Motivating scenarios rely on stories or examples to circumscribe the target application and to provide reasons for developing the ontology (purpose). Each scenario elaborates on the requirements for ontology development and points to possible solution pathways for the problems addressed. The scenarios provide initial ideas of the intended semantics of the classes and properties of the envisioned ontology.

The second activity produces *informal competency questions*, which determine the scope of the ontology. Informal competency questions correspond to natural language questions that the envisioned ontology should answer. Such competency questions represent some kind of informal requirements specification, which also allows for evaluating the ontology. Grüninger and Fox (1995) recommend stratifying the competency questions by constructing both simple and complex questions.

The activity *formal terminology* uses FOL as the knowledge representation paradigm. Based on the informal competency questions and the respective answers, the ontology developer formalises the relevant classes and properties. The developer needs to be familiar with FOL (subsequently terms of FOL are used) and to identify objects and predicates (e.g. unary predicates, binary predicates).

The activity *formal competency questions* uses FOL for rewriting the informal competency questions to ensure consistency with the outcome of the preceding activity.

The fifth activity defines *formal axioms* that specify the definitions of the terms and the constraints holding on their use by means of FOL. The resulting formal axioms must satisfy the formal competency questions and characterise their answers.

The last activity *completeness theorems* defines the conditions under which the answers of the formal competency questions are complete.

For knowledge acquisition, Grüninger and Fox's Method contains one specific activity *formal terminology*, which adopts the transfer view. This activity concerns the identification and interpretation prior to directly implementing the knowledge in KIF. The preceding activities of *motivating scenarios* and *competency questions* could be used to mitigate linguistic and cognitive problems but lack details and procedures for knowledge acquisition. To construct the GFM's outcome in terms of an ontology, knowledge acquisition is centred on the role of the ontology developer and governed by intuition as the basic principle. Due to the method's high level of abstraction, it does not supply metamodels and is independent of specific techniques and tools for supporting knowledge acquisition.

2.3.2.3 Uschold and King's Method

Uschold and King (1995) proposed an ontology development method that originates from experiences made during the Enterprise Project (Enterprise Project 1997) and within the development of the Enterprise Ontology. Uschold and King's Method (1995) (UKM) is also known as the Enterprise Method. It represents a major outcome of Uschold and Grüninger's

(1996) seminal work on principles, methods, and applications of ontologies. The method contains four activities: identify purpose, building, evaluation, and documentation (Figure 13).

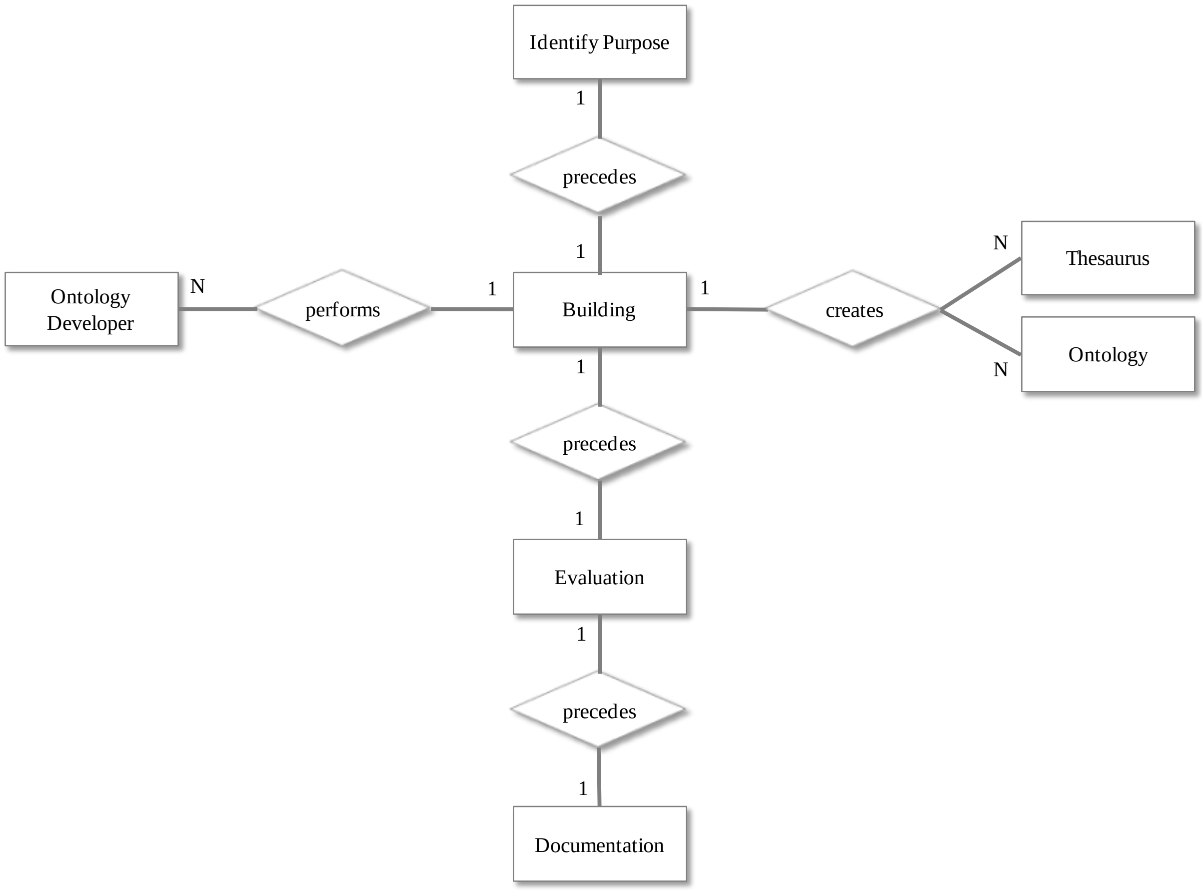

Figure 13: Method Metamodel of UKM

The first activity is defined as *identifying the purpose and the scope* of the ontology. This activity covers the clarification of the reasons for ontology development (purpose), the definition of the intended application area of the ontology (scope), and the localisation and restriction of the domain of interest.

The second activity *building* has three sub-activities: capture, coding, and integrating. *Capture* is the acquisition of relevant knowledge with respect to the ontology's purpose and scope. The proposed procedure includes identifying key classes and properties in the domain of interest, creating unambiguous natural language definitions for these classes and properties, and identifying the terms that refer to such classes and properties. Uschold and Grüninger (1996) point out three different approaches for capture: The *top-down approach* starts with the most abstract classes and properties. Then, it specialises them into more specific classes and properties. Uschold and Grüninger (1996) argue that this approach allows for a better control of the level of detail but also requires higher effort because of not needed and arbitrary high-level classes. In contrast, the *bottom-up approach* starts with the most specific classes and properties. Then, it generalises them into more abstract classes and properties. This approach achieves a high level of detail but additionally increases the overall effort, aggravates the detection of common characteristics between classes and properties, and increases the risk of inconsistencies. The *middle-out approach* starts in the middle, i.e. it identifies core classes and properties. Then, it generalises and specialises them as needed. It thus maintains a balance between the two former approaches. *Coding* concerns the implementation of the ontology in a formal language. *Integrating* focuses on the reuse of existing ontologies to complement the constructed ontology. It is possible to carry out this sub-activity in parallel to capture and coding (Figure 14).

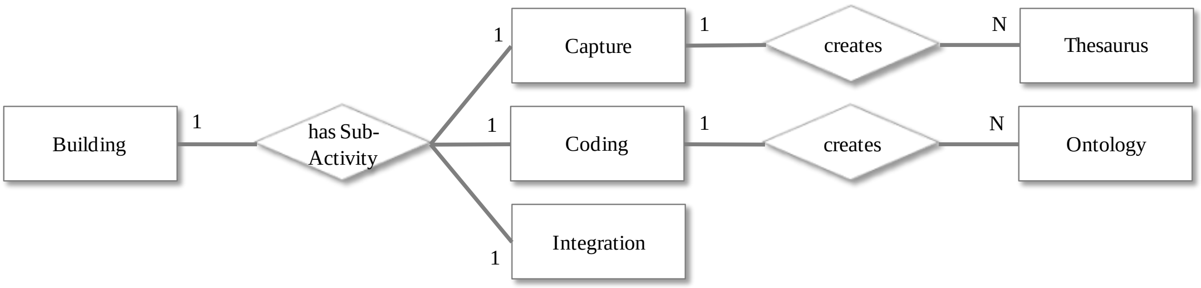

Figure 14: Method Metamodel of Building in UKM

The third activity is *evaluation*. This assessment concerns the design and content of the ontology, its target application, and its documentation with regard to a frame of reference, e.g. a requirements specification and a set of competency questions.

The fourth activity is *documentation* by defining guidelines and naming conventions.

The analysis shows that UKM, i.e. the particular sub-activity capture, belongs to knowledge acquisition based on a constructivist understanding. While this sub-activity consists of three tasks for identifying and interpreting the relevant knowledge, it lacks further details about the knowledge acquisition procedures besides indicating an application of brainstorming and motivating scenarios. Additionally, the method proposes top-down, bottom-up, and middle-out as the principles and requires the role of the ontology developer. However, the method's description provides neither techniques and metamodels nor details about the adoption of specific tools.

2.3.2.4 METHONTOLOGY

Firstly introduced by Gómez-Pérez et al. (1996), Fernandéz-López et al. (1997) proposed the ontology development method METHONTOLOGY (MET). The design of this method was influenced by the IEEE software development process (IEEE 1997) and knowledge engineering methods (e.g. Waterman 1986). MET served as the methodical foundation for developing a chemical ontology (Fernandéz-López et al. 1999), an environmental pollutants ontology (Gómez-Pérez and Rojas-Amaya 1999), and the so-called reference ontology (Arpírez et al. 1998).

METHONTOLOGY is a rather comprehensive ontology development method since it distinguishes between different activities that not only cover ontology development but also support and management. For the purpose of this review, the following activities are relevant: specification, conceptualisation, formalisation, implementation, maintenance, and, most notably, the activity of knowledge acquisition (Figure 15).

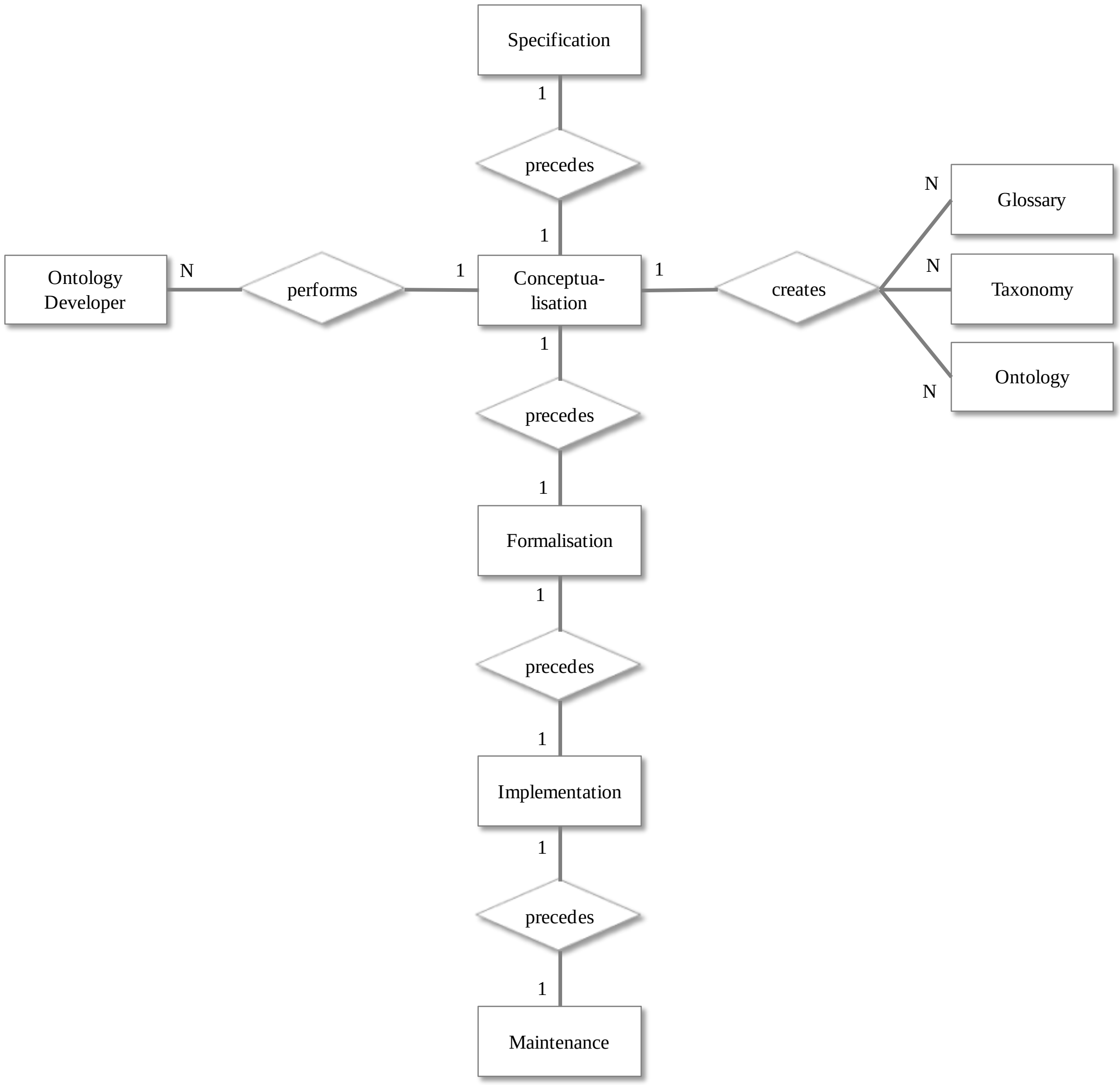

Figure 15: Method Metamodel of METHONTOLOGY

Specification defines the reason for ontology development, the intended application area, the required degree of formality, and the prospective end-users. This activity also defines the purpose and scope of the ontology.

Conceptualisation produces an abstract representation of the domain of interest. Conceptualisation literally corresponds to assembling and completing a jigsaw puzzle with its pieces supplied by the support activity of knowledge acquisition. This activity structures and interlinks the knowledge independent of a specific knowledge representation paradigm and ontology language. To create a set of intermediate representations, i.e. semi-formal specifications in tabular and graph notations within this activity, the ontology developer performs eleven sub-activities as described in Figure 16.

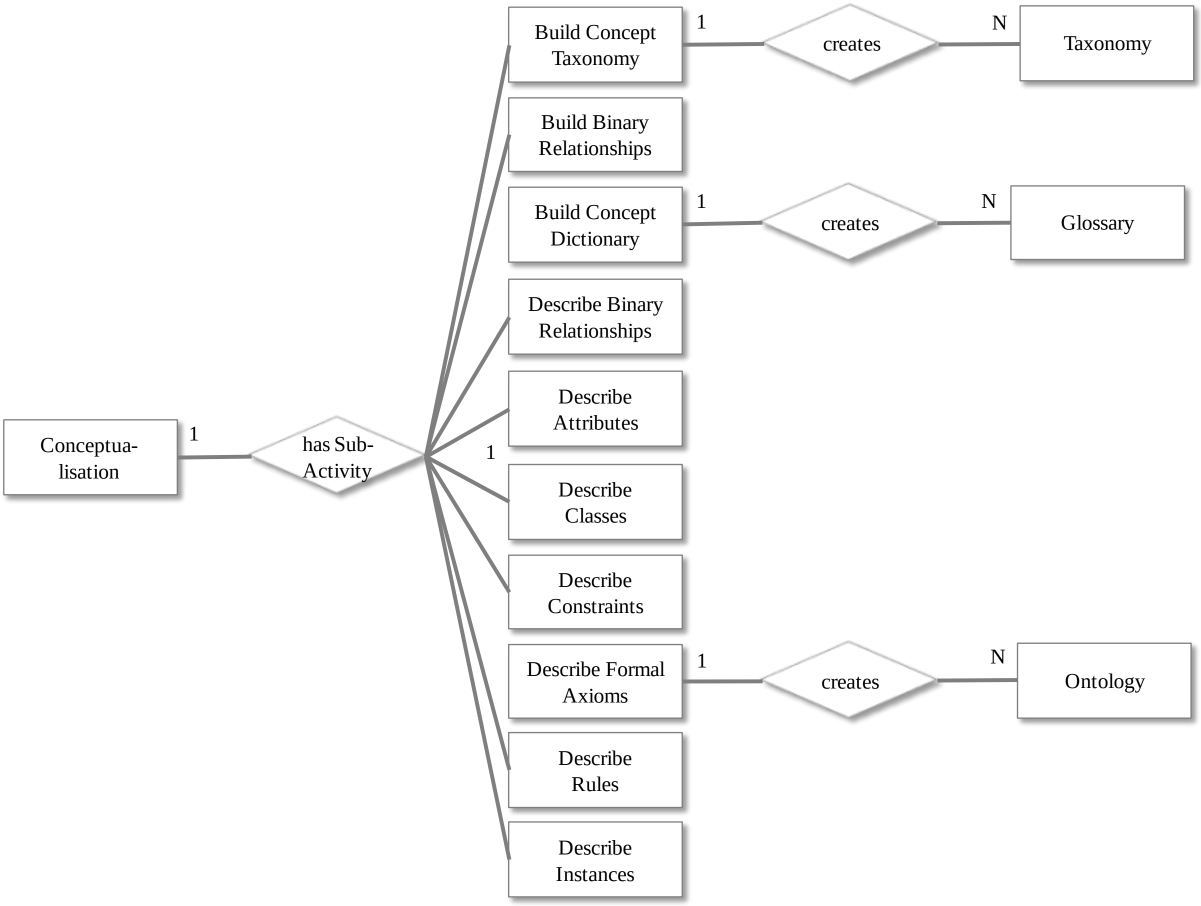

Figure 16: Method Metamodel of Conceptualisation in MET

Formalisation generates a formal model, whereas *implementation* applies an ontology language on the formalisation to create a computable model. The activity of formalisation is optional and the activity of implementation does not subscribe to a specific ontology language.

Maintenance deals with updating, refining, and reusing the ontology.

Furthermore, MET classifies knowledge acquisition as a support activity that starts with the development activity of specification and ends with the development activity of maintenance.

METHONTOLOGY incorporates two distinct activities of *knowledge acquisition* and *conceptualisation* that together rely on a constructivist understanding. *Conceptualisation* consists of eleven sub-activities with respectively predefined outcomes (e.g. glossary and taxonomy). Thus, it primarily aims at structuring and interlinking as well as recommends the three approaches as proposed by Uschold and King (1995) as the guiding principles. MET lacks further explicit information about the activity of *knowledge acquisition*, which presumably denotes an understanding in terms of the mere elicitation of knowledge. METHONTOLOGY suggests the role of the ontology developer to perform the different ontology development activities. Further, this method refrains from suggesting techniques and metamodels but proposes the use of tools such as ODE (Ontology Development Environment) (Blázquez et al. 1998) and WebODE (Web Ontology Development Environment) (Corcho et al. 2002; Arpírez et al. 2003).

2.3.2.5 Noy and McGuiness' Method

Noy and McGuiness (2001) proposed an ontology development method (NMM). First, they discussed several issues in ontology development and then posited three general guidelines for developing ontologies:

- There is no single correct way for developing ontologies but rather viable alternatives exist. The purpose and scope of the ontology, however, represents a recommended starting point for ontology development.
- Ontology development is supposed to be iterative.
- Classes and properties of the ontology should be close to the concepts and the relationships in the real world. Classes most likely relate to the nouns and properties to the verbs that are used for describing the domain of interest.

Second, the authors defined seven activities that structure the ontology development process (Figure 17).

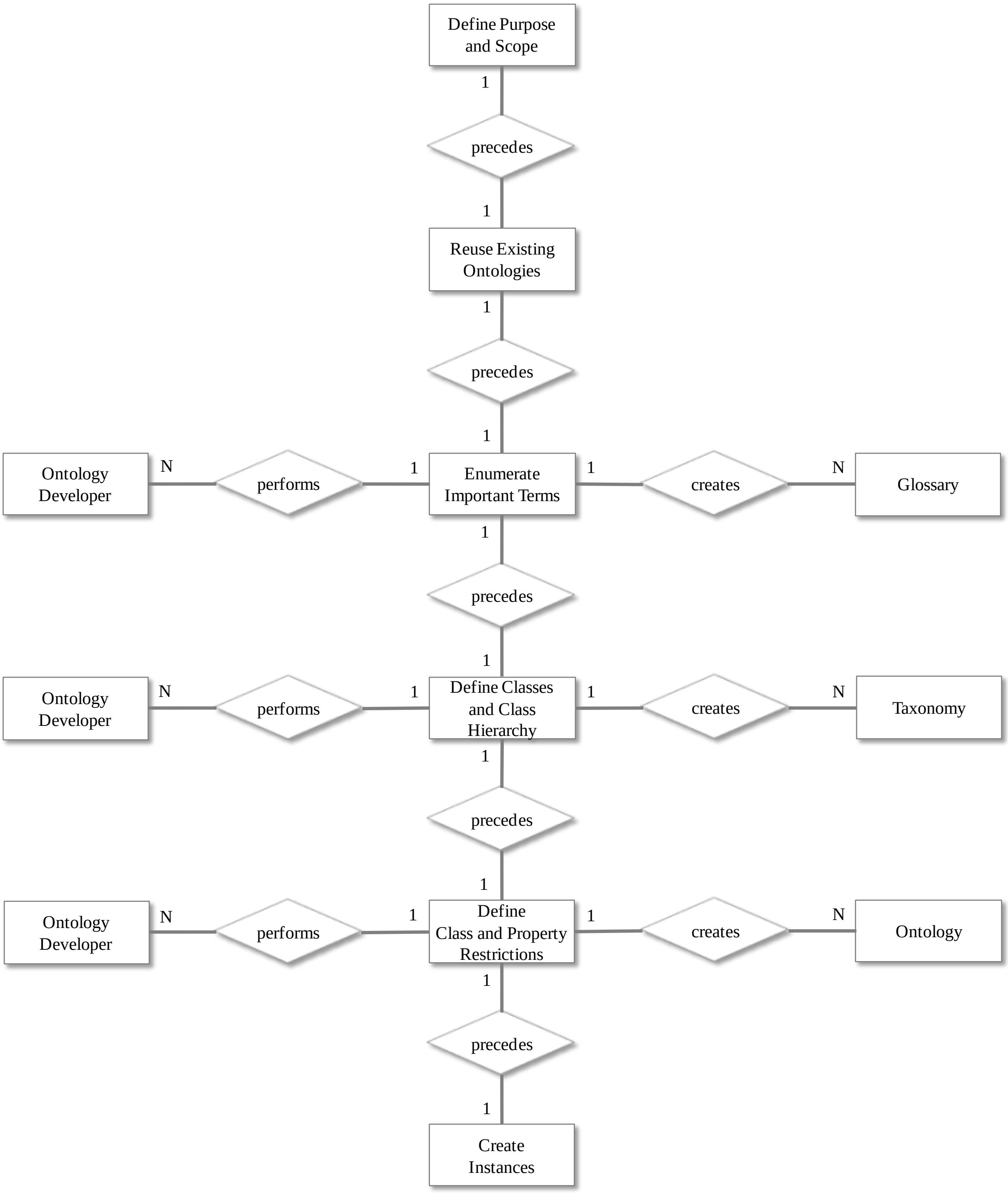

Figure 17: Method Metamodel of NMM

As shown in Figure 17, the activities are as follows:

1. The activity *define the purpose and scope* of the ontology by means of competency questions as proposed by Uschold and King (1995).
2. The activity *reuse of existing ontologies* concerns the identification of appropriate ontologies and their potential integration.

3. The activity *enumerate important terms* generates a comprehensive list of the terms about the domain of interest and organises them into a glossary.
4. The activity *define classes and class hierarchy* uses either a top-down, middle-out, or bottom-up approach as proposed by Uschold and King (1995).
5. The activity *define properties* establishes the internal structure of the ontology beyond taxonomic relations.
6. The activity *define class and property restrictions* comprises the use of domain and range restrictions, cardinalities, and value types.
7. The activity *create instances* populates the ontology. It requires selecting a particular class, creating an instance of that class, and filling in the (restriction) values.

Noy and McGuiness' Method belongs to knowledge acquisition based on a constructivist understanding and, referring to this, includes four activities, i.e. *enumerating important terms*, *defining classes and class hierarchy*, *defining properties*, and *defining class and property restrictions*. These activities aim at structuring and interlinking while, particularly, the third and fourth activity have predefined outcomes (glossary and taxonomy). NMM offers three abstract guidelines for ontology development and specifically the fourth activity proposes the use of the principles proposed by Uschold and King (1995). All activities are subject of the role of the ontology developer. Selecting knowledge acquisition techniques is delegated to this role, though the method recommends using the Protégé Ontology Editor and Knowledge Acquisition System. Competency questions serve as a mean for mitigating linguistic and cognitive problems. NMM also lacks information about method and outcome metamodels for knowledge acquisition.

2.3.2.6 On-To-Knowledge Method

Staab et al. (2001) proposed the On-To-Knowledge (OTK) method as a result of the On-To-Knowledge project. This project studied the application of ontologies on electronically available information for enhancing the quality of knowledge management in large and distributed organisations. Therefore, the project proposed a set of methods and tools access large amounts of semi-structured and textual data sources from web sources.

The OTK method defines five activities as shown in Figure 18.

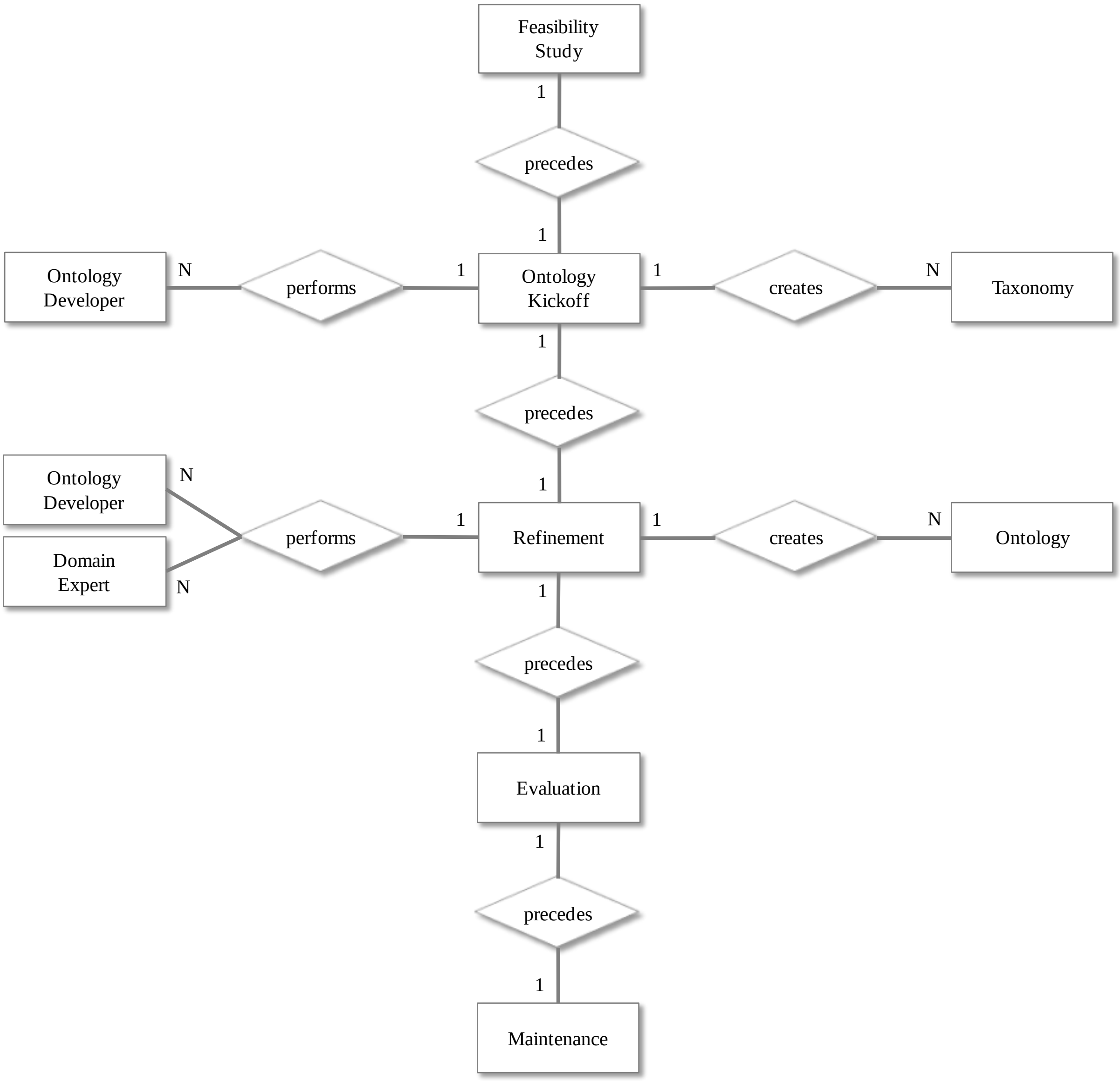

Figure 18: Method Metamodel of OTK

Feasibility study establishes the project setting for ontology development. This setting includes the identified problem and opportunity areas. A selection of the most promising focus areas and target solutions in the context of organisational knowledge management is covered as well. The study assesses the economical and technical feasibility of the entire ontology development project. As such, the feasibility study needs to be conducted before ontology development starts.

Ontology kickoff specifies the requirements for ontology development. This activity deals with requirements specification, ontology reuse, and the development of a baseline taxonomy. The *requirements specification* defines the purpose and scope of the ontology, guidelines for ontology design (e.g. naming conventions), available knowledge sources (e.g. books, magazines, newspapers), potential users, use cases, and target applications. To support requirements specification, OTK suggests the use of competency questions as proposed by Uschold and King

(1995). The ontology developer should asses existing ontologies with respect to their potential for *ontology reuse* and integration and then develop a draft of a *baseline taxonomy*. This taxonomy captures the most relevant concepts of the domain of interest.

Refinement develops an application-oriented ontology and has two sub-activities: *knowledge elicitation* with domain experts and *formalisation*. In the former sub-activity, the developers interact with domain experts based on the initial baseline taxonomy to gather the relevant knowledge, and then, built an intermediate representation (Fernandéz-López et al. 1997). The latter sub-activity formalises the abstract representation by using an ontology language (Figure 19).

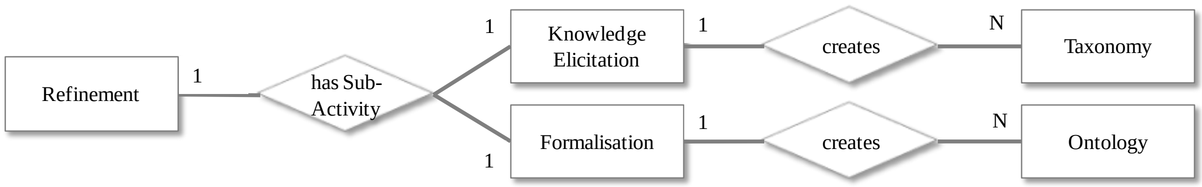

Figure 19: Method Metamodel of Refinement in OTK

Evaluation assesses the effectiveness of the ontology for the target application. It contains two sub-activities. The first sub-activity assesses the ontology with respect to the requirements and competency questions, whereas the second sub-activity studies the use of the ontology in the target application. Evaluation can result in the need for additional iterations of the refinement activity.

Maintenance determines who is responsible for maintenance and how to carry out maintenance. OTK proposes to integrate this activity into the general maintenance of the target application that uses the ontology. Again, maintenance can trigger the need for additional iterations of the refinement activity.

In summary, knowledge acquisition based on a constructivist understanding is subject of two activities, i.e. *ontology kickoff* and *refinement*. Ontology kickoff defines competency questions for mitigating linguistic and cognitive problems though OTK does not provide concrete procedures for this sub-activity. *Refinement* is concerned with knowledge elicitation but appears to aim at structuring and interlinking. It lacks further information about its sub-activities and principles that may guide knowledge acquisition. The OTK method suggests the outcomes of the activities as a taxonomy and ontology respectively. Further, the method assumes two roles (ontology developer and domain expert) and the availability of various knowledge sources (domain experts, documents, etc.). Particular knowledge acquisition techniques (e.g. interview, text analysis) are not explicitly reported. Similar to that, metamodels and recommendations for tool support were not found in the method.

2.3.2.7 A Helix-Spindle Model for Ontology Development

Kishore et al. (2004b) proposed the so-called helix-spindle model for ontology development (HSM). The HSM aims at building ontologies based on principles of software engineering, ontology engineering, and experiences made from developing an ontology for multiagent-based integrative business information systems (i.e., the Multiagent-based Integrative Business Modelling Language). The key characteristic of the HSM is the combination of theoretical and pragmatic approaches instead of solely relying on a single approach. The method defines an incremental forward-lockstep build-test process, which consists of three major phases: conception, elaboration, and definition (Figure 20).

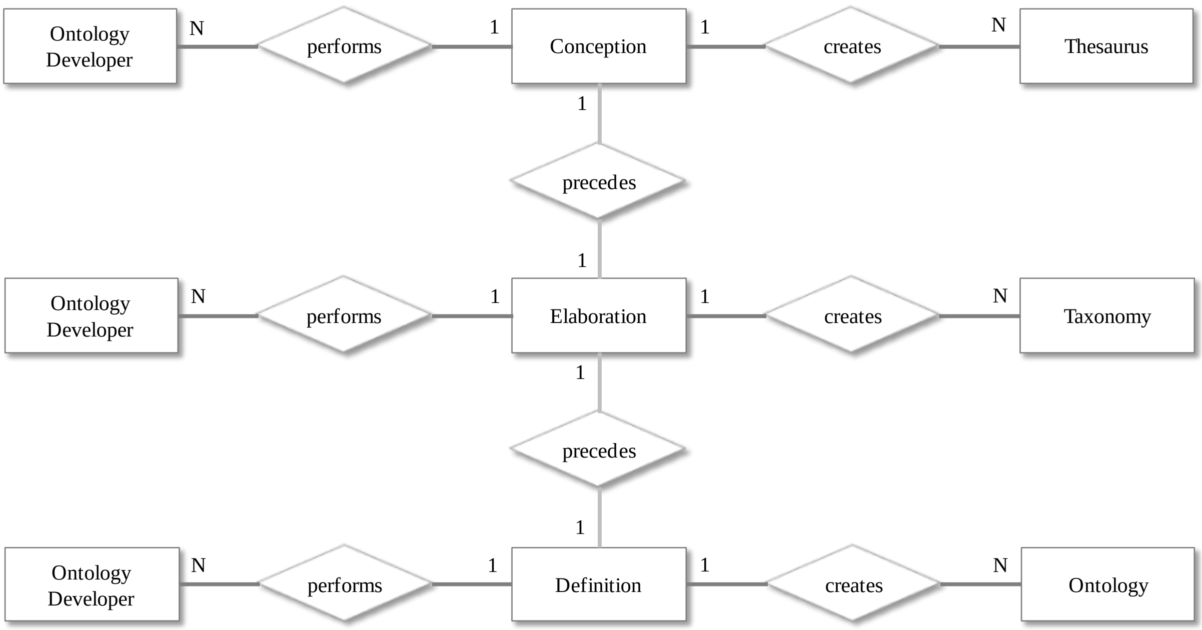

Figure 20: Method Metamodel of HSM

Conception defines an informal representation of the relevant knowledge. This representation corresponds to a conceptualisation, which is specified by means of an informal ontology, e.g. natural language.

Elaboration concerns the refinement of the informal representation and applies a semi-formal ontology, e.g. the Unified Modelling Language (UML), to construct a graphical representation of the relevant knowledge.

Definition generates a formal representation by using an ontology language such as FOL.

With regard to knowledge acquisition, the *conception* phase concentrates on structuring and interlinking based on a constructivist understanding. Therefore, this activity combines the principles of deduction and induction. However, the HSM disregards linguistic and cognitive problems and lacks further information about knowledge acquisition such as method and outcome metamodels. The method involves the role of the ontology developer who decides on knowledge acquisition techniques and tools.

2.3.2.8 The Unified Process for Ontology Building

De Nicola et al. (2009) proposed the Unified Process for Ontology Building (UPON), which builds upon the Unified Software Development Process (Jacobsen et al. 1999). It aims to support the development of large-scale ontologies. For instance, the ATHENA (Advanced Technologies for Interoperability of Heterogeneous Networks and their Application) project used UPON for developing an e-procurement domain ontology (Ruggaber 2006).

UPON proposes a use case driven, iterative, and incremental development of ontologies. The goal of *use case driven* development is to ensure that the ontology satisfies its purpose. *Iterative* denotes that each development activity can be repeated several times. *Incremental* means that the iterations extend and refine the ontology. Similarly to the Unified Software Development Process, UPON distinguishes between cycles, phases, iterations, and workflows. Each *cycle* consists of four phases. The four *phases* are inception, elaboration, construction, and transition. The completion of the four phases results in a new version of the ontology. Each phase may have multiple *iterations*, which consist of five *workflows*. These workflows are defined as requirements, analysis, design, implementation, and test. UPON involves ontology developers and domain experts. The involvement of domain experts concerns the early workflows, whereas the ontology developer mainly participates in the later workflows.

For the purpose of this review, the five phases and the five workflows are of interest. The phases of UPON are as follows:

1. *Inception* focuses on gathering requirements and deals with an initial conceptual analysis.
2. *Elaboration* concerns the (conceptual) analysis for identifying and loosely structuring basic concepts, i.e. classes and properties.
3. *Construction* incorporates most of the design and implementation of the ontology.
4. *Transition* is about testing the ontology.

These phases are used for structuring all five workflows, which are subject of the following paragraphs (Figure 21).

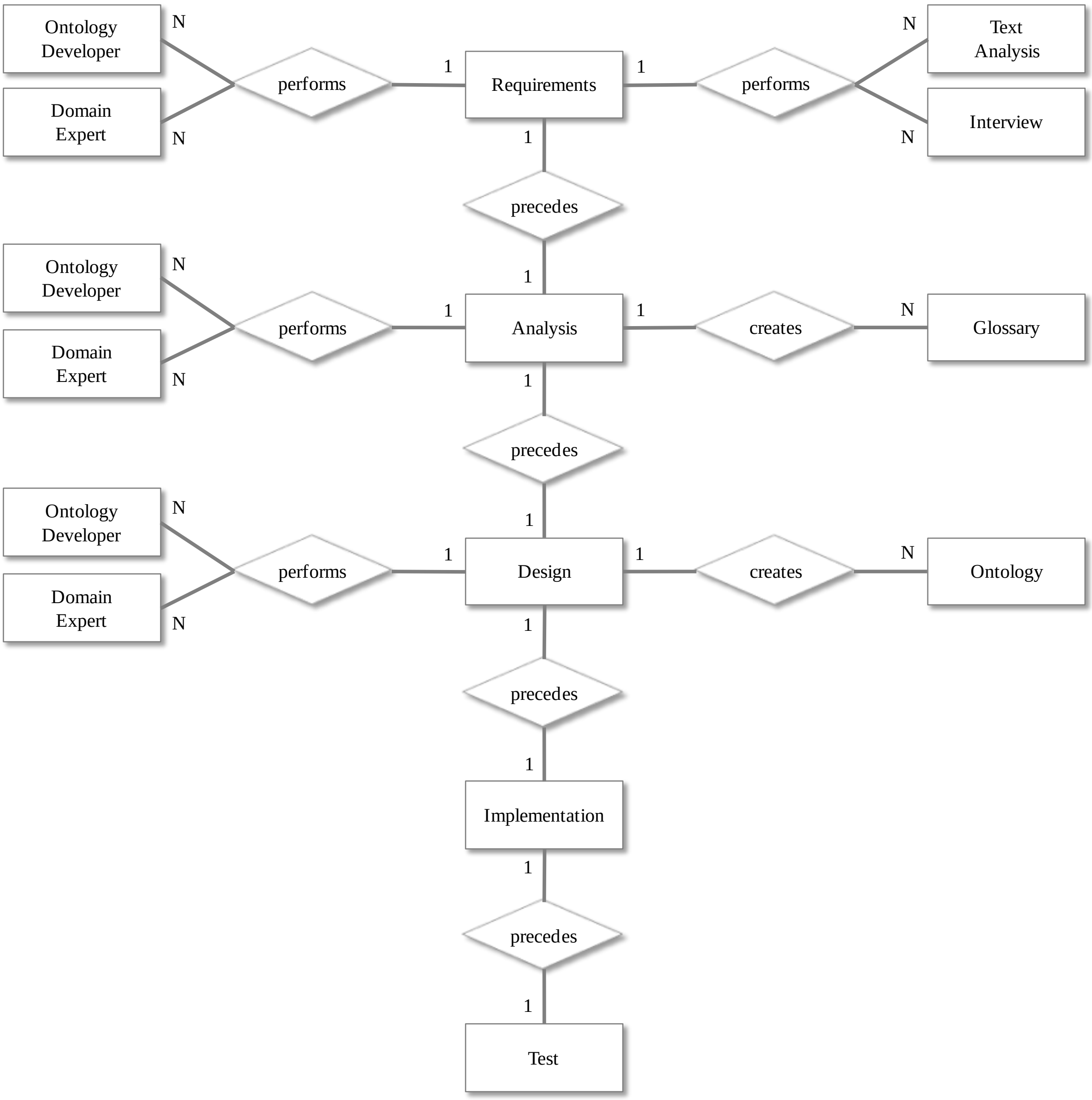

Figure 21: Method Metamodel of Unified Process for Ontology Building

Requirements specifies the requirements for ontology development. Therefore, it aims at reaching an agreement between the users and the ontology developers and comprises six activities.

Analysis comprises four activities to structure the outputs of the previous workflow. The first sub-activity analyses textual knowledge sources (e.g. reports, technical manuals) to acquire domain knowledge for transforming the application lexicon into a domain lexicon. The second activity merges the application and domain lexicon to create a reference lexicon, which should at least include all the terms of the intersection of the application and domain lexicon. The third activity uses UML diagrams, i.e. use case diagrams, class diagrams, and activity diagrams to represent application scenarios. The fourth activity adds informal semantics to the reference lexicon for constructing the reference glossary, which represents the major outcome of this workflow (Figure 22).

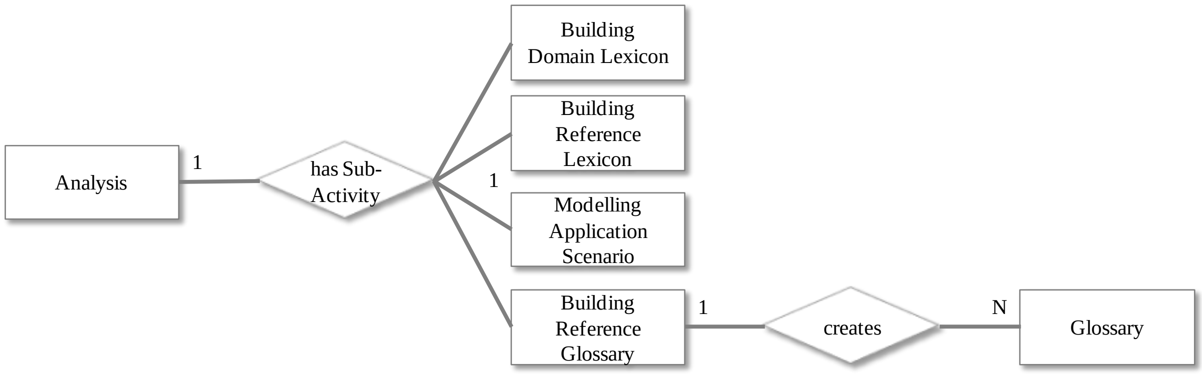

Figure 22: Method Metamodel of Analysis in UPON

Design organises the terms of the reference glossary in class hierarchies, attaches attributes to classes, and adds axioms. Therefore, this workflow comprises an activity that classifies the terms of the reference glossary with regard to the components of ontology. Another activity formally structures the components of the ontology and introduces formal axioms (Figure 23).

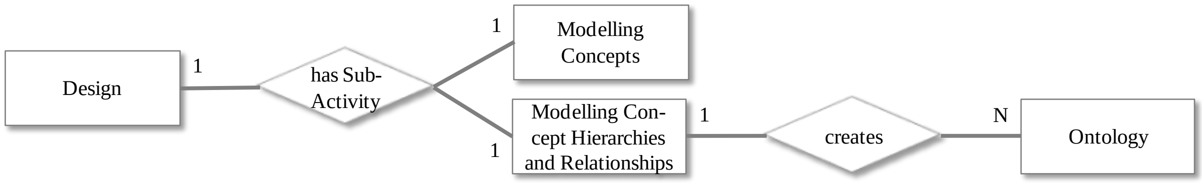

Figure 23: Method Metamodel of Design in UPON

Implementation applies an ontology language on the outcome of the previous activity for reasons of ontology implementation.

Test evaluates the ontology through verifying its semantic quality (coherence and consistency) and pragmatic quality (fidelity, relevance, and completeness). Syntactic quality is subject to the implementation workflow, whereas social quality as defined by Semiotics Theory (Burton-Jones et al. 2005) can only be checked after publication of the ontology.

In summary, UPON comprises three phases, i.e. *inception*, *elaboration*, and *construction* as well as three workflows, i.e. *requirements*, *analysis*, and *design* for knowledge acquisition based on a constructivist understanding. *Inception* and partially *elaboration* as well as *requirements* and partially *analysis* consider the identification and interpretation. Storyboards, competency questions and use cases are applied to mitigate linguistic and cognitive problems. *Elaboration* and *construction* as well as *analysis* and *design* partially cover structuring and interlinking. The method involves both the role of the ontology developer and domain expert as well as defines knowledge acquisition techniques. However, UPON lacks principles, does not supply metamodels, and it is independent of specific tools for supporting knowledge acquisition

2.3.2.9 A Generic Ontology Development Framework

A generic ontology development framework (GODF) proposed by Rajpathak and Chougule (2011) aims at the systematic construction of various types of ontologies. For instance, GODF was applied to develop a fault diagnosis ontology and an equipment spare parts ontology for decision-making support in the automotive and shop floor management domain, respectively (Rajpathak and Chougule 2011, pp. 162-163).

GODF represents a comprehensive method, which defines three phases, nine activities, and 20 sub-activities (Figure 24).

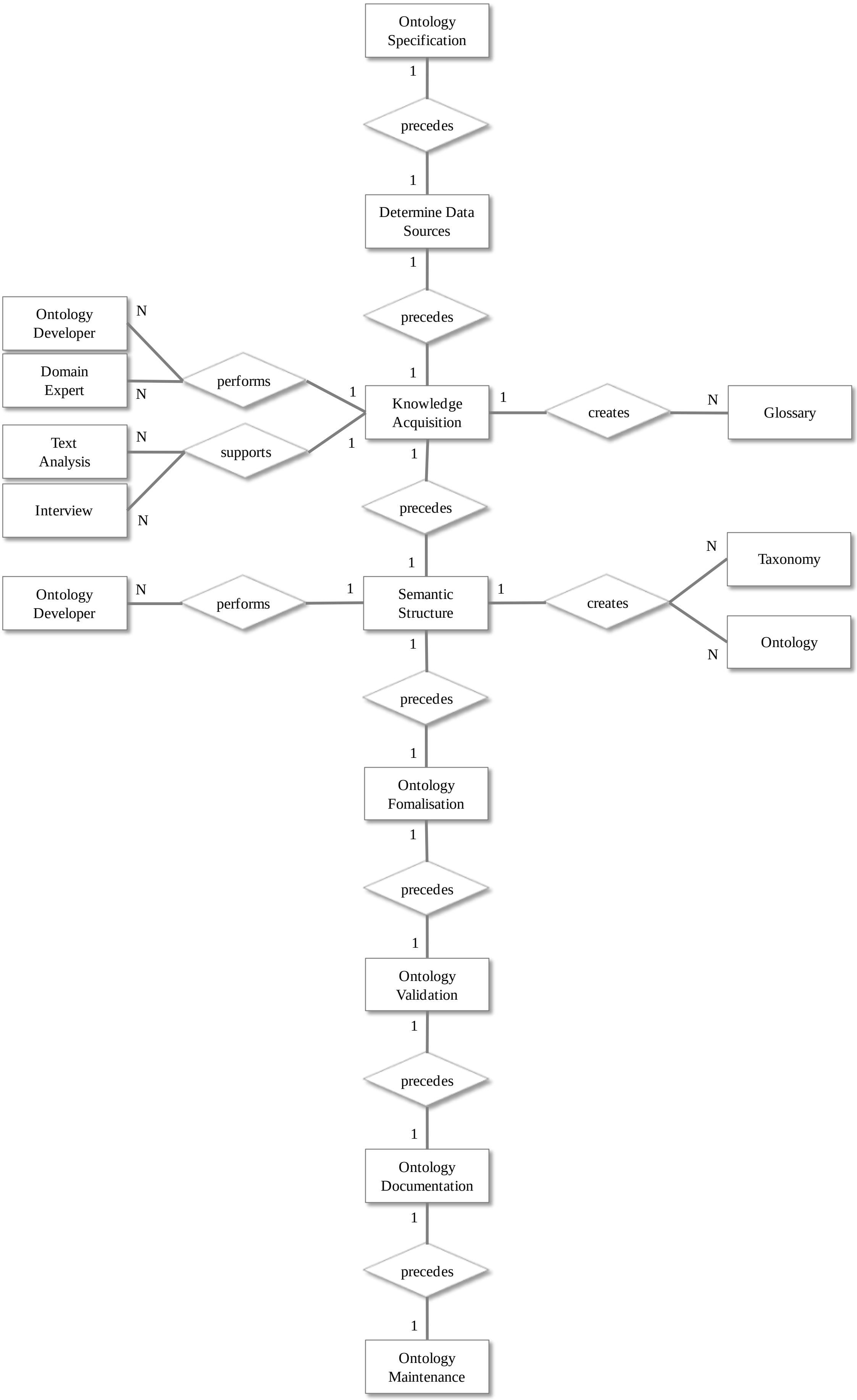

Figure 24: Method Metamodel of Ontology Development in GODF

Knowledge acquisition is subject of two phases.

The *pre-development phase* comprises ontology specification document, determine data sources, and knowledge acquisition.

Ontology specification document provides a complete characterisation of the planned ontology. It encompasses four sub-activities for documenting the definition of the ontology's scope and purpose, the results of an initial analysis of frequently occurring concepts and relationships in the domain of interest, the competency in terms of target applications, and the degree of formal semantics.

Determine data sources performs a survey of frequently used data sources in the domain of interest. This activity has three sub-activities, which focus on identifying the relevant databases, textual documents, and domain experts.

Knowledge acquisition captures the knowledge of the domain of interest. Therefore, it consists of knowledge elicitation to illuminate and transfer the relevant knowledge, semi-automatic concept extraction for mining the identified data sources, and conducting structured and semi-structured interviews (Figure 25).

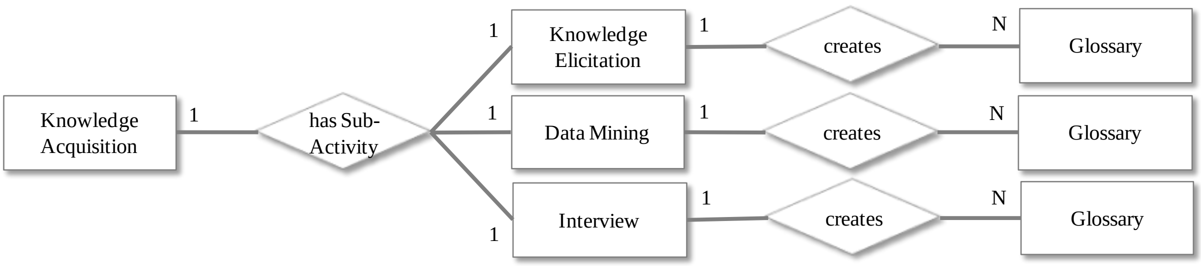

Figure 25: Method Metamodel of Knowledge Acquisition in GODF

The *development phase* has three activities but, due to the scope of this inquiry, the focus is on semantic structure and ontology formalisation.

Semantic structure uses the outcomes of the knowledge acquisition activity as the input for performing three sub-activities. These sub-activities comprise the definition of domain concepts in class-subclass hierarchies together with attributes, relations, and axioms (Figure 26).

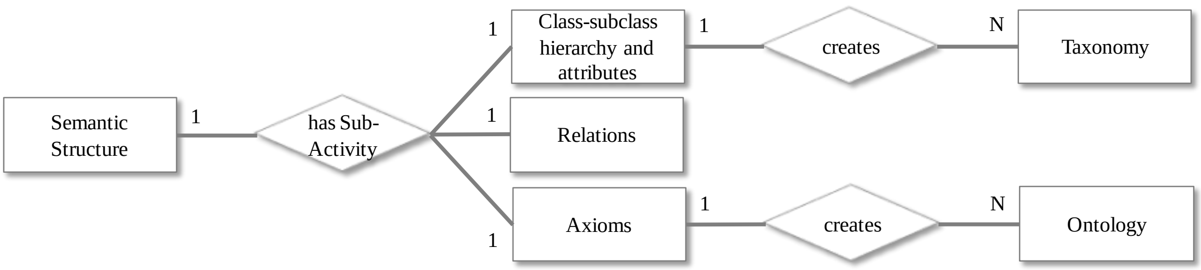

Figure 26: Method Metamodel of Semantic Structure in GODF

Ontology formalisation comprises three sub-activities: selecting a knowledge management tool and ontology language, generating application-specific instances, and merging and aligning complementary ontologies.

In Summary, based on a constructivist understanding, knowledge acquisition in the generic ontology development framework is subject of three activities i.e., *ontology specification document*, *knowledge acquisition*, and *semantic structure*. Thereby, the activities of ontology specification document and knowledge acquisition focus on the identification and interpretation, whereas the activity of semantic structure concerns the structuring and interlinking. It can be assumed that domain analysis and surveys are applied to mitigate linguistic and cognitive problems in knowledge acquisition, which is guided by the principles top-down and bottom-up. Despite the method lacks information about metamodels, it proposes the role of the ontology developer and domain expert, recommends text analysis and interviews as techniques, defines the outcomes (glossary, taxonomy and ontology), and suggests custom tool support for (semi-)automatic concept extraction.

2.3.2.10 Methods from Knowledge Engineering

In contrast to the methods discussed in the preceding sections, the three methods below originate from the discipline of knowledge engineering but may provide relevant insights.

Knowledge acquisition is also central to knowledge engineering as it concerns the development of knowledge-based systems. Based on the two predominant paradigms in the history of knowledge engineering (Section 2.1.3.2), methods can be assigned to the two groups that account for the view of knowledge acquisition (mining view vs. modelling view).

Knowledge engineering methods that subscribe to the mining view directly transfer and transform the required knowledge to develop a knowledge-based system (Buchanan et al. 1983; Freiling et al. 1985; Harmon and King 1985). This group restricts knowledge acquisition to the mere elicitation of knowledge (mining view); thus, it disregards linguistic, cognitive, modelling and methodical difficulties of knowledge acquisition.

In contrast, *knowledge engineering methods that subscribe to the modelling view* follow the constructivist understanding of knowledge acquisition (Studer et al 1998, pp. 163, 168-175). This group of methods provides the prerequisites for subsequent analysis. The most prominent methods are CommonKADS, MIKE, and Protégé-II, which all made important contributions to the field and inspired many other methods.

CommonKADS was introduced by Schreiber et al. (1994; 2002) as a sequel to KADS (Knowledge Acquisition and Documentation Structuring) (Schreiber et al. 1993). For the development of knowledge-based systems, CommonKADS considers the construction of six different models: organisation model, task model, agent model, communication model, design model, and knowledge model. In particular, the knowledge model captures and represents the task, inference, and domain knowledge. CommonKADS puts strong emphasis on (re-)using partial knowledge models. These partial knowledge models are some form of task templates. Task templates specify the task and inference knowledge for a specific task type to guide knowledge acquisition.

Protégé-II (Eriksson et al. 1995; Grosso et al. 1999; Puerta et al. 1992) focuses on the reuse of ontologies and Problem-Solving Methods (PSM) to develop knowledge-based systems. PSM decompose a particular task type into a set of subtasks down to a level on which primitive methods (mechanisms) are available to solve the corresponding subtasks. As such, PSM establish a task-method-decomposition structure. At the lowest level, Protégé-II proposes to define mappings between the mechanisms and finally an ontology that captures domain knowledge.

MIKE (Model-based and Incremental Knowledge Engineering) was proposed by Angele et al. (1998) as a comprehensive method that spans from knowledge acquisition to the implementation of knowledge-based systems. MIKE relies on the CommonKADS knowledge model and transforms it from a semi-formal representation into a formal representation. The reuse of the CommonKADS knowledge model implies that MIKE also uses task templates for acquiring domain knowledge.

The common characteristic of all three methods is the use of task templates, i.e. partial knowledge models for guiding the acquisition. Partial knowledge models provide a specification of the task and inference knowledge with regard to a particular type of task independent of implementation details. They make the task and inference knowledge explicit and incorporate the use of knowledge roles. Knowledge roles determine the role of domain knowledge for solving a particular task type. Therefore, partial knowledge models are blueprints that guide knowledge acquisition.

Partial knowledge models come in different forms as follows: Problem-Solving Methods, Role-Limiting Methods, Configurable Role-Limiting Methods, Generic Tasks, and the Task-Structure Approach.

Problem-Solving Methods propose a task-method-decomposition structure in terms of a set of inference actions, the operational sequence of these inference actions, and the associated knowledge roles. These knowledge roles determine the role of domain knowledge with respect to an inference action (Birmingham and Klinker 1993).

Role-Limiting Methods (RLM) are implementations of specific PSM that are capable to solve a particular task type. RLM have a fixed structure for the acquisition of domain knowledge (e.g. knowledge roles) but they lack flexibility when a task requires a combination of several PSM. For instance, the RLM SALT used the so-called PSM propose-and-revise for solving the parametric design task of elevator configuration (Marcus 1988; Marcus and McDermott 1989; Marcus et al. 1988).

Configurable Role-Limiting Methods (CRLM) have been proposed to increase the flexibility of RLM. CRLM decompose a complex PSM into several subtasks so that different methods can be assigned to solve each of the subtasks. In addition, CRLM provide predefined communication paths and have a fixed schema of knowledge types. This schema specifies the structure of domain knowledge that is required for solving the particular task type (Poek and Gappa 1993; Puppe et al. 1996).

Generic Tasks (GT) are building blocks of the problem-solving component of a knowledge-based system. GT provide a generic description of input and output, a fixed schema of knowledge types to specify the structure of the domain knowledge, and a fixed problem-solving strategy to determine the inference steps (Bylander and Chandrasekaran 1987; Chandrasekaran 1986).

The *Task-Structure Approach* was proposed to overcome two major limitations of Generic Tasks. First, GT mix up the concept of task and PSM since each GT includes a predetermined problem-solving strategy. Second, GT have different degrees of complexity so that the appropriate level of granularity remains unspecified (Chandrasekaran et al. 1992).

In summary, these different forms of partial knowledge models serve as a blueprint to guide knowledge acquisition for reducing linguistic, cognitive, modelling, and methodical difficulties. Partial knowledge models use explicit procedural knowledge that is inherent to a particular type of task. Based on that task, knowledge roles and fixed schema of knowledge types predefine and specify the domain knowledge.

Partial knowledge models conceptually represent a candidate approach for reducing the problem of knowledge acquisition. Their use requires both the existence of a problem-solving task and the specification of its characteristics as a sufficient basis for knowledge acquisition. The existence of a problem-solving task is inherent to the development of knowledge-based system; however, it does not necessarily constitute a compulsory condition for ontology development. This issue is accompanied by the fact that knowledge acquisition by means of partial knowledge models presupposes an appropriate specification of the task and its characteristics. Nonetheless, partial knowledge models have contributed to advance knowledge acquisition within knowledge engineering and have the potential to inspire further approaches for reducing the problem of knowledge acquisition in ontology development.

2.3.3 Summary and Implications

The results from reviewing existing knowledge acquisition methods are summarised with regard to their capabilities for reducing the problem of knowledge acquisition.

The review results demonstrate that none of the knowledge acquisition methods conclusively addresses linguistic and cognitive difficulties. Three methods (CYC, MET, and HSM) disregard linguistic and cognitive difficulties, whereas six methods (GFM, UKM, NMM, OTK, UPON, and GODF) partly consider them by proposing various means such as scenarios or competency questions. However, no method reports how these means contribute to reducing these two difficulties. Regarding modelling difficulties, all but two methods focus on either the identification and interpretation of data (CYC, GFM, and UKM) or the structuring and interlinking of information (MET, NMM, OTK, and HSM). Only UPON and GODF, consider both, which indicates the growing awareness of dealing with modelling problems. However, shortcomings exist in the creation of conceptualisations in terms of the identification and elicitation of (data), the interpretation of this data (information), and the structuring and interlinking of this information (knowledge). Table 9 provides a summary of these findings.

	Linguistic Difficulties	Cognitive Difficulties	Modelling Difficulties
CYC	Not considered		Identification, interpretation
GFM	Partly considered by scenarios and competency questions		Identification, interpretation
UKM	Partly considered by scenarios and brainstorming		Identification, interpretation
MET	Not considered		Structuring, interlinking
NMM	Partly considered by competency questions		Structuring, interlinking
OTK	Partly considered by competency questions		Structuring, interlinking
HSM	Not considered		Structuring, interlinking
UPON	Partly considered by storyboards, competency questions, and use cases		Identification, interpretation, structuring, interlinking
GODF	Partly considered by domain analysis and survey		Identification, interpretation, structuring, interlinking

Table 9: Linguistic, Cognitive and Modelling Difficulties in Knowledge Acquisition

All knowledge acquisition methods are afflicted with methodical difficulties. Each method provides at least one specific activity for knowledge acquisition and defines at least one outcome (e.g. glossary, taxonomy, ontology). However, none of these methods provides corresponding metamodels for the method and its outcomes. Similarly, all methods centre on the role of the ontology developer whereas only two methods (OTK and GODF) additionally involve domain experts. Knowledge acquisition techniques (e.g. text analysis, interviews) and tools (e.g. Protégé, (Web-)ODE) are proposed (CYC, MET, NMM, and GODF). Principles for supporting knowledge acquisition primarily focus on either top-down and bottom-up approaches or deduction and induction. These results, which are summarised in Table 10, corroborate that the current methods lack adequate means to assist ontology developers in acquiring domain knowledge effectively.

	Methodical Difficulties						
	Activities	**Outcomes**	**Roles**	**Techniques**	**Meta-models**	**Tools**	**Principles**
CYC	5 activities	Ontology	Ontology developer	Text analysis	Not reported	Tool (not defined)	Bottom-up
GFM	1 activity	Ontology	Ontology developer	Not reported	Not reported	Not reported	Intuition
UKM	1 sub-activity	Thesaurus, ontology	Ontology developer	Not reported	Not reported	Not reported	Top-down, middle-out, bottom-up
MET	2 activities, 14 sub-activities	Glossary, taxonomy, ontology	Ontology developer	Not reported	Not reported	(Web) ODE	Top-down, middle-out, bottom-up
NMM	4 activities	Glossary, taxonomy, ontology	Ontology developer	Not reported	Not reported	Protégé	3 general guidelines, top-down, middle-out, bottom-up
OTK	2 activities	Taxonomy, ontology	Ontology developer, domain expert	Not reported	Not reported	Not reported	Not reported
HSM	1 activity	Thesaurus, taxonomy, ontology	Ontology developer	Not reported	Not reported	Not reported	Deduction, induction
UPON	3 phases, 3 work-flows	Glossary, ontology	Ontology developer, domain expert	Text analysis, interviews	Not reported	Not reported	Not reported
GODF	3 activities, 7 sub-activities	Glossary, taxonomy, ontology	Ontology developer, domain expert	Text analysis, interviews	Not reported	Custom tool	Top-down, bottom-up

Table 10: Methodical Difficulties in Knowledge Acquisition

Reflecting on these review results leads to the question of how to enhance the state-of-the-art. The following consideration provides a potential direction. Partial knowledge models for knowledge acquisition as used in knowledge engineering methods represent a promising but yet only rudimentarily exploited approach in ontology development. Partial knowledge models originally guide knowledge acquisition based on explicit procedural knowledge of a particular type of task. The required knowledge about the domain of interest is specified in terms of knowledge roles and schema.

Based on this general idea, this thesis proposes theory-based knowledge acquisition for ontology development. Theory-based knowledge acquisition suggests guiding knowledge acquisition by using theories in business economics. These theories can be regarded as partial knowledge models, which describe the business domain of interest by constructs, relationships, and dependencies, which then allow for guiding knowledge acquisition.

The role of theories in business economics with regard to the problem of knowledge acquisition is summarised in Table 11.

Problem of Knowledge Acquisition	**Role of Theories in Business Economics**
Linguistic Difficulties	Providing a common language containing terms and definitions of a particular domain of interest for enhancing the communication about this domain (e.g. clarifying obscurities, potential misunderstandings).
Cognitive Difficulties	Providing models that incorporate the main constructs, relationships, and dependencies of a particular domain of interest for enhancing the understanding of this domain (e.g. supporting the effort, subjective judgement for understanding).
Modelling Difficulties	Providing models that serve as a blueprint for creating conceptualisations by grounding the identification and elicitation of data, the interpretation of this data (information), and their structuring, interlinking (knowledge).
Methodical Difficulties	Providing models that capture the characteristics of a domain of interest for designing specific methods for knowledge acquisition (e.g. incorporating domain-specific best practices, guidelines to handle the complexity of the domain).

Table 11: Knowledge Acquisition and the Role of Theories in Business Economics

3 Requirements

This chapter reports the deduction of requirements from a theory in the field of business economics as the basis for designing a method for theory-based knowledge acquisition. First, the domain of transport chains is defined as the underpinning example. Second, an adequate theory from the organisational sciences is selected. Third, a model of this theory is selected. Fourth, the actual requirements are deduced by referring to theoretical constructs.

3.1 Domain Description

For demonstrating the development of a theory-based knowledge acquisition method, the domain of transport chains represents the underlying example. The term transport chain not only constitutes a key concept in the disciplines of Logistics and Logistics Management (DIN 1989; Pfohl 2010) but also in Operations Management (Thonemann 2010) and more specifically Supply Chain Management (Christopher 2005; Mentzer 2001).

A transport chain is defined by the German Institute for Standardisation as the ordered set of technically and organisationally interlinked actions to realise a flow of persons and/or goods from source to destination (DIN 1989, p. 3). Inspecting this definition in more detail unveils three key characteristics of transport chains:

– *Transport chain goals*, which govern the transport chain organisation as well as coordinate the transport chain actions and roles.

– *Transport chain organisation* in terms of an ordered set of technically and organisationally interlinked actions concerns both *structural and procedural issues*.

– *Transport chain actions and roles*, which realise the flow of logistics objects form source to destination.

These three characteristics also reflect the understanding of transport chains in Supply Chain Management. In this context, Mentzer et al. (2001, pp. 3-5) study the evolution of various definitions of supply chain and shape the concept of the ultimate supply chain. The ultimate supply chain refers to "as a set of three or more entities (organisations or individuals) directly involved in the upstream and downstream flows of products, services, finances, and/or information from a source to a customer" (Mentzer et al. 2001, p. 4). This definition incorporates the above characteristics of transport chains. Additionally, it substantiates them with regard to a minimum number of transport chain members, different types of flows along the transport chain, and the term customer. Particularly, this customer-orientation points out that transport chains are subject to (environmental) influence factors. For instance, such influence factors could refer to different and changing customer demands. That is, *transport chain influence factors* add another key characteristic to transport chains.

By combining the disciplines of Logistics and Supply Chain Management, Christopher (2005, p. 6) defines a supply chain as “a network of connected and interdependent organisations mutually and cooperatively working together to control, manage, and improve the flow of materials from suppliers to end users”. This definition exhibits the four characteristic of transport chains as depicted above. Thus, it demonstrates a consensual and inter-disciplinary understanding of transport chains.

The four key characteristics of transport chains, i.e. transport chain goals, transport chain organisation, transport chain actions and roles, as well as transport chain influence factors, are reflected in the following definition, which lays the structure for the subsequent domain description:

> *A transport chain is an ordered set (sequence) of technically and organisationally interlinked actions to realise a flow of logistics objects from source to destination according to customer demands.*

3.1.1 Transport Chain Goals

Transport chain goals both govern the organisation of transport chains as well as coordinate the transport chain actions and roles according to customer demands. These goals consider issues of value creation and economics.

The goal of value creation denotes the effectiveness of transport chains. It deals with the realisation of the flow of logistics objects from source to destination. Realising these flows requires planning, implementation, and monitoring of transport chain actions and roles. Thus, effectiveness measures the degree to which transport chains satisfy the customer demands independent of necessary efforts (Aberle 2009, pp. 506-507; Autry et al. 2008, pp. 38-42).

The economic goal concerns monetary values in terms of profit, return, and turnover. It maintains a means-end relationship to the goal of value creation. As such, economic goals concern the efficiency of transport chains. This efficiency depicts the ratio of transport chain performance and transport chain costs. It induces that efficiency means either to achieve a given level of performance with a minimum of costs or to achieve a maximum level of performance with given costs (Ihde 2001, pp. 1-5; Pfohl 2010, pp. 39-40). For measuring performance, the delivery service in terms of its constituent components provides appropriate means. That is, the *delivery time* (or *lead-time*) depicts the time between an incoming order (received by the provider) and the receipt of goods (by the requester). The *delivery reliability* measures the adherence to the agreed delivery dates. This measure reflects the likelihood of keeping the delivery time exactly. The *delivery quality* determines the quality of the delivered goods with respect to type, amount, damage, etc. This measure depicts the likelihood of customer complaints. Further, the *delivery flexibility* delineates the fulfilment of specific customer demands in terms of quantity of orders, purchase quantity, type of packaging, or shipment tracking (Christopher 2005, pp. 46-50; Pfohl 2010, pp. 32-39).

On the contrary, the monetary values (expenses) to achieve the aspired level of performance correspond to transport chain costs. These costs are typically measured according to a classification scheme, which is based on different types of transport chain actions such as transport, handling, and storage (Pfohl 2010, pp. 29-32; Straube 2004, pp. 56-57).

3.1.2 Transport Chain Organisation

The transport chain organisation coordinates heterogeneous and globally dispersed transport chain actions on a goal-oriented basis. Therefore, the transport chain organisation is divided in a structural organisation and a process organisation, which represent two mutually dependent constituents (Cooper et al. 1997, pp. 5-9; Lambert and Cooper 2000, pp. 69-74, 77-78; Mentzer et al. 2001, pp. 16-17).

The structural organisation considers the arrangement of transport chains from a static viewpoint. It relies on four basic organisational forms described in Logistics and Logistics Management literature: strategic organisation, regional organisation, operational organisation, and virtual organisation (Pfohl 2010, pp. 297-298; Straube 2004, pp. 45-47; Sydow 2010, pp. 373-458):

The *strategic organisation* has a focal company (e.g. manufacturer, wholesaler, retailer), which maintains close and contractually regulated relationships for strategically managing transport chain actors. For this purpose, it undertakes investments in transport chain specific resources for achieving mutual competitive advantages. Target markets typically have highly predictable customer demands and expose a certain degree of overall market stability (e.g. automotive sector).

The *regional organisation* refers to a cluster with occasional but recurring cooperation. This cooperation is built on latent relationships between several, small, and spatially close transport chain actors. Personal contact, similar corporate cultures, and a high degree of specialisation further characterise regional organisations. For instance, the textile industry in Northern Italy exhibits the characteristics of regional organisations.

The *operational organisation* centres on the idea of inter-organisational information systems to enable common access to production and logistics capacities (e.g. for peak load balancing) across transport chains. Operational organisations deal with standardised transactions for value-added processes and particularly focus on capacities instead of physical objects. They are comparable to electronic market places but have a higher degree of organisation.

The *virtual organisation* corresponds to a cooperation between independent transport chain actors with respective core competencies and a joint understanding of their business operations. A major goal of virtual organisations is to realise synergy effects (e.g. economies of scale and scope). Characteristics of virtual organisations are project-based cooperation, mutual trust, spatial distribution, and intense use of IS, absence of detailed contracts and specific investments, as well as a consistent and coherent appearance towards the customers. For instance, virtual organisations emerge in software production, clothing and toys, as well as microelectronics and biotechnology.

Moreover, the process organisation considers procedural issues of transport chains by taking a dynamic viewpoint. Procedural issues comprise the number of transport chain tiers, the types of logistics object flows, and the phases of transport chains (Harland 1996, pp. 66-72; Huang et al. 2003, pp. 1488 1490; Pfohl 2010, pp. 5-7, 151-152):

The *number of transport chain tiers* depicts the different stages of value creation across the entire transport chain. Correspondingly, single-tier transport chains involve two interlinked transport chain actors, whereas multi-tier transport chains incorporate at least three (serially) interlinked transport chain actors.

The *type of logistics object flows* denotes the movement of logistics objects from source to destination. A serial (dyadic) flow of logistics objects delineates direct (linear) relations between source and destination. A divergent flow of logistics objects depicts one-to-many relations between source and destination. The source refers to a break-bulk-point, which splits up the flow of logistics objects. A convergent flow of logistics objects presents many-to-one relations between source and destination. The destination refers to as a consolidation point, which merges the flows of logistics objects.

The *phases of transport chains* classify transport chains according to three consecutive operational sections. The pre-carriage depicts the flow of logistics objects from a source to a consolidation point (e.g. feeder services). The on-carriage presents the flow of logistics objects from a break-bulk point to the (final) destination (e.g. distribution of logistics objects). The main carriage characterises the flow of logistics objects from a consolidation point to a break-bulk point (e.g. airfreight transport).

The choice of a specific transport chain organisation, i.e. a structural and process organisation depends on both the transport chain goals and the influence factors (e.g. global, complex, dynamic, buyer market), which reflect the characteristics of the business environment (Baumgarten et al. 2002, pp. 34-41).

3.1.3 Transport Chain Actions and Roles

Transport chain actions concern spatio-temporal transformations for realising the flow of logistics objects. These transformations can be distinguished according to three main categories: transport, handling, and storage (Gudehus 2010, pp. 990-992; Pfohl 2010, pp. 7-9; Schulte 2009, p. 17):

Transport realises spatial transformations of logistics objects between organisations (inter-organisational transport) through manual and/or technical means. In contrast, intra-organisational transport, which takes place within organisations, complies with conveying.

Handling encompasses the entirety of conveying (and stocking) before and after transport. For instance, conveying and stocking comprises loading, unloading, and reloading of logistics objects. Handling takes place in break-bulk points, consolidation points, and at locations of production and consumption.

Storage realises temporal transformations of logistics objects and corresponds to planned interruptions in the flow of logistics objects. Similar to handling, storage takes place in break-bulk points, consolidation points, and the locations of production and consumption.

Supporting and value-adding actions accompany these transport chains actions. Supporting actions realise quantitative transformations with regard to a logistics objects transport, handling, and storage characteristics. For instance, supporting actions comprise picking, packaging, and signing. Moreover, value-added actions provide additional customer benefits in terms of customs clearance, filling, and collections (Gudehus 2010, pp. 990-992; Pfohl 2010, pp. 7-9; Scholz-Reiter et al. 2008, pp. 582-587).

Moreover, transport chain roles depict the responsibilities and competences independent of the particular transport chain actors. As such, they can be classified based on a transport chain actor’s corporate purpose (Pfohl 2010, p. 4).

On the one hand, transport chain roles that provide abstract descriptions of transport chain actions are second-party, third-party, and fourth-party logistics providers. *Second-party logistics providers* offer single actions such as transport, storage, and handling. Further, *third-party logistics providers* compose single actions to composite transport chain actions. They are also capable to combine these bundles with logistics management capabilities through using their own assets. At last, *fourth-party logistics providers* coordinate transport chain actions along the entire transport chain without possessing own logistics assets. In contrast to these kind of roles, first-party logistics providers focus on intra-organisational flows of logistics objects. Thus, they do not correspond to a transport chain role (Gudehus 2010, pp. 993-998; Scholz-Reiter et al. 2008, pp. 581-589; Straube 2004, pp. 52-55, 214-220).

On the other hand, transport chain roles that do not provide descriptions of transport chain actions typically follow a classification according to the different stages of value creation: supplier, manufacturer, wholesaler, retailer, public households (public authorities), and private households. That is, these roles request and consume transport chains actions instead of providing them (Kaczmarek 2006, pp. 23-26).

3.1.4 Transport Chain Influence Factors

Transport chain organisations and actions are subject of various and changing factors influencing transport chains. As such, these influence factors not only concern the business environment of a transport chain but also directly affect the fulfilment of the function of transport chains in terms of realising the flow of logistics objects.

Specifically in the context of newly emerging (manufacturing) concepts such as quick response, accurate response, efficient customer response, lean and agile manufacturing, or mass customisation, Fisher (1997) raises the question of how to devise the “right” transport chain for specific logistics objects. According to Fisher (1997), devising the “right” transport chain requires assessing the nature of goods (e.g. functional, innovative) in advance as a prerequisite for enabling the organisation of customer-tailored transport chains.

Similarly, Christopher (2000) highlights the risks of lengthy, rigid, and slow-moving transport chains for competitiveness and emphasises the importance of a higher degree of manoeuvrability in fast changing market environments. Manoeuvrability of transport chains is comparable to the organisational ability to respond to environmental changes quickly, e.g. market changes in terms of volume and variety.

Additionally, Kirn et al. (2008, pp. 3-60) delineate that the organisational ability of transport chains to adapt to a changing business environment is a prerequisite for fulfilling individual customer demands. The authors put a focus on exploiting spatial, temporal, and economic potentials for realising an adaptable organisation of transport chains.

Based on that, transport chain influence factors originate from an increasing individualisation of customer demands, e.g. in terms of less predictable (volatile) customer demands and market developments, high variety of customer demands, customers demand for advanced products and services, or customised and lower order sizes with shorter order cycles. With regard to transport chains, these factors can be characterised by the following five effects (Aberle 2009, pp. 91-98; Ihde 2001, pp. 58-66; Pfohl 2010, pp. 46-49, 309-311):

The *substitution effect* characterises the increasing share of road transport. In comparison to rail and air transport, the shift towards road transport is due to its specific properties, which favour tailored transport chains to fulfil the customer demands.

The *freight structure effect* depicts the impacts of changes in the macroeconomic production structure on transport chains. These changes occur in developed economies and appear in form of an increasing share of high quality consumer and capital goods. For instance, this increasing share results in smaller shipment sizes and increasing numbers of piece goods.

The *logistics effect* captures the impacts of emerging concepts in commerce and industry on transport chains. Mainly as a result from growing customer demands (e.g. in terms of lead time, delivery flexibility), this effect challenges not only physical aspects of transport chains but also information systems, which support the organisation of transport chains. Empirical findings show that the logistics effect amplifies the freight structure effect.

The *integration effect* denotes the impacts of the economic integration, e.g. within the European Union, in terms of rising transport volumes, transport distances, and cross-border transports. The integration effect induces increased requirements on the organisation of transport chains primarily with regard to performance and costs.

The *logistics interface effect* circumscribes the impact of using multiple modes of transport. Combining road transport with other modes such as rail or air transport provides additional benefits particularly with regard to the final mile. However, this combination equally accounts for additional efforts for avoiding or solving organisational interface problems.

3.2 Theory Selection

3.2.1 Rationale and Review Strategy

For selecting a theory of business economics from the knowledge base, a survey is performed to study how existing Organisation Theories provide models in terms of partial knowledge models that allow for capturing the main constructs, relationships, and dependencies of the domain of transport chains. This survey encompasses Organisation Theories that make an original contribution to their respective field.

For describing, explaining, predicting, and designing various characteristics and issues of the domain of transport chains, literature frequently relies on the mature body of knowledge inherent to the field of Organisation Theory (cf. Gudehus 2010, pp. 3-98; Ketchen and Hult 2007, pp. 574-579; Klaas 2002; Lemoine and Dagnaes 2003, pp. 211-214; Pfohl 2010, pp. 229-303). Organisation Theory studies various types of organisations with regard to their creation, existence, evolution, change, and functionality. The term organisation has its roots in French and covers three complementary meanings. The (singular) noun *organisation* depicts the concept of functional design and the systematic arrangement or structure. In contrast, the (plural) noun *organisations* delineates groups or associations that represent common interests of several persons for accomplishing shared goals. Finally, the verb *organise* denotes the fact to join together for reasons of pursuing common interests and shared goals (Brockhaus 2005, pp. 4556-4557; Wermke et al. 2006, p. 751).

Since Organisation Theory represents a large area of research, the list of search results was reduced iteratively by adding constraints such as source type (book, journal and proceeding) as well as expanded by adding alternative terms. This procedure led to a shorter list, which was then manually inspected by analysing the abstracts and skimming the content. The search yielded a total of five Organisation Theories. These theories represent major contributions in the field of Organisation Theory (Table 12).

Organisation Theory	Author(s) and Year
Theory of Bureaucracy	Weber 1922, 1972
Management Theory	Taylor 1911; Fayol 1919; Nordsieck 1934; Kosiol 1962; Grochla 1995
Human-Relations Approach	Roethlisberger and Dickson 1939
Behavioural Decision Theory	Simon 1957; March and Simon 1958; Cyert and March 1963; March and Olsen 1976
Contingency Approach	Burns and Stalker 1961; Litwak 1961; Pugh et al. 1963; Pugh and Hickson 1971, 1976; Pugh 1981; Kieser and Kubicek 1992; Donaldson 2001; Kieser and Walgenbach 2007

Table 12: Identified Organisation Theories

Additionally, the search identified the Market-based View (e.g. Porter 1981) and the Resource-based View (Barney 1991a, 1991b) as two further theories to be considered.

3.2.2 Theory Analysis

3.2.2.1 Theory of Bureaucracy

The Theory of Bureaucracy has its roots in the beginnings the 20th century in the work "Wirtschaft und Gesellschaft" (Weber 1922). At that time, an increasing predictability and controllability of various types of problems (e.g. social, technical) by means of science, technology, and organisation, inter alia, induces a process of rationalisation on an institutional level. This process shapes the work of Weber (1922) and results in the concept of bureaucracies. Bureaucracies correspond to legal forms of governance apart from the charismatic and traditional types of governance (i.e. reign). In this understanding, bureaucracies are not restricted to administrations, i.e. authorities, but equally cover (commercial) companies (Weber 1972, pp. 17-30, 122-148).

Based on that, bureaucracies can be described according to the following four characteristics (Weber 1972, pp. 551-565):

Division of labour and specialisation concerns decision-making authorities in terms of factually logical, task-based responsibilities and competences as well as authorities to issue directives. Decision-making authorities and authorities to issue directives are independent of individual members of bureaucracies, which allows for establishing a stable and long-term organisational structure.

Hierarchy depicts a vertically aligned and ordered structure of super- and subordination. This super- and subordination induces that higher hierarchical levels possess corresponding decision-making authorities and authorities to issue directives for supervising lower hierarchical levels.

Regulations reflect technical rules and norms to determine decision-making authorities and authorities to issue directives. Inter alia, these rules and norms concern the tasks needed to accomplish the goals of bureaucracies, the procedures to perform these tasks, and the ways of organisational communication.

Decision-making authorities and authorities to issue directives rely on *documentation and document management*. Both assure controllability of the bureaucracy and continuing operations independent of individual members.

These four characteristics describe the ideal type of bureaucracy. Since this type provides a higher degree of efficiency, it is supposed to be superior in comparison to other types of organisation (Weber 1972, pp. 561-562).

The Theory of Bureaucracy constitutes seminal work in the field of Organisation Theory and leads to the advent of further Organisation Theories. It rather aims at describing and explaining the emergence and functionality of large organisations than formulating principles to devise and optimise organisations. Weber (1972) characterises the ideal type of bureaucracies by four characteristics, which comply with constants. This rigidity exhibits that this theory falls short in considering various types of bureaucracies that differ in their degrees of division of labour and specialisation, the characteristics of the hierarchy, the amount and type of regulations, as well as the degree of documentation and document management. Furthermore, the Theory of Bureaucracy puts a strong emphasis on accomplishing organisational goals by focussing on the organisational structure. Thus, it insufficiently takes account of organisational actions and roles as well as organisational influence factors.

3.2.2.2 Management Theory

Management Theory studies organisations as systems that perform (organisational) actions based on the organisational goals. It represents an amalgamation of various areas of Organisation Theory that reflects the works of Taylor (1911), Fayol (1919), Nordsieck (1934), Kosiol (1962), and Grochla (1995). As such, this theory complies with a set of methods, principles, and guidelines for devising organisations. For instance, guidelines for division of labour, specialisation, and coordination are proposed.

With reference to the process of rationalisation of institutions, Taylor (1911) establishes the *Scientific Management*. It deals with enhancing manufacturing organisations for reasons of higher degrees of efficiency. Therefore, Taylor (1911) proposes four main principles: division between manual and intellectual labour, workload and bonus, selection and adaptation of workforce, as well as reconciliation between workforce and management. These principles result from experiments, which focus on the decomposition of organisational actions down to an elementary level to determine the optimal sequence of their execution. The objectives of these experiments indicate that the organisational principles rather aim at creating solutions for actual organisational problems than verifying scientific hypotheses as the term Scientific Management might indicate.

Fayol (1919) proposes a more systematic approach to Management Theory. In contrast to Scientific Management, Management Theory takes a more comprehensive viewpoint on organisations and proposes 14 general organisational principles to support the management of several types of organisations. These principles represent flexible guidelines in terms of division of labour, authority, discipline, issuing of directives, uniform management subordination of the single interests to the common interest, fair remuneration, centralisation, hierarchical organisation, order, poetic justice, loyalty, initiative, and team spirit. Despite these organisational principles resemble both the principles of Scientific Management (Taylor 1911) and the ideal type of bureaucracy (Weber 1922), Fayol (1919) centres on management processes. Fayol (1919) correspondingly attributes planning, organisation, issuing directives, coordination, and control as the five basic properties to management processes.

In line with an increasing interest in explaining, controlling, and devising organisations, Nordsieck (1934) establishes the basis for examining organisational issues from a *business management* viewpoint. This viewpoint addresses organisational tasks and actions, which leads to a differentiation in a structural and a process organisation. The structural organisation corresponds to a static point of view and concerns the allocation of organisational actions with their associated roles. In contrast, the process organisation takes a dynamic point of view, which concerns the spatial and temporal realisation of organisational actions. Based on that, Kosiol (1962) points out that organisations pursue organisational goals, which need to be transformed and decomposed into organisational tasks to induce corresponding actions. Accordingly, task analysis and synthesis serve as the main instruments for devising organisations. Moreover, Grochla (1995) summarises possible combinations of different organisational characteristics to various types of organisational structures. These structural types allow for devising organisations, for instance, in terms of divisional, functional, or matrix structures.

Management Theory represents an accumulation of different methods, principles, and guidelines that primarily aim at devising organisations. It primarily addresses the design of organisational structures for accomplishing organisational goals. On the contrary, organisational actions and roles are considered less important and organisational influence factors are not covered.

3.2.2.3 Human-Relations Approach

The Human-Relations Approach considers organisations as interactive and behavioural systems. It has a special interest in studying the organisational members with reference to their organisational actions and roles. This approach has its roots in the Hawthorne-Experiments, which were conducted between 1924 and 1934. These experiments demonstrated the effect of human relationships on the satisfaction and motivation of organisational members when carrying out organisational actions and, thus, fulfilling their organisational roles (Roethlisberger and Dickson 1939).

In contrast to the Theory of Bureaucracy and Management Theory, the Human-Relations Approach focuses on the satisfaction and motivation of the organisational members. In particular, the focus is on the behaviour of superiors, relationship within a group of organisational members, and material incentives. The role and characteristics of the organisational structure as a determinant of the organisational actions was largely excluded. This exclusion reveals the main difference between this Approach and the two previously mentioned Organisation Theories. Nonetheless, the Human-Relations Approach has contributed to improve the understanding of possible types of organisations beyond the organisational understanding of the Theory of Bureaucracy and Management Theory.

The Human-Relations Approach considers organisations as interactive and behavioural systems. It studies the satisfaction and motivation of the organisational members as well as their organisational actions and roles with regard to the behaviour of superiors, the relationship within a group of organisational members, and material incentives. In contrast, this approach attaches minor importance to the organisational structure and lacks a consideration of the organisational influence factors.

3.2.2.4 Behavioural Decision Theory

Behavioural Decision Theory originates in the seminal work of Barnard (1938), whereas Simon (1957), March and Simon (1958), Cyert and March (1963), as well as March and Olsen (1976) have contributed to its further development. This theory studies organisations as decision-making systems in which the need to coordinate the decisions of the organisational members with respect to the organisational goals arises.

Behavioural Decision Theory assumes bounded rationality with respect to the decision-making capabilities of the organisational members. Bounded rationality depicts that the organisational members have limited capabilities to make rational decisions, which primarily results from incomplete knowledge about the conditions that determine the consequences of alternative decisions, limited capabilities to a priori assess future events, and limited capabilities to consider all decision alternatives simultaneously. Based on that, this theory posits that the formal structure of organisations allows for assuring rational decision-making by providing an instrument to reduce the organisational complexity and uncertainty. For reducing the complexity and uncertainty, the following five organisational instruments are proposed (Bea and Göbel 2010, pp. 113-125):

Division of labour fosters specialisation as it decomposes organisational actions into specialised actions to restrict the decision-making authorities of the organisational members. This is because decomposed organisational actions reduce the number of potential decision alternatives and resultant decision consequences.

Standardised procedures and programmes determine the operations that organisational members need to perform when carrying out specific organisational actions. Thus, organisational members are able to make routine decisions with known consequences instead of assessing decision alternatives and their consequences at each time.

Hierarchy depicts super- and subordination of the organisational members to restrict the decision-making authorities in a similar way as division of labour. Hierarchical organisations also contribute to reducing the complexity and uncertainty for the members of the organisations.

Communication induces filtering and condensing the information that is relevant for organisational members to fulfil their decision-making authorities. Similar to hierarchy, communication fosters the reduction of organisational complexity and uncertainty.

Indoctrination concerns the loyalty and identification of the organisational members with the organisation. Because subordinate members have privileged information about specific areas of decision-making, it is crucial to ensure that decisions are made on a common basis according to the organisational goals.

The Behavioural Decision Theory considers organisations as decision-making systems with a need to coordinate the decisions of organisational members based on the organisational goals.

Due to the assumption of bounded rationality on the decision-making capabilities, this theory relies on the formal organisational structure as an instrument to reduce the complexity and uncertainty inherent to organisations. In contrast to considering the organisational structure and organisational actions, Behavioural Decision Theory falls short with regard to the organisational goals and does not cover organisational influence factors.

3.2.2.5 Contingency Approach

The Contingency Approach draws upon the Theory of Bureaucracy and Management Theory. In the 1960s, Burns and Stalker (1961), Litwak (1961), and Pugh et al. (1963) introduce the Contingency Approach, whereas Pugh and Hickson (1971; 1976), Pugh (1981), Kieser and Kubicek (1992), Donaldson (2001), as well as Kieser and Walgenbach (2007) have contributed to its further development. Whereas the English literature naturalises the term Contingency Approach, the German literature refers to the terms “Situativer Ansatz” and “Kontingenzansatz” (Kieser 2006, p. 217). In the following, the term Contingency Approach is used for reasons of a clarity.

The Contingency Approach studies organisations with a particular interest in their formal structure. This formal structure characterises the organisational structure independent of the competences and responsibilities of specific organisational roles. Against this background, the Contingency Approach relies on two basic assumptions: there is a lack of a universally efficient formal organisational structure and devising the formal structure of organisations contributes to efficiently achieving the organisational goals. That is, accomplishing the organisational goals requires a flexible adaptation of the formal organisational structure to specific situations, i.e. organisational influence factors. Thereby, the formal organisational structure coordinates the organisational actions and roles with regard to the organisational goals while considering the organisational influence factors. As such, the Contingency Approach considers the formal structure of an organisation in terms of variables as opposed to the Theory of Bureaucracy. For instance, organisations with a different size (e.g. small, medium, large) or organisations with different business environments (e.g. dynamic, static) respectively require different formal organisational structures to accomplish their organisational goals efficiently (Kieser and Kubicek 1992; Kieser and Walgenbach 2007):

The Contingency Approach needs to conceptualise and operationalise not only the constructs of the organisational goals, the formal organisational structure, the organisational actions and roles, as well as the organisational influence factors but also the relationships and dependencies between them (Kieser and Kubicek 1992, pp. 26-28, 45-67; Kieser and Walgenbach 2007, pp. 43-46).

Generally, the Contingency Approach characterises the formal structure of an organisation in terms of division of labour and specialisation, coordination, configuration, delegation of decision-making competencies, and documentation. These characteristics are variable and independent of the organisational roles. For instance, the formal organisational structure might incorporate a lower or higher degree of division of labour, coordination, configuration, delegation of decision-making competencies, and documentation (Kieser and Kubicek 1992, pp. 63-167; Kieser and Walgenbach 2007, pp. 77-177). It distinguishes between four coordination mechanisms, which correspond to personal directives, self-coordination, programming, and planning (Kieser and Kubicek 1992, pp. 7-117; Kieser and Walgenbach 2007, pp. 77-122).

In addition, the Contingency Approach highlights organisational influence factors because they determine the situation of organisations. These influence factors can be categorised as internal and external influence factors. Internal influence factors concern the internal situation of organisations in terms of its size, legal form, manufacturing technology, and Information Technology. External influence factors consider the external situation, i.e. the business environment of organisations with regard to global influence factors (e.g. socio-cultural conditions) and task-specific influence factors (e.g. competitors, customers) (Kieser and Kubicek 1992, pp. 199-225; Kieser and Walgenbach 2007, pp. 207-230).

The Contingency Approach studies the formal structure of organisations. This approach assumes a lack of a universally efficient formal organisational structure and that devising the formal structure of organisations contributes to efficiently accomplishing the organisational goals. In this context, the formal organisational structure coordinates the organisational actions and roles in accordance to the organisational goals while considering the organisational influence factors. For these reasons, the Contingency Approach covers the organisational goals, the formal organisational structure, the organisational actions and roles, as well as organisational influence factors.

3.2.2.6 Additional Organisation Theories

In contrast to the above theories, the Market-based View and Resource-based View are discussed together because they centre on the industry sector and corporate resources respectively.

The Market-based View (Theory of Industrial Organisation) studies the advantageously positioning of organisations in their competitive business environment from the perspective of an industry sector. The industry sector, for instance, with regard to its size, number of competitors, and product-market combinations, represents the subject of interest. That is, the Market-Based View takes an external (outside-in) viewpoint, which enables the analysis of risks and opportunities inherent to the business environment for characterising the organisational competiveness. Particularly, the Market-based View concentrates on the market, i.e. the organisational business environment, which includes external information such as customers and competitors. Thereby, it emphasises the role of the structure of an industry sector for the success and performance of organisations. While the Market-based View focuses on organisational influence factors, however, it falls short in considering further characteristics of organisations with regard to their goals, structure, as well as actions and roles (Caves 1980; Caves and Porter 1977; Porter 1981).

The Resource-based View focuses on the strengths and weaknesses of organisations within an industry sector by taking an internal (inside-out) perspective. It considers the role and characteristics of specific organisational resources as the central determinants of organisational success and performance. By relating the (strategic) competitive advantages of an organisation to its resources, the Resource-based View addresses the core competencies of an organisation. Based on that, its line of argumentation states that competitive organisations exploit their resources better than non-competitive organisations. Correspondingly, organisations should pursue the creation of a unique selling proposition based on their core competencies to achieve increasing corporate profits. However, the Resource-based View not only falls short in considering organisational resources that are critical to organisational success and performance but also neglects the impact of influence factors (Barney 1991a, 1991b; Prahalad and Hamel 1990; Wernefelt 1984).

3.2.3 Summary and Implications

Table 13 summarises the results from reviewing the identified Organisation Theories with regard to their capabilities to capture and represent the key characteristics of the underpinning example of transport chains.

	Characteristics of Transport Chains			
Organisation Theory	**Goals**	**Structure**	**Actions and Roles**	**Influence Factors**
Theory of Bureaucracy	Considered	Considered	Not considered	Not considered
Management Theory	Considered	Considered	Not considered	Not considered
Human-Relations Approach	Considered	Not considered	Considered	Not considered
Behavioural Decision Theory	Not considered	Considered	Considered	Not considered
Contingency Approach	Considered	Considered	Considered	Considered
Market-based View	Not considered	Not considered	Not considered	Considered
Resource-based View	Considered	Considered	Considered	Not considered

Table 13: Organisation Theories and the Characteristics of Transport Chains

These results indicate that the Contingency Approach allows for capturing the main constructs, relationships, and dependencies of the domain of transport chains. As such, it could provide a model in terms of a partial knowledge models that serves as the basis for theory-based knowledge acquisition.

3.3 Model Selection

3.3.1 The Analytic Model

The analytic model of the Contingency Approach aims at describing and explaining organisations. The focus is on providing answers to why-questions that concern organisational issues and phenomena by means of empiricism. For instance, corresponding questions to be asked can have a form like “why does the formal structure of organisations vary across different organisations” or “why do organisational roles result in a different behaviour in terms of their organisational actions within and across different organisations”. Corresponding answers represent empirical-cognitive statements or theories, which are suggested to provide explanations and empirical evidence (Kieser and Kubicek 1992, p. 56).

Against that background, the analytic model as depicted in Figure 27 studies the organisational influence factors as the independent variable and the formal organisational structure as the dependent variable. The influence factors, which reflect the situation of an organisation, e.g. in terms of its business environment, lack a priori specification. They comprise the number of factors that contribute to an empirical explanation of differences in the formal structures of organisations. That is, not only the definition of the influence factors but also the main constructs, relationships, and dependencies (black box) in terms of the causal mechanisms between the influence factors and the formal structure remain vague and demand for further specification (Kieser and Kubicek 1992, pp. 56-57).

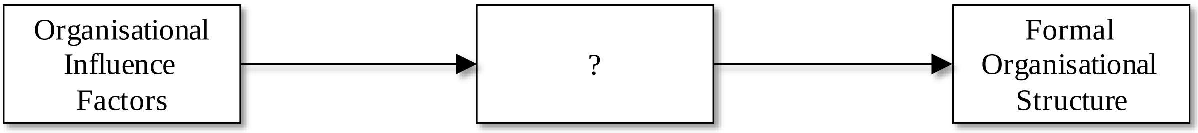

Figure 27: Analytic Model (Kieser and Kubicek 1992, p. 57)

For alleviating this vagueness, an extension of this model (Figure 28) takes into account the behaviour of the organisational members, i.e. the organisational actions and roles, as well as organisational efficiency in terms of efficiently accomplishing the organisational goals (Kieser and Kubicek 1992, p. 57).

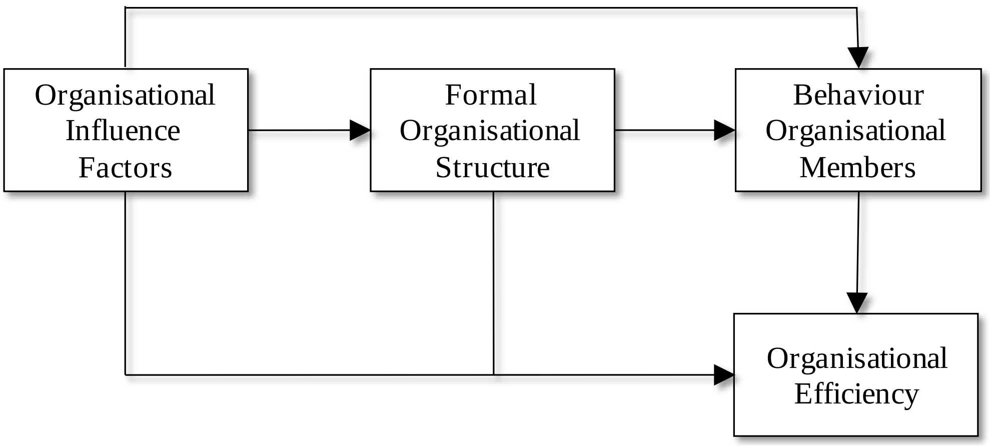

Figure 28: Extended Analytic Model (Kieser and Kubicek 1992, p. 57)

The Aston-Group contributed to the further development of the Contingency Approach during the 1970s and 1980s (Pugh and Hickson 1976; Pugh 1981) through introducing a more comprehensive model (Figure 29).

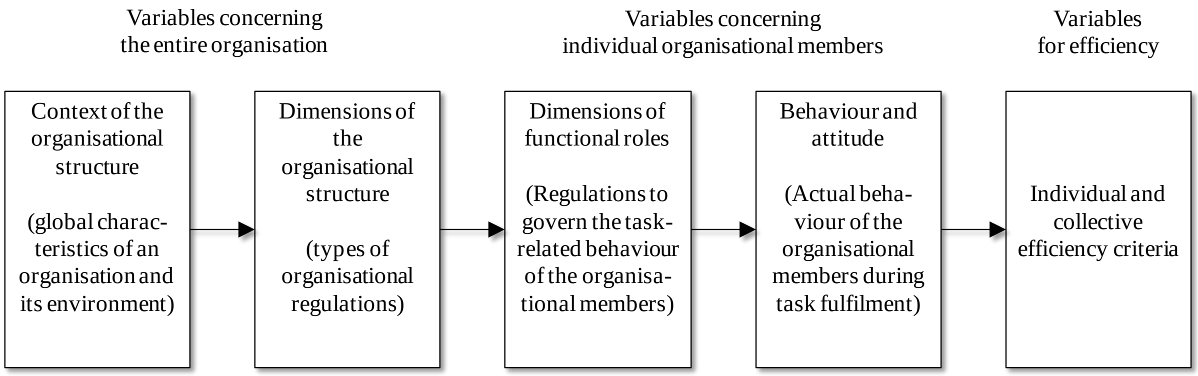

Figure 29: Extended Analytic Model by the Aston-Group (Kieser and Kubicek 1992, p. 58)

This extension distinguishes between five categories, which respectively concern different organisational properties. Each category includes the definition of multiple variants and an operationalisation of the corresponding variables. The rationale behind this model denotes the concept of gradual connections between the different categories. The influence factors provide the context of the organisational structure. As a result, they affect the formal organisational structure. This formal structure concerns the regulations that govern the task-related behaviour of the organisational members, i.e. the predefined organisational actions and roles. These roles have an effect on the actually performed organisational actions, which affect achieving the goals of the organisation with regard to the individual and collective efficiency (Kieser and Kubicek 1992, pp. 57-59).

This branch of the Contingency Approach comprises three different analytic models: the analytic model, the extended analytic model, and the extended analytic model by the Aston-Group. Each of these models respectively refines its predecessor. Nonetheless, all of them pursue the identification of influence factors, which correlate with specific characteristics of the formal structure of different organisations. The common objective is to describe and explain deviations of the formal structure of organisations with regard to various factors influencing the organisation on an empirical basis.

3.3.2 The Pragmatic Model

The purpose of the pragmatic model of the Contingency Approach is to describe and understand organisations to provide a basis for devising them, i.e. organisational design. As such, it provides answers to how-questions that deal with the design of organisations. For instance, corresponding questions concern issues like “how to devise the formal structure of organisations in response to the challenges that arise from several specific influence factors” or “how to devise the formal structure of organisations for defining the organisational roles for efficiently performing the organisational actions” (Kieser and Kubicek 1992, p. 56).

In this context, the subject of the pragmatic model as depicted in Figure 30 is the design of the efficient, formal structure of organisations to accomplish the specific organisational goals. Based on these goals, the formal organisational structure serves as an instrument in terms of an action parameter that allows for governing and coordinating the organisational actions and roles. The organisational influence factors correspond to preconditions and constraints, which restrict the solution space of devising the formal structure of organisations. In addition, they directly and indirectly affect the definition of the organisational actions and roles (Kieser and Kubicek 1992, pp. 59-60).

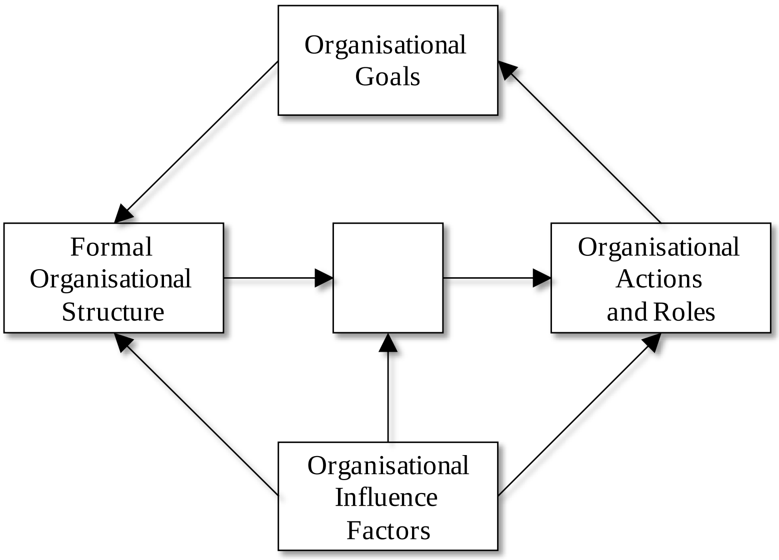

Figure 30: Pragmatic Model (Kieser and Kubicek 1992, p. 60)

Subsequently, the main constructs, relationships, and dependencies of the pragmatic model are presented (Kieser and Kubicek 1992, pp. 60-63):

1. The *organisational goals* constitute the starting point for devising the formal organisational structure. These goals directly affect the formal structure and result from performing the organisational actions.

2. The *formal organisational structure* governs and coordinates the organisational actions and roles. It provides action parameters, which cover not only division of labour and coordination but also further regulations to align the organisational actions and roles with the organisational goals.

3. The *organisational influence factors* directly and indirectly affect the formal structure of organisations as well as the organisational actions and roles.

4. The envisaged (not actual) *organisational actions and roles* result from the combined effects of both the formal structure and the influence factors. This *structure-situation combination* constitutes the key characteristic of the pragmatic model of the Contingency Approach.

5. The design of the formal organisational structure aims at governing and coordinating the organisational actions and roles for accomplishing the organisational goals by taking into consideration the direct and indirect effects of the organisational influence factors.

6. The pragmatic model assumes that a deviation between the envisaged and actual organisational actions results from an insufficient accuracy of fit between the formal

organisational structure and the organisational influence factors. For alleviating this insufficiency, there is a need to adapt the formal structure with regard to its influence factors.

Multiple Organisation Theories, e.g. Eckhardt (1979), Mintzberg (1992), Hill et al. (1994a, 1994b), and Grochla (1995), adopted the pragmatic model since its main constructs, relationships, and dependencies allow for extension and refinement (Kieser and Kubicek 1992, p. 62). For instance, the pragmatic model is widely adopted in Organisation Theory literature (cf. Bea and Göbel 2010, pp. 11-30, 253-411; Kieser and Kubicek 1992, pp. 73-199, 449-480; Schreyögg 2008, pp. 87-129, 251-270; Schulte-Zurhausen 2010, pp. 33-47, 151-254). However, adopting the pragmatic model presupposes to consider the complementarities of the purpose, scope, and underlying assumptions of the Contingency Approach and further Organisation Theories (Kieser and Walgenbach 2007, p. 46).

3.3.3 Summary and Implications

The analytic and the pragmatic branch of the Contingency Approach attend to similar research methods for empirical data collection and statistical evaluation but concentrate on substantially different organisational issues and phenomena (i.e. answering why-questions in contrast to how-questions). Consequently, the two branches differently interpret and draw conclusions from empirical data. For instance, the analytic branch primarily aims at refining the methods for collection and statistical evaluation of empirical data whereas the pragmatic branch mainly focuses on interpreting and drawing conclusion from these results for reasons of designing organisations. Considering this difference and the key characteristics of transport chains, the pragmatic model provides the required basis for theory-based knowledge acquisition in the form of a partial knowledge model.

Further confirming the need for the pragmatic model requires a more detailed comparison with the analytic models of the Contingency Approach.

The analytic model vaguely specifies the main constructs, i.e. influence factors and formal structure, as well as rudimentarily considers the relationships and dependencies between these constructs. In contrast, the extended analytic model adds two constructs, i.e. the organisational actions and roles as well as organisational goals, but still lacks a sufficient specification predominantly with regard to the relationships and dependencies between them. The model proposed by the Aston-Group covers three constructs, i.e. the influence factors, formal structure, as well as the organisational actions and roles. However, it falls short in defining the corresponding relationships and dependencies. In this context, it is worth pointing out that, for studying organisations, these three models consider the influence factors of an organisation as their starting point.

On the contrary, the pragmatic model not only specifies four constructs, i.e. the organisational goals, the formal organisational structure, the organisational actions and roles, as well as the influence factors but also defines multiple direct and indirect relationships and dependencies between these constructs. Compared to the three analytic models, the organisational goals represent the starting point for devising the formal organisational structure, whereas the influence factors represent preconditions and constraints for organisational design.

Based on this comparison, there is a further need for relating the pragmatic model, i.e. main constructs, relationships, dependencies, and underlying assumptions with the key characteristics of transport chains. Since abstraction and generalisation is therefore required, it is objectively not possible to assess the pragmatic model as a partial knowledge model for the domain of transport chains precisely and comprehensively. In contrast, a precise and comprehensive comparison is neither necessary nor expedient for the research objective of this thesis. Instead, the plausibility of using the pragmatic model with regard to transport chains and potential contradictions (inconsistencies) should be assessed. That is, such an assessment focuses on plausibility and consistency, first with regard to the main constructs and then considering the relationships and dependencies.

In general, the pragmatic model assumes that there is a lack of a universally efficient formal organisational structure and devising the formal structure of organisations contributes to achieving the organisational goals. For accomplishing the organisational goals efficiently, organisations need to adapt flexibly their formal organisational structure to their organisational influence factors (e.g. specific business environment) for coordinating the organisational actions and roles as well as aligning them with the organisational goals. These influence factors correspond to preconditions and constraints, which restrict the solution space for designing organisations. Transferring these assumptions to transport chains leads to the following result: Transport chains lack a universally efficient formal organisational structure because they are subject to various influence factors, which result from operating in dynamic and complex business environments. Devising their formal structure contributes to accomplishing the transport chain goals. For achieving these goals efficiently, transport chains need to adapt flexibly their formal structure to their specific influence factors (e.g. changing customer demands) for coordinating the transport chain actions and roles as well as aligning them with the transport chain goals. Similarly, these influence factors correspond to preconditions and constraints, which restrict the solution space for designing transport chains.

Based on that, Table 14 enumerates the main constructs of the pragmatic model and contrasts them with the key characteristics of transport chains to demonstrate plausibility and consistency.

Constructs of the Pragmatic Model	Characteristics of Transport Chains
Organisational Goals	Transport Chain Goals
Organisational Structure	Transport Chain Structure
Organisational Actions and Roles	Transport Chain Actions and Roles
Organisational Influence Factors	Transport Chain Influence Factors

Table 14: Pragmatic Model and Transport Chains

Subsequently, the focus is on the relationships and dependencies between these constructs. Therefore, the pragmatic model is adopted with reference to transport chains (Figure 31).

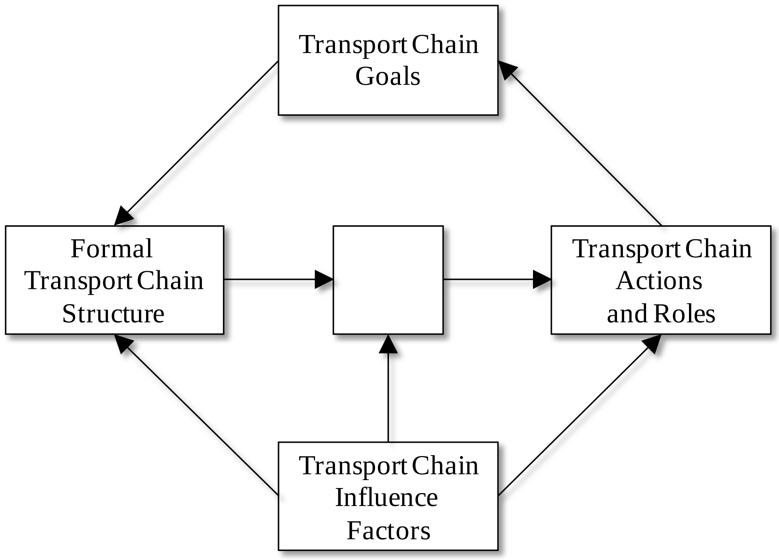

Figure 31: Pragmatic Transport Chain Model

Figure 31 presents the pragmatic transport chain model and points out the relationships and dependencies:

1. The *transport chain goals* constitute the starting point for devising the formal transport chain structure. These goals directly affect the formal transport chain structure and result from performing the transport chain actions.
2. The *formal transport chain structure* governs and coordinates the transport chain actions and roles. The formal structure provides action parameters, which cover not only division of

labour and coordination but also further regulations to align the transport chain actions and roles with the transport chain goals.

3. The *transport chain influence factors* exert a direct and indirect influence on the transport chain actions and roles as well as on the formal structure of transport chains.
4. The envisaged (not actual) *transport chain actions* result from the combined effects of both the formal structure and the influence factors, i.e. *structure-situation combination*.
5. The design of the formal transport chain structure aims at governing and coordinating the transport chain actions and roles for accomplishing the transport chain goals by considering the direct and indirect effects of the transport chain influence factors.
6. The pragmatic transport chain model assumes that a deviation between the envisaged and actual transport chain actions results from an insufficient accuracy of fit between the formal transport chain structure and the transport chain influence factors. For alleviating this insufficiency, there is a need to adapt the formal structure with regard to the influence factors.

The above statements demonstrate the plausibility and consistency of using the pragmatic model as a partial knowledge model for theory-based knowledge acquisition with reference to the domain of transport chains. Extending and further detailing this inquiry is considered neither necessary nor expedient as Organisation Theory is frequently applied to describe, explain, predict, and design different phenomena and issues of transport chains (Section 3.2.1).

Using the pragmatic model of the Contingency Approach (hereafter denoted as *model*) as a partial knowledge model for theory-based knowledge acquisition requires the deduction of corresponding requirements.

3.4 Requirements Deduction

Requirements deduction is constituent for design science research. It means to draw consistent, coherent, and intersubjective repeatable conclusions from the knowledge base. As such, it can rely on informal (argumentative based on natural language), semi-formal (conceptual), or formal (mathematical) means (Gregor 2006; Gregor and Jones 2007; Gregory and Muntermann 2011; Haynes and Carroll 2010; Hevner et al. 2004).

To design a method for theory-based knowledge acquisition, deduction makes use of a semi-formal approach, which relies on the respective constructs, relationships, and dependencies of the pragmatic model of the Contingency Approach and the model of transport chains. If required, deduction additionally uses supplementing literature, e.g. Hill et al. (1994a, 1994b) and Grochla (1995).

3.4.1 Organisational and Transport Chain Goals

Organisational and transport chain goals are of major significance as they provide the reference point for devising the formal structure, which serves the coordination of the actions and roles. Organisational and transport chain goals unify multiple individual goals of their members and, thus, represent a bundle of common and shared goals. They represent economic issues and reflect political, social, legal, and ecological interests. There are two types of goals: material goals and economic goals. Material goals address the creation of economic value, i.e. the production of material goods and the provision of services. Economic goals focus on monetary values such as profit, return, and turnover. Between economic and material goals are hierarchical means-end relationships. These relationships indicate that accomplishing economic goals requires achieving material goals. That is, economic value is created. Further, organisational and transport chain goals are permanent, which indicates their independence from individual economic actors. It includes the fact that they can change over time, which is rather typical for dynamic organisational environments.

Requirements deduction from the organisational and transport chain goals results in different types of goals as well as different types of relationships and dependencies (e.g. means-end relationships). Since goals are common and shared in organisations and transport chains, the process of accomplishing the goals needs to consider different types of goal conflicts and complementarities. Moreover, goals maintain various types of relationships and dependencies to the structure, actions and roles, as well as influence factors. Based on that, the following four deductions can be made with regard to various types of:

- goals, e.g. material goals, economic goals.
- relationships between goals, e.g. hierarchical, means-end.
- goal conflicts and complementarities, e.g. temporal, factually logical.
- relationships and dependencies to the structure, actions and roles, as well as influence factors, e.g. direct, indirect, reverse.

3.4.2 Organisational and Transport Chain Structure

Organisational and transport chain structures focus on the formal structure as an instrument to define the actions and roles and coordinate with regard to goal achievement. Therefore, it provides different types of regulations. These regulations deal with the division of labour through analysis and synthesis of actions and roles as well as their purposeful coordination by means of corresponding mechanisms (e.g. personal directives, planning). The formal structure of organisations and transport chains distinguishes between the structural and process organisation. The structural organisation provides a static viewpoint on organisations. Thus, it concerns the definition, distribution, and allocation of responsibilities and competences in terms of roles. For instance, organisational charts represent the structural organisation. On the contrary, the process organisation takes a dynamic viewpoint on organisations. As such, it focuses on the realisation of responsibilities and competences by organisational actions. For instance, flow charts depict the process organisation.

Deducing requirements from the organisational and transport chain structure needs to consider the various types of the structural and process organisation as well as the corresponding multiplicity of diverse relationships and dependencies. These relationships and dependencies additionally concern the goals, actions and roles, as well as influence factors. Consequently, there are four deductions in the form of various types of:

- the structural organisation, e.g. virtual, strategic, operational.
- the process organisation, e.g. convergent, divergent.
- relationships and dependencies with the goals, actions and roles, as well as influence factors, e.g. direct, indirect, reverse.

3.4.3 Organisational and Transport Chain Actions and Roles

Accomplishing the organisational goals requires performing organisational and transport chain actions through the organisational members as defined by the corresponding roles. Based on a utilitarian understanding (Etzioni 1961), in which organisations refer to commercial companies and authorities in contrast to other organisational types such as prisons, hospitals, churches, or associations, organisational membership allows for determining the borders of organisations and transport chains. Accordingly, membership corresponds to the calculated engagement of the individual members and the exercise of remunerative power, i.e. governance based on material rewards. The relationship between calculated engagement and exercise of remunerative power is regulated by (employment) contracts, which are based on roles (responsibilities and competences) as abstract definitions of actions. This understanding induces that economic actors can participate in more than one organisation and transport chain simultaneously. Further, fulfilling roles accounts for actions, which vary in their type, degree of complexity and dynamicity, as well as their constitution in the form of atomic and composite actions. These kinds of actions directly and indirectly, i.e. in terms of primary and secondary actions, contribute to goal achievement.

Against this background, deducing requirements needs to take into account the purpose of actions in terms of primary and secondary actions as these two types differently contribute to goal achievement. These types of actions similarly allow for various degrees of decomposition and composition, which enables different potential types and combinations of actions with regard to their constitution. Performing such actions is based on specific roles, i.e. abstract definitions of responsibilities and competences, so that there are multiple types of roles according to the purpose and constitution of the actions. Additionally, actions and roles maintain various types of relationships and dependencies to the goals, structure, and influence factors. As a result, the requirements deduction of organisational and transport chain actions and roles leads to various types of:

- actions based on their purpose, e.g. primary actions, secondary actions.
- actions based on their constitution, e.g. atomic actions, composite actions.
- roles based on the purpose of actions, e.g. roles in operations or management.
- roles based on the constitution of actions, e.g. machine control or operations management.
- relationships and dependencies to the goals, structure, and influence factors, e.g. direct, indirect, reverse.

3.4.4 Organisational and Transport Influence Factors

Organisational and transport chain influence factors comprise internal and external factors. Internal factors distinguish between historical and present influences whereas external factors focus on task-specific and global influences. Accordingly, there are different types of influence factors based on their viewpoint (internal and external) and source (present, historical, global, and task-specific). These influences have various types of relationships and dependencies to the goals, structure, as well as actions and roles. Thus, the following deductions can be made in terms of various types of:

- influence factors based on their viewpoint, e.g. internal, external.
- influence factors according based on their source, e.g. present, historical, global, task-specific.

– relationships and dependencies to the goals, structure, as well as actions and roles, e.g. direct, indirect reverse.

3.4.5 Summary and Implications

Table 15 summarises the results of the requirements deduction from the model of the Contingency Approach and the transport chain model as depicted in Section 3.3.2 and 3.3.3 respectively.

Constructs of Model	**Requirements Deduction**
Organisational and Transport Chain Goals	– goals, e.g. material goals, economic goals – relationships between goals, e.g. hierarchical, means-end – goal conflicts and complementarities, e.g. temporal, factually logical – relationships and dependencies to the structure, actions and roles, as well as influence factors, e.g. direct, indirect, reverse
Organisational and Transport Chain Structure	– the structural organisation, e.g. virtual, strategic, operational – the process organisation, e.g. convergent, divergent – relationships and dependencies with the actions and roles, and influence factors, and goals, e.g. direct, indirect, reverse
Organisational and Transport Chain Actions and Roles	– actions based on their purpose, e.g. primary actions, secondary actions – actions based on their constitution, e.g. atomic actions, composite actions – roles based on the purpose of actions, e.g. roles in operations or management – roles based on the constitution of actions, e.g. machine control or operations management – relationships and dependencies to the goals, structure, and influence factors, e.g. direct, indirect, reverse
Organisational and Transport Influence Factors	– influence factors based on their viewpoint, e.g. internal, external – influence factors according based on their source, e.g. present, historical, global, task-specific – relationships and dependencies to the goals, structure, as well as actions and roles, e.g. direct, indirect, reverse

Table 15: Results from Requirements Deduction

The deduction result comprises 15 statements, which represent the functional requirements for designing the method for theory-based knowledge acquisition.

4 Design

This chapter describes the design of the method for theory-based knowledge acquisition. First, the principles of method engineering are defined to guide the design. Second, the underlying assumptions and requirements for the design are discussed. Third, the method in form of its metamodels, activity model, outcomes, roles, and techniques is specified.

4.1 Approach

Method Engineering is the field that studies the development of methods for information systems. It can be defined similarly to the IEEE definition of Software Engineering, which indicates that Method Engineering concerns all activities associated with the development of methods (Brinkkemper 1996, pp. 275-277):

> *Method Engineering designs, constructs, and adapts methods for the development of (parts of) information systems.*

Method Engineering represents a form of design science research. The literature provides several approaches for constructing methods as design artefacts. These approaches commonly assume that there is no universally applicable method being able to solve all potentially occurring problems. Thus, problem-solving demands the tailoring of existing or construction of new methods to fulfil the specific requirements of the problem domain (Brinkkemper 1996, pp. 276-279). Four general approaches can be identified (Hendersson-Sellers and Ralyté 2010, pp. 443-447; Ralyté et al. 2004, pp. 203-204):

The *assembly-based approach* aims at reusing method components, i.e. atomic elements of methods to construct new methods. It assumes that method components are separately available from existing methods, include a meaningful description, and are stored in a method repository. Then, it is possible to select components from the repository and use predefined rules to assemble a new method that satisfies the requirements of the problem domain. This approach distinguishes two types of assembling method components (Brinkkemper et al. 1998). *Association* concerns the assembly of method components with different purposes, whereas *integration* considers overlapping method components that share the same or similar purposes but provide different means to satisfy the respective requirements.

The *paradigm-based approach* relies on a metamodel that originates from a specific theoretical framework (Ralyté et al. 2003). This metamodel corresponds to an as-is model, which is subject of instantiation, generalisation, or adaptation for developing it towards a to-be model. This to-be model is suggested to fulfil the specific characteristics of the problem domain (Gupta and Prakash 2001). For instance, in the case of adaptation, the as-is model and to-be model are at the same level of abstraction, whereas in cases of specialisation and generalisation, the as-is and to-be models pertain to different levels of abstraction.

The *extension-based approach* focuses on enhancing methods with new characteristics, which allow for better meeting the specific requirements of the problem domain. Therefore, this approach builds on existing methods and adds novel features to them. For instance, Baresi et al. (2001) and Gehlert et al. (2004) provide examples of the extension-based approach.

The *ad-hoc approach* aims at developing novel methods from scratch with neither assembling existing method components, drawing upon metamodels, nor extending existing methods (Ralyté et al. 2004). Two reasons motivate this approach for the development of new methods: (specific) methods do not sufficiently cover a new or existing problem domain and the characteristics of the problem domain significantly differ from former ones. In this context, the term ad-hoc needs to be understood in terms of suited to purpose or tailored to the problem domain (Glass 2000, pp. 127-128).

Method Engineering proposes three out of four approaches that require reusable and available method components, metamodels, and entire methods. Based on the findings of the review in Section 2.3, existing knowledge acquisition methods in ontology development do not fulfil these requirements. Further, it is unclear how to solely construct method components, how to find relevant parts of methods (e.g. method components, metamodels), and how to combine both of them for developing useful methods. These obstacles indicate that only the fourth approach, i.e. the ad-hoc approach provides adequate capabilities to guide the development of a method for theory-based knowledge acquisition in ontology development.

Method Engineering suggests three main steps as shown (Figure 32), which can be taken to design the knowledge acquisition method as follows.

Figure 32: Process Model for Method Design

Requirements Definition and Assumptions (Chapter 4.2) makes explicit the underlying assumptions and defines the requirements for method development. These requirements centre on design, functionality, and method components. The design-related requirements reflect the basic characteristics of the knowledge acquisition method, whereas the functional requirements incorporate the results from requirements deductions. The component-related requirements define the method components, which constitute the knowledge acquisition method.

Method Development (Chapter 4.3) specifies the knowledge acquisition method. The specification consists of five constituent components, i.e. the method and outcome metamodel, activity model, outcomes, roles, and techniques.

Method Evaluation (Chapter 5) evaluates the method for theory-based knowledge-acquisition. It comprises a criteria-based evaluation with regard to design science, design-related, functional, and component-related requirements as well as a scenario-based evaluation with an example of intermodal transport chains.

4.2 Requirements Definition and Assumptions

4.2.1 Assumptions

Designing the knowledge acquisition method requires making several assumptions. These assumptions primarily result from the fact that the theory-based knowledge acquisition method constitutes an integral part of ontology development. For instance, the method should be compatible with existing approaches for developing ontologies. Based on that, method development draws upon the following assumptions:

– Knowledge acquisition is based on a constructivist understanding. It consists of the identification (and elicitation) of data, the interpretation of this data (information), and the structuring and interlinking of this information (knowledge). Consequently, knowledge acquisition methods represent a systematic and intersubjective repeatable process based on certain principles to acquire knowledge for developing ontologies.

– Based on the classification of knowledge acquisition in ontology development, there is a preceding activity of ontology specification, which defines the purpose and scope of the ontology, as well as a succeeding activity of formalisation and/or implementation, which makes the ontology processible for their target application. Ontology development tools that support knowledge acquisition (e.g. Protégé, NeOn Toolkit (NeOn 2012)) typically integrate formalisation and/or implementation.

– The knowledge acquisition method does not focus on selecting knowledge sources (e.g. textbooks, technical articles) and knowledge acquisition techniques (e.g. text analysis, interview). This scope is because of the dependence of knowledge sources and techniques on the particular domain of interest, the purpose and scope of the ontology, as well as the characteristics of the ontology development project (e.g. availability of domain experts). Therefore, the method deals with the use of knowledge sources and techniques on a high level of abstraction.

- Developing the knowledge acquisition method reuses tools for supporting knowledge acquisition. This is due to the multiplicity of widely adopted tools (e.g. Protégé, NeOn Toolkit), which are subject of continuous further development. Despite, method design is independent from the characteristics of specific tools so that the use of adequate tools is possible.

4.2.2 Requirements Definition

4.2.2.1 Design-related Requirements

The design-related requirements reflect the basic characteristics for method design. These requirements comprise general issues of method design and more specific issues of knowledge acquisitions methods.

General design requirements consider common issues, which are typically generic in their nature and, thus, applicable to a larger set of methods. Among others, modelling principles proposed by Becker et al. (1995) and their further development by Schütte (1997; 1998, p. 112) are adopted. Accordingly, the following general design-releated requirements should be fulfilled (Table 16).

	Description
rd1	*Minimalism* requires that method design is focused on the relevant facts with regard to the method's purpose.
rd2	*Intra-method relationships* requires that all relevant relationships between the activities of a method are represented.
rd3	*Inter-method relationships* requires that all relevant relationships between the constituent components of a method are represented.
rd4	*Language adequacy* requires that the description of the method (method metamodel) and its outcomes (outcome metamodel) is covered.
rd5	*Syntactical correctness* requires that the method description with regard to its underlying metamodel is adequate.
rd6	*Clarity* requires that the understandability in terms of decomposition (e.g. decomposing activities into sub-activities) and readability in terms of layout design (e.g. graphical representation) is ensured.
rd7	*Efficiency* requires that the usefulness of the method for creating a benefit is given. For instance, carrying out knowledge acquisition with equal quality of the results in less time or carrying out knowledge acquisition in the same time with higher quality results.

Table 16: General Requirements

Specific design requirements deal with particular issues of knowledge acquisition methods. Based on Freiling et al. (1985, p. 152) as well as Gruber and Cohen (1987, pp. 144-146), method design should meet the subsequent specific design-related requirements (Table 17).

	Description
rd8	Integration of a gradual approach for knowledge acquisition.
rd9	Capture of the key concepts and relationships of the domain of interest.
rd10	Capability for representing the terminology of the domain of interest.
rd11	Provision of tangible results.
rd12	Integration in existing ontology development methods.
rd13	Applicability of various knowledge acquisition techniques.
rd14	Applicability of various knowledge acquisition (ontology development) tools.

Table 17: Specific Requirements

4.2.2.2 Functional Requirements

The functional requirements synthesise the deductions from the basic model of the Contingency Approach and its adoption to the domain of transport chains (Section 3.4). They represent the functionality of the knowledge acquisition method (Table 18).

	Description
rf1	Acquiring the goals.
rf1.1	Considering various types of goals.
rf1.2	Considering various types of relationships between goals.
rf1.3	Considering various types of conflicts between goals.
rf1.4	Considering various types of complementarities between goals.
rf1.5	Considering various types of relationships and dependencies to the structure, actions and roles, as well as influence factors.
rf2	Acquiring the structure.
rf2.1	Considering various types of structural organisation.
rf2.2	Considering various types of process organisation.
rf2.3	Considering various types of relationships and dependencies to the goals, actions and roles, as well as influence factors.
rf3	Acquiring the actions and roles.
rf3.1	Considering various types of actions with regard to their purpose.
rf3.2	Considering various types of actions with regard to their constitution.
rf3.3	Considering various types of roles with regard to the purpose of actions.
rf3.4	Considering various types of roles with regard to the constitution of actions.
rf3.5	Considering various types of relationships and dependencies with the goals, structure, as well as influence factors.
rf4	Acquiring the influence factors.
rf4.1	Considering various types of influence factors according to their viewpoint.
rf4.2	Considering various types of influence factors according to their source.
rf4.3	Considering various types of relationships and dependencies with the goals, structure, as well as actions and roles.

Table 18: Functional Requirements

4.2.2.3 Component-related Requirements

Determining the method components is crucial since literature lacks a uniform and prevailing understanding (Section 2.1.2.1). Correspondingly, pursuing the goal of a method description to be as complete as possible, the method should meet the following component-related requirements (Table 19).

	Description
rc1	Specification of two *metamodels* to define the constructs of the activity model and the outcomes of the knowledge acquisition activities.
rc2	Specification of the *activity model* to define an ordered, coherent, and finite set of knowledge acquisition activities.
rc3	Specification of the *outcomes* of each knowledge acquisition activity.
rc4	Specification of a set of *roles* that carry out the knowledge acquisition activities.
rc5	Specification of a set of *techniques* that support the knowledge acquisition activities.

Table 19: Component-related Requirements

4.3 Method Development

This section reports the proposed method by describing its metamodels, activity model, outcomes, roles, and techniques as discussed in Section 2.1.2.1.

4.3.1 Metamodels

4.3.1.1 Method Metamodel

The method metamodel specifies the constructs, relationships, dependencies, and corresponding consistency conditions of the method. The language used for defining the metamodels is the ERM, which was also used for the description of existing methods in Section 2.3.1 (Chen 1976). Figure 33 presents the metamodel of the knowledge acquisition method.

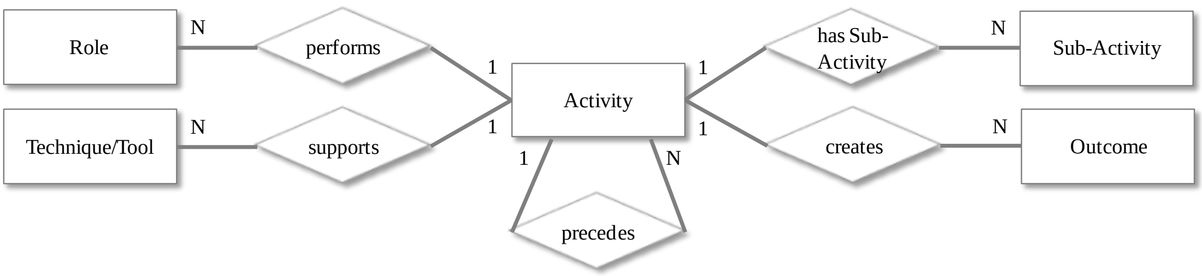

Figure 33: Method Metamodel

4.3.1.2 Outcome Metamodel

The rationale for specifying the outcome metamodelling language originates from the principle of structure-preserving design (Benjamins and Aben 1997; Speel and Aben 1998). According to this principle, the structure and content of the outcomes created within knowledge acquisition should be maintained for ontology development. That is, an ontology language is used to model the outcomes of the knowledge acquisition activities such that subsequent activities of formalisation and/or implementation require as little modifications as possible for constructing the ontology. The specification uses OWL 2 (Full) as the de-facto standard language in ontology development and particularly applies OWL 2 DL.

OWL 2 (Full) represents a comprehensive ontology language to enable precise statements with a formally defined meaning (W3C OWL Working Group 2012). Because its comprehensiveness puts high demands on the ability and skills of ontology developers, the syntactically restricted but computationally decidable variant OWL 2 DL is used (Hitzler et al. 2012, pp. 35-42).

There are semantic differences between OWL 2 (Full) and OWL 2 DL, which should be mentioned. OWL 2 (Full) refers to *RDF-based semantics* (Schneider 2012; W3C OWL Working Group 2012), which extends the semantics defined for the Resource Description Framework Schema (RDF(S)) (Hayes 2004). Thus, OWL 2 ontologies are considered as Resource Description Framework (RDF) graphs so that the meaning is directly assigned to the graph and indirectly to ontology structures. In contrast, OWL 2 DL is associated to the *direct model-theoretic semantics* (Motik et al. 2012), which applies Description Logics (Baader et al. 2003) to directly assign the meaning to ontology structures. These semantics are compatible with the model theoretic semantics of the SROIQ Description Logics. SROIQ Description Logics represents a fragment of First-Order Logic with useful computational properties and corresponds to the knowledge representation paradigm that underpins OWL 2 DL (Horrocks et al. 2006).

The outcome metamodelling language consists of the following language constructs and construction rules:

1. *Entities*: Statements in OWL 2 DL represent objects of the (real-)world and the relations that hold between these objects. They are not all of piece but incorporate an explicitly represented internal structure. The atomic constituents of OWL 2 DL statements, i.e. objects, categories, and relations are termed entities. OWL 2 DL denotes objects as *individuals*, categories as *classes*, and relations as *properties* (Section 2.1.1.2). Such properties further comprise the types of *object properties*, *datatype properties*, and *annotation properties*.

2. *Axioms*: OWL 2 DL ontologies assume that knowledge consists of atomic pieces, which correspond to statements. Statements that compose an OWL 2 DL ontology are axioms, which are asserted to be true for a given state of affairs.

3. *Expressions*: OWL 2 DL allows for combining (names of) entities by the use of constructors into expressions. Expressions represent combinations of entities to form complex descriptions from atomic ones. They represent new entities, which are defined by their structure. The constructors for the different kinds of entities vary greatly. For classes, the expression language is very rich, whereas it is more restrictive for properties.

A detailed description of the OWL DL entities, axioms, and expressions can be found in the appendix 1, 2, and 3.

For serialising an OWL 2 DL ontology, several syntactic formats have been proposed. Table 20 enumerates them and briefly denotes their respective benefits (Hitzler et al. 2012, p. 3).

Syntax	Benefits (short)
RDF/XML (Beckett 2004)	Ease of interchange between OWL 2 tools.
OWL/XML (Hori et al. 2003)	Ease of processing in XML tools.
Functional Syntax (Motik et al. 2009)	Ease of specification.
Manchester Syntax (Horridge and Patel-Schneider 2009)	Ease of reading and writing for non-logicians.
Turtle (Beckett and Berners-Lee 2008)	Ease of reading and writing of RDF triples.

Table 20: OWL 2 Syntactic Formats

OWL 2 Functional Syntax is used for the outcome metamodel. It is both easy to read and understand as well as concise (Vrandecic 2010, pp. 31-32).

A de-facto standard for visualising OWL ontologies is still missing (Katifori et al. 2007). However, dedicated and equally adequate representation formats such as OWL Viz (Horridge 2010) in Protégé and KC-Viz (Motta et al. 2011) in NeOn Toolkit are available. In principle, notations to be applied could build on graph-like or table-type structures. For example, tables could be used to represent *Class–ObjectProperty–Class* structures, whereas extant graphical notations (e.g. UML), which originate from other fields (e.g. Software Engineering), could also be used (Cranefield and Purvis 1999; Wang and Chan 2001).

4.3.2 Activity Model

The activity model specifies the temporally and factually logical sequence of the knowledge acquisition activities and their relationships with the various method components, i.e. outcomes, roles, and techniques. As an instance of the method metamodel, this method consists of four knowledge acquisition activities:

1. Acquisition of the Goals

2. Acquisition of the Structure

3. Acquisition of the Actions and Roles

4. Acquisition of the Influence Factors

Each activity is decomposed into four sub-activities, which deal with identifying data, structuring this data (information), interlinking this information (knowledge), and refining the knowledge.

The activity model also denotes for each activity the outcomes, involved roles, and proposed techniques, if required. Further, it relies on five principles for dealing with the structural and logical complexity of knowledge acquisition:

1. The activity model is *incremental* and *iterative* (Cockburn 2008). That is, the activities incorporate repeating (reworking) cycles (iterative) and gradually acquire knowledge (incremental).
2. The activity model has a *hierarchical organisation*. As such, the activities, their respective sub-activities, and the corresponding outcomes are subject of decomposition.
3. The activity model, i.e. its functionality reflects the deductions from the Contingency Approach. Thus, it implements the functional method requirements for realising *theory-based knowledge acquisition*.
4. The activity model allows for adopting various types of *principles* such as top-down, middle-out, or bottom-up. In the case of the top-down approach, the construction of outcomes starts from an abstract level and increases the level of detail until the outcomes exhibit the envisaged level of granularity.
5. The outcomes have a *modular structure* based on the subject area of the knowledge acquisition activities and the categorisation inherent to the semantic spectrum.

4.3.2.1 Goals Acquisition

The first activity acquires knowledge about goals for developing a respective ontology. Therefore, this activity consists of the following four sub-activities:

1. Identifying Goals
2. Structuring Goals
3. Interlinking Goals
4. Refining Goals

These sub-activities respectively contribute to the creation of the goal ontology by consecutively developing a controlled vocabulary, glossary, thesaurus, and taxonomy as defined in the semantic spectrum. Constructing these outcomes involves the roles of the ontology developer and domain expert as well as the techniques text analysis and interview (Figure 34).

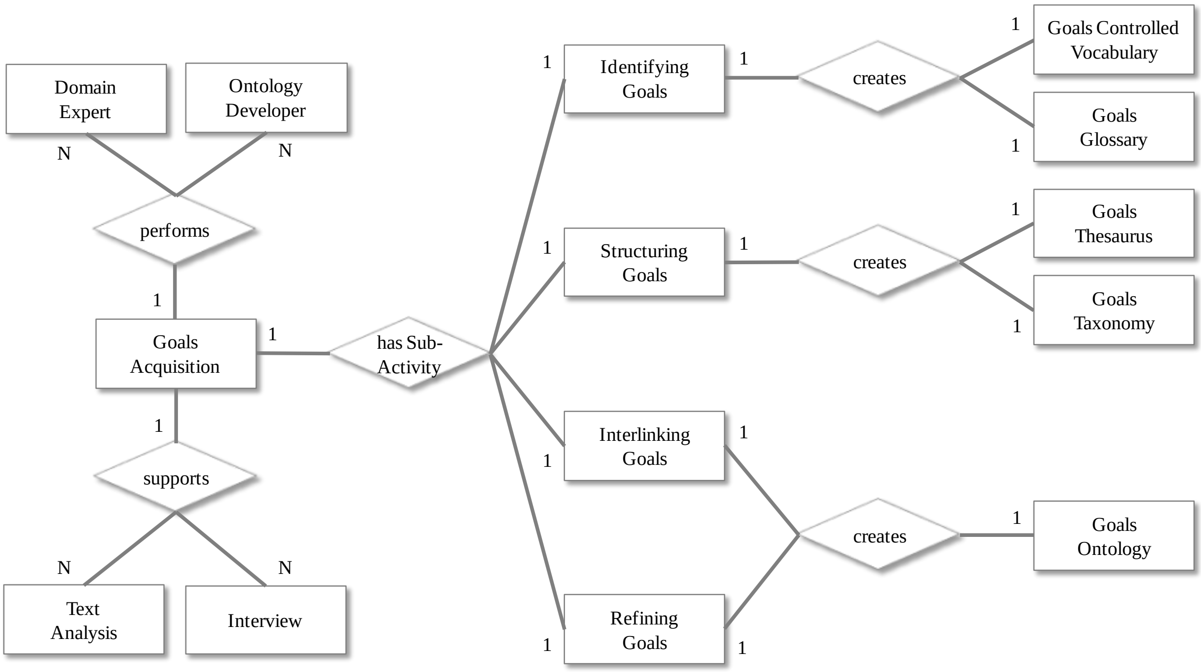

Figure 34: Goals Acquisition

4.3.2.1.1 Identifying Goals

The first sub-activity deals with the identification of goals. It covers both the identification and interpretation since identification implies (human) interpretation with regard to a specific context based on the ontology's scope and purpose. Thereto, the constructs of material and economic goals provide the context. Material goals represent objectives related to value creation (e.g. type, quality, quantity), whereas economic goals consider monetary objectives (e.g. turnover, profit, return). At this point, the means-end relationships and dependencies between material and economic goals are subordinate.

Based on that, this sub-activity aims at constructing two types of outcomes: a controlled vocabulary and a glossary of goals. The controlled vocabulary corresponds to an enumeration, i.e. a finite list of terms of different goals. It is extended towards a glossary by specifying the meaning of the terms through adding natural language definitions. Thereto, it is recommended to reuse and adapt existing controlled vocabularies and glossaries. In the case of existing thesauri, taxonomies, and ontologies about goals, this sub-activity proposes to proceed with the subsequent sub-activities and to postpone the reuse of these knowledge sources.

4.3.2.1.2 Structuring Goals

The second sub-activity focuses on structuring goals. It builds on the previous outcome and adds taxonomic relations for constructing a taxonomy of goals. Therefore, a thesaurus is initially constructed and, based on that, a taxonomy is developed.

For structuring goals with regard to building a thesaurus, it is necessary to capture synonyms, antonyms, broader terms, and narrower terms of the goals. Synonymous and antonymous terms could either merely draw upon the terms in the glossary or demand further identification. Despite synonyms and antonyms can be added continuously, this sub-activity recommends capturing them as early as possible to avoid ambiguities and misunderstandings as potential sources of errors. Capturing broader and narrower terms is supported by making explicit hierarchical and means-end relationships as well as dependencies between material and economic goals. Additionally, classifying or clustering the goals mutually exclusive and collectively exhaustive with regard to more specific goal types (e.g. strategic, tactical, operational) assists in constructing the thesaurus.

Further developing this thesaurus towards a taxonomy requires dealing with generalisation/ specialisation relationships and instantiations by applying subClassOf-/subPropertyOf-relations and typeOf-relations respectively. Introducing subClassOf-/subPropertyOf-relations relies on hierarchical relationships between different goal types, whereas typeOf-relations, for example, allow for representing goals in the form of value partitions and enumerated classes.

4.3.2.1.3 Interlinking Goals

The third sub-activity considers the interlinking of goals by modelling additional relationships and dependencies (non-taxonomic relations) beyond the previously established taxonomic relations. As such, interlinking concerns aspects, which the preceding two sub-activities have not captured yet. The result is an ontology about goals. Its construction particularly requires the acquisition of different types of conflicts and complementarities between goals as well as various types of constraints, which hold on the feasibility of goals.

The underlying taxonomy could either implicitly contain the knowledge for interlinking such that this knowledge merely needs to be made explicit or the construction of the goal ontology requires further knowledge acquisition with regard to the activities of identification and structuring. Developing the taxonomy towards the goal ontology, an increasing use of axioms and expressions based on the specification of the outcome metamodelling language is needed.

4.3.2.1.4 Refining Goals

The fourth sub-activity deals with refining the goal ontology. It mainly concerns coherence in terms of checking whether the knowledge about the goals is logically connected and consistency, which means to check whether the knowledge is afflicted with contradictions. Depending on the specific modelling language, i.e. ontology language and graphical notation, this sub-activity could also cope with syntactic errors in terms of the correct use of ontology language and graphical notation as well as semantic errors, which complies with the correct use of primitives of the ontology language and graphical notation.

Additionally, ontology refinement focuses on establishing unidirectional relationships from the goal ontology to the other ontologies. It is based on the Contingency Approach with regard to the structure, actions and roles, as well influence factors. Temporarily, these relationships are denoted as unidirectional because they are set up from the viewpoint of the goal ontology without taking into account the viewpoints of the other ontologies. Transforming these unidirectional relationships into bidirectional relationships requires the consideration of the remaining viewpoints, which are subject of the following three knowledge acquisition activities. As such, knowledge acquisition proceeds gradually and incrementally across the iterations of the method until the ontology fits its scope and purpose.

4.3.2.2 Structure Acquisition

The second activity acquires knowledge about structure to build a corresponding ontology. This activity succeeds the acquisition of the goals and precedes the acquisition of the action and roles. For knowledge acquisition, this activity comprises the following sub-activities:

1. Identifying Structure
2. Structuring the Structure
3. Interlinking Structure
4. Refining Structure

Similar to the preceding activity, the respective outcomes follow the categorisation of the semantic spectrum. Accordingly, the focus is on developing a controlled vocabulary, glossary, thesaurus, and taxonomy as the predecessors of the ontology, which contains knowledge about the structure. The roles of the ontology developer and domain expert as well as text analysis and interview as the techniques are central to this activity (Figure 35).

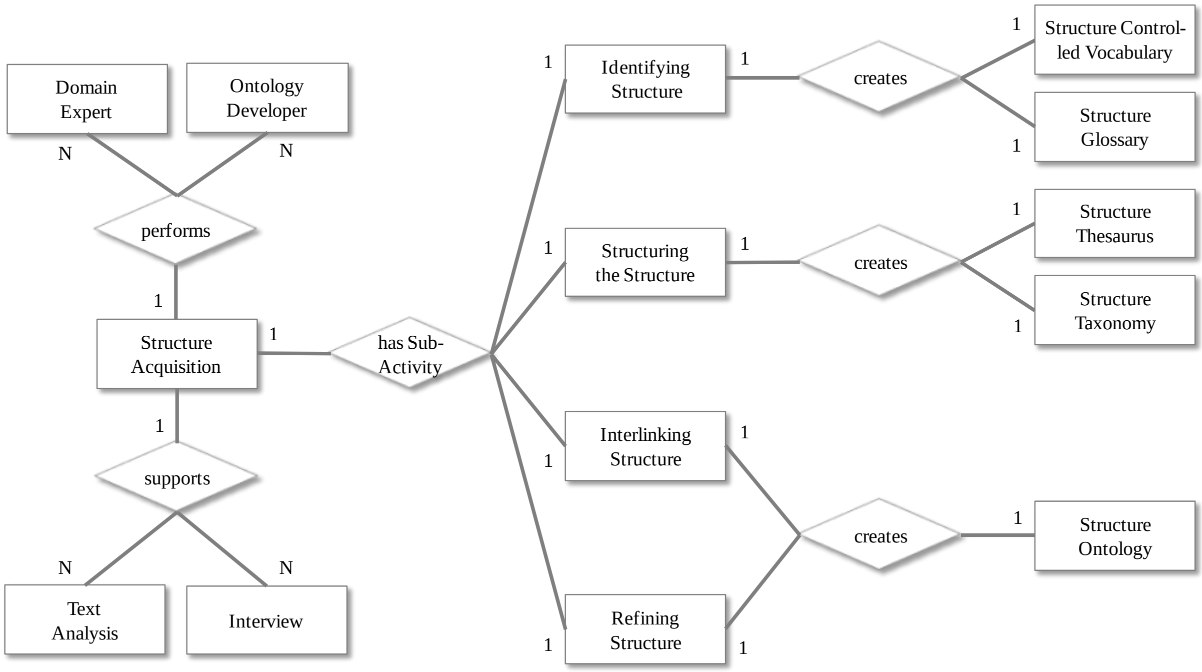

Figure 35: Structure Acquisition

4.3.2.2.1 Identifying Structure

The first sub-activity concerns identifying the structure, which is performed analogous to the identification of the goals. In contrast, for reasons of identification and interpretation a frame of reference is provided by the structural and process organisation. The structural organisation takes a static point of view on organisations (e.g. virtual organisation), whereas the process organisation incorporates a dynamic viewpoint (e.g. flows of objects). Hereby, the relationships and dependencies between these two constructs are subordinate.

The objective is to develop a controlled vocabulary and a glossary of the structure. As a finite list of terms, the controlled vocabulary is firstly developed and then extended to a glossary. The glossary specifies the meaning of terms about structure by adding natural language definitions. With regard to knowledge reuse, this sub-activity primarily focuses on controlled vocabularies and glossaries, whereas corresponding thesauri, taxonomies, and ontologies should be considered in subsequent sub-activities.

4.3.2.2.2 Structuring Structure

Based on the glossary, this sub-activity focuses on establishing taxonomic relations for building a taxonomy of the structure. Prior to creating this taxonomy, this sub-activity proposes the development of a thesaurus, which requires coping with synonyms, antonyms, broader terms, and narrower terms concerning the structure.

The glossary could already contain synonyms and antonyms or, if not, a further need for identifying such terms could arise. To avoid errors, which are due to ambiguities and misunderstandings, it is recommended to capture synonyms and antonyms at an early stage in knowledge acquisition. Defining broader and narrower terms could benefit from making explicit the implicit relationships and dependencies between the terms of the structural and process organisation. Classification or clustering could be applied to specific types of terms of the structure (e.g. department, team) to foster the construction of a thesaurus.

The introduction of generalisation/specialisation and instantiations by means of subClassOf-/ subPropertyOf-relations and typeOf-relations allows for transforming the thesaurus into a taxonomy. The subClassOf-/subPropertyOf-relations could be applied to represent hierar-chical relationships between various structural types. Further aspects of the structure could be represented by value partitions and enumerated classes, which rely on typeOf-relations.

4.3.2.2.3 Interlinking Structure

The third sub-activity deals with non-taxonomic relations. As such, it considers additional relationships and dependencies beyond taxonomic relations for developing an ontology of the structure. Interlinking covers all aspects related to the process and structural organisation, which have not been captured in the previous outcomes. For that purpose, this sub-activity considers different types of constraints, which hold on the feasibility and affect the components and relationships of the structural and the process organisation.

Such knowledge could be implicitly contained in the taxonomy of the structure, which indicates a need for transforming this implicit into explicit knowledge. Instead, this sub-activity potentially could also require the identification and structuring of additional terms of the structure. Independently, developing this ontology demands for using axioms and expressions, which are specified in the outcome metamodelling language.

4.3.2.2.4 Refining Structure

The fourth sub-activity aims at improving the ontology by checking its coherence and consistency. This means to assess whether the knowledge about the structure is logically connected and afflicted with contradictions. Based on a specific modelling language, this sub-activity considers syntactic and semantic errors by assessing the correct use of the primitives of the ontology language and graphical notation.

In addition, the focus is on establishing unidirectional relationships to the two constructs of actions and roles as well as influence factors. Similar to the goal ontology, these unidirectional relationships need to be transformed into bidirectional relationships within the subsequent activities. Prior to that, this sub-activity has to align the unidirectional relationships between the goals and the structure to establish bidirectional relationships by means of:

- removing them if they do not hold, i.e. they are not true,
- modifying them, i.e. change the direction or attach specific characteristics to the relations (e.g. inverse, symmetric, functional), or
- confirming them if they hold, i.e. they are true.

Further iterations of the knowledge acquisition method allow for including additional aspects. Such aspects might become important in the course of further knowledge acquisition, and, thus, enable for gradually refining the ontology according to its scope and purpose.

4.3.2.3 Actions and Roles Acquisition

The third activity acquires knowledge about actions and roles for constructing a respective ontology. It succeeds the acquisition of the structure and precedes the acquisition of the influence factors. For acquiring the knowledge to develop this ontology, this activity encompasses four sub-activities:

1. Identifying Actions and Roles
2. Structuring Actions and Roles
3. Interlinking Actions and Roles
4. Refining and Extending Actions and Roles

Based on the categories of the semantic spectrum, the sub-activities consecutively build a controlled vocabulary, glossary, thesaurus, taxonomy, and an ontology. This involves the roles of the ontology developer and domain expert and relies on techniques like text analysis and interview (Figure 36).

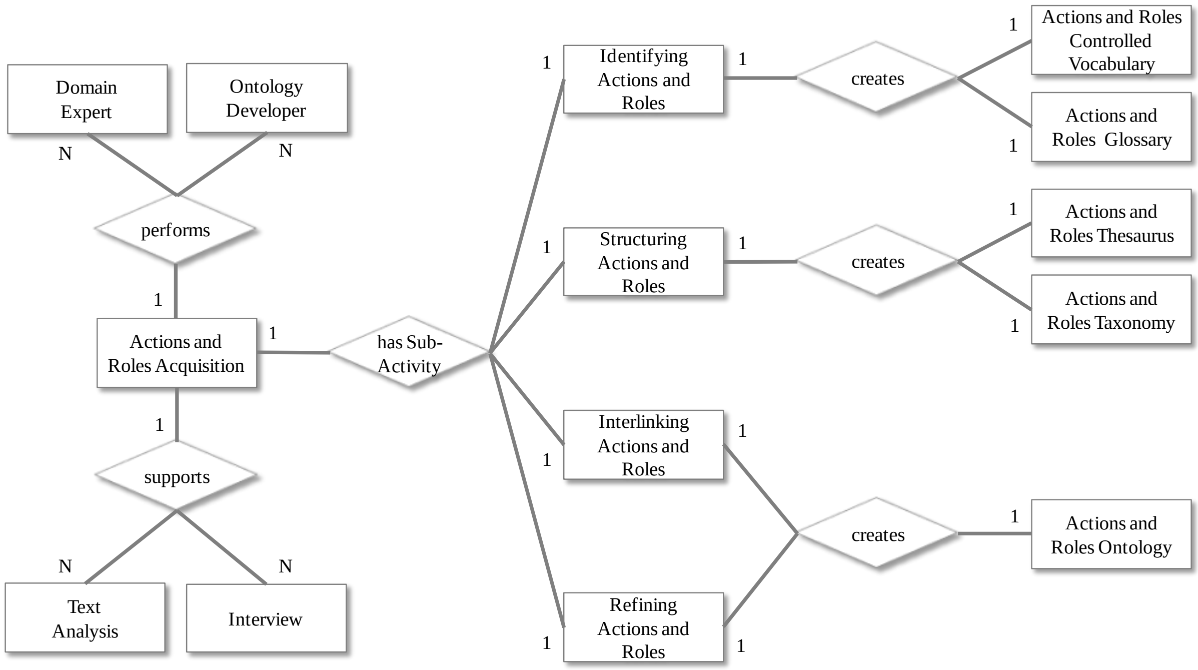

Figure 36: Actions and Roles Acquisition

4.3.2.3.1 Identifying Actions and Roles

The first sub-activity aims at identifying actions and roles analogously to the acquisition of the goals and structure. For identification and interpretation, the different types of actions based on their corporate purpose (i.e. primary and secondary actions) provide a basis. Primary actions (e.g. transport, handling, storage) directly and secondary actions (e.g. picking, packaging, signing) indirectly contribute to value creation. These actions are associated to roles, which reflect the corresponding responsibilities and competences. To complete this sub-activity, considering the relationships and dependencies between and within the actions and roles is less important.

Accordingly, knowledge acquisition focuses on the creation of a controlled vocabulary and a glossary of the actions and roles. While the controlled vocabulary represents a finite list of terms, the specification of the meaning of these terms by adding natural language definitions results in the glossary. For developing these two outcomes, reuse and adaptation of existing controlled vocabularies and glossaries is recommended. However, existing thesauri, taxonomies, and ontologies should be incorporated in the following sub-activities.

4.3.2.3.2 Structuring Actions and Roles

Based on the glossary, the second sub-activity structures the actions and roles with regard to taxonomic relations for constructing a taxonomy. This objective indicates to develop a thesaurus before building the taxonomy of actions and roles, which takes into account synonyms, antonyms, broader terms, and narrower terms of actions and roles.

The glossary could already contain synonymous and antonymous terms. Then, there is a need to make them explicit and add them to the thesaurus. Otherwise, synonyms and antonyms have to be identified as early as possible within knowledge acquisition to avoid ambiguities and misunderstandings as potential sources of errors. The relationships and dependencies between and within the actions and roles could be considered to capture broader and narrower terms. With regard to further specific types of the actions and roles, classifying and clustering them in a mutually exclusive and collectively exhaustive way supports constructing the thesaurus.

For transforming this thesaurus into a taxonomy of actions and roles, subClassOf-/ subPropertyOf-relations and typeOf-relations could be used to represent generalisation/ specialisation and instantiations. SubClassOf-/subPropertyOf-relations allow for modelling hierarchical relationships within and between various types of actions and roles, whereas typeOf-relations, for instance, allow for representing types of actions and roles in terms of value partitions and enumerated classes.

4.3.2.3.3 Interlinking Actions and Roles

The third sub-activity deals with the interlinking of the actions and roles. To construct an ontology about actions and roles, it considers relationships and dependencies beyond taxonomic relations (i.e. non-taxonomic relations). As such, this sub-activity covers all aspects related to the actions and roles, which the preceding sub-activities have not addressed. Therefore, the focus is on capturing different types of actions with respect to their purpose and constitution as well as different types of roles with regard to different types of actions. In addition, various types of constraints, which hold on the feasibility of the actions and roles, are included.

Capturing this knowledge could partly rely on the taxonomy but also invoke a need for further knowledge acquisition with regard to identifying and structuring additional actions and roles. For interlinking, the outcome metamodelling language provides means in terms of axioms and expressions to develop the ontology about the action and roles.

4.3.2.3.4 Refining Actions and Roles

The fourth sub-activity concentrates on improving the ontology by checking whether the knowledge about the actions and roles is logically connected (coherent) and contains any contradictions (consistent). Syntactic and semantic errors in terms of using the primitives of the ontology language and graphical notation correctly are also subject of ontology refinement.

Moreover, refining this ontology focuses on constructing unidirectional relationships to the construct of influence factors. Similar to the ontologies of the goals and structure, transforming these unidirectional relationships into bidirectional relationships is subject of the subsequent activity. Despite, aligning unidirectional relationships between the goals, the structure, as well as the actions and roles requires corresponding transformation based on removing, modifying or confirming exiting (unidirectional) relationships.

These transformations could induce further method iterations, which allow for taking into account additional aspect for enhancing the ontology about actions and roles.

4.3.2.4 Influence Factors Acquisition

The fourth activity acquires knowledge about influence factors to build an ontology about influence factors. Therefore, this activity is decomposed into four sub-activities:

1. Identifying Influence Factors
2. Structuring Influence Factors
3. Interlinking Influence Factors
4. Refining Influence Factors

These sub-activities consecutively develop a controlled vocabulary, glossary, thesaurus, taxonomy, and an ontology according to the semantic spectrum. Ontology construction involves both the roles of the ontology developer and domain expert as well as the knowledge acquisition techniques text analysis and interview (Figure 37).

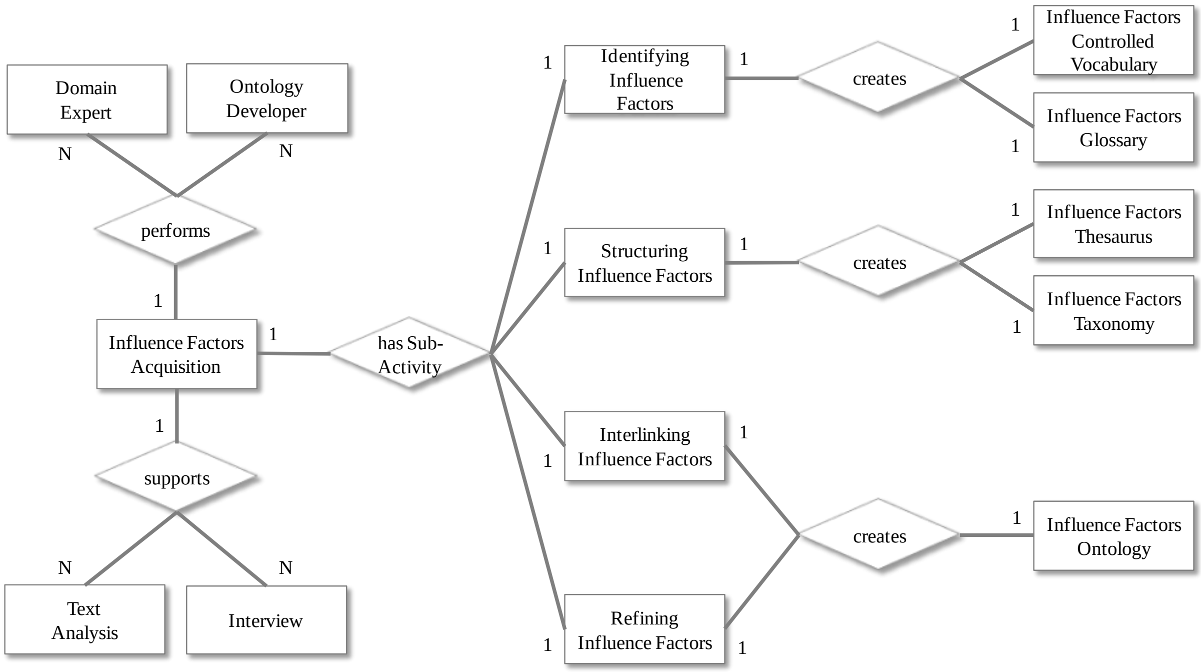

Figure 37: Influence Factors Acquisition

4.3.2.4.1 Identifying Influence Factors

The first sub-activity addresses the identification of the influence factors. Similar to this type of previously conducted sub-activities, the different types of influence factors (i.e. internal and external) provide a frame of reference for reasons of identification and interpretation. Internal influence factors distinguish between historical and present factors, whereas external factors deal with the global and task-specific environment. Predominant relationships and dependencies between them are subordinate within this sub-activity.

Against that background, the objective is to create two types of outcomes: a controlled vocabulary and a glossary. The controlled vocabulary enumerates a finite list of terms, which reflect the influence factors. These terms are further specified through adding natural language definitions to obtain the glossary of influence factors. Thereto, existing controlled vocabularies and glossaries can be reused and adopted. The reuse of other existing knowledge sources such as thesauri, taxonomies, and ontologies is suggested to be postponed to the subsequent sub-activities.

4.3.2.4.2 Structuring Influence Factors

This sub-activity considers the structuring of influence factors. Structuring influence factors means to establish taxonomic relations between the terms in the glossary. The initial result is a thesaurus, which is further developed to a taxonomy.

For building a thesaurus demands for dealing with synonyms, antonyms, broader terms, and narrower terms. Synonymous and antonymous terms could already be found in the glossary or, if not, such terms need be identified. Avoiding ambiguities and misunderstandings as potential sources of errors is supported by identifying synonyms and antonyms as early as possible within knowledge acquisition. The consideration of relationships and dependencies could assist in capturing broader and narrower terms. Moreover, classification and clustering of additional types of influence factors in a mutually exclusive and collectively exhaustive way supports the creation of the thesaurus.

Further developing the thesaurus towards a taxonomy relies on introducing generalisation/ specialisation and instantiation in terms of subClassOf-/subPropertyOf-relations and typeOf-relations respectively. Proposing subClassOf-/subPropertyOf-relations serves the modelling of hierarchical relationships within and between various types of influence factors, whereas typeOf-relations, for example, allow for using value partitions and enumerated classes to represent influence factors.

4.3.2.4.3 Interlinking Influence Factors

The third sub-activity deals with interlinking of influence factors by focussing on non-taxonomic relations. For constructing an ontology about influence factors, this sub-activity comprises all aspects, which the respective outcomes of the preceding two sub-activities have not dealt with. As such, this sub-activity considers different types of constraints, which hold on the feasibility of the influence factors and affect the relationships and dependencies between them.

This knowledge is either already contained in the taxonomy or it needs to be acquired by the identification and structuring of corresponding terms. Further, ontology development relies on an increasing use of axioms and expressions as specified in the outcome metamodelling language.

4.3.2.4.4 Refining Influence Factors

The fourth sub-activity aims at enhancing the ontology of influence factors by refining it. This refinement is about assessing the coherence and consistency as well as checking for syntactic and semantic errors.

Ontology refinement mainly concerns the transformation of unidirectional into bidirectional relationships. Thus, the focus is on aligning the unidirectional relationships between the influence factors and the ontologies about the goals, structure, as well as actions and roles. Accordingly, the transformation of the relationships induces removal, modification or confirmation. Within further iterations of the method, knowledge acquisition allows for dealing with additional aspects to develop the ontology of influence factors gradually.

4.3.3 Outcomes

The outcome metamodel specifies the outcomes, which adopt the categorisation of the semantic spectrum. That is, the knowledge acquisition activities and sub-activities result in five types of outcomes: controlled vocabulary, glossary, thesaurus, taxonomy, and ontology (Table 21).

Activity	Outcome
Identification	Controlled Vocabulary, Glossary
Structuring	Thesaurus, Taxonomy
Interlinking	Ontology
Refining	Ontology

Table 21: Outcomes of Knowledge Acquisition Activities

Based on Table 21, the knowledge acquisition activities result in following different but consecutive types of outcomes:

1. A *Controlled vocabulary* and a *glossary* are created by *identification (and interpretation)*.
2. A *thesaurus* and taxonomy is created by *structuring*.
3. An *ontology* is created by *interlinking* as well as *refining*.

Carrying out multiple iterations of the knowledge acquisition method allows for further developing the respective outcomes until the ontology fits its defined scope and purpose. As such, the outcomes reflect the incremental and iterative process as well as the hierarchical and modular structure of knowledge acquisition.

For constructing the outcomes, knowledge acquisition recommends ontology reuse, ontology design principles, and ontology design patterns.

Ontology reuse denotes the use of existing bodies of knowledge according to the categorisation of the semantic spectrum. Reuse presupposes considering the specific knowledge acquisition activities as well as the scope and purpose of the ontology to be developed. For instance, reusing ontologies with regard to a particular domain of interest could include the specialisation of top-level ontologies as well as the adaption and refinement of controlled vocabularies, glossaries, thesauri, taxonomies, or other specific domain ontologies (Simperl 2009).

Ontology design principles correspond to quality criteria in terms of desiderata, i.e. desired properties that the ontology should exhibit though their direct assessment is difficult and achieving them completely is often not possible. For instance, the design principles of minimal encoding bias and minimal ontological commitment are difficult to access for an objective assessment but depict relevant quality criteria for ontology development (Arpírez-Vega et al. 1998, pp. 16-18; Borgo et al 1996, pp. 5-6; Gómez-Pérez 2004; Gruber 1995, pp. 909-911; Grüninger and Fox 1995; Obrst et al. 2007).

Ontology design patterns represent basic building blocks that offer a practical way to deal with recurring issues in ontology development. These issues, inter alia, concern the ontology structure, content, and representation. For instance, design patterns for ontology content comprise patterns for modelling agent and roles, collections, simple or aggregated objects, or time indexed situations (Blomqvist and Sandkuhl 2005; Gangemi 2005; Presutti and Gangemi 2008; Gangemi and Presutti 2009).

4.3.4 Roles

The knowledge acquisition method involves the roles of the ontology developer and domain expert for performing its activities. The ontology developer performs the activities, applies the techniques, and constructs the outcomes during knowledge acquisition. The domain expert assures that the ontology fits its scope and purpose as well as provides a specific source of domain knowledge.

The allocation of tasks and responsibilities indicates that knowledge acquisition centres on the role of the ontology developer. During knowledge acquisition, the involvement of the two roles remains constant across several iterations of the knowledge acquisition activities but varies within the respective sub-activities (Figure 38).

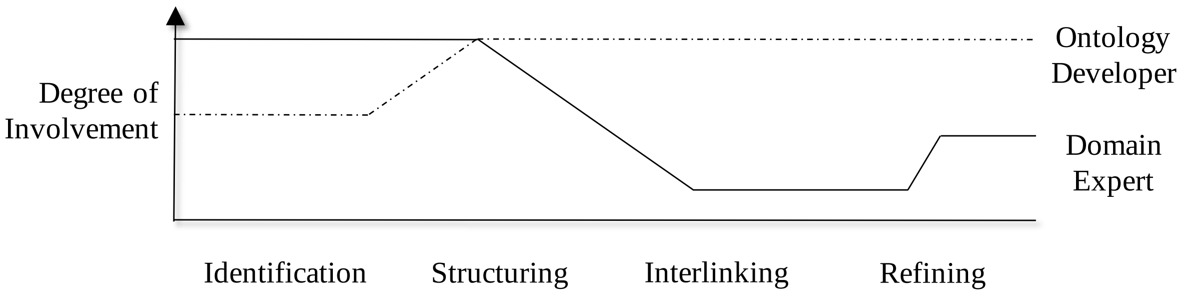

Figure 38: Involvement of Roles in Knowledge Acquisition

Figure 38 shows a higher degree of involvement of the domain expert during the sub-activity of identification and during large parts of structuring than the ontology developer. This difference in involvement intends to align knowledge acquisition with the ontology's scope and purpose to fulfil the specific requirements of the target application at an early stage. For instance, an initial high involvement of the domain expert contributes to obtaining a more concrete and comprehensive understanding of the particular domain of interest. In contrast, the ontology developer’s involvement increases during the sub-activities of structuring, interlinking, and refining, whereas the domain expert becomes less involved. However, within the sub-activity of refining, the domain expert’s involvement increases due to conducting a target/actual comparison of the outcomes of knowledge acquisition. The result of this comparison potentially triggers further iterations of knowledge acquisition.

This pattern of role involvement results from the characteristics of the knowledge acquisition (sub-)activities. It holds across the multiple method iterations. Nonetheless, the pattern allows for adaptations according to the particular requirements of knowledge acquisition. For instance, in case there is a lacking availability of domain experts in ontology development.

4.3.5 Techniques

Techniques provide particular types of procedures for supporting the knowledge acquisition activities. They can be applied with regard to various ontology components (e.g. classes, properties, hierarchies), framework conditions (e.g. unfamiliar or familiar domains of interest), and at different stages of knowledge acquisition (e.g. early, late). Predominantly, their focus is on the identification (and elicitation) of knowledge while presupposing an involvement of domain experts. Against this background, knowledge acquisition techniques can be distinguished along the following categories (Boose 1989, pp. 12-13; Byrd et al. 1992, pp. 119-133; Schreiber 2002, pp. 191-214):

Interview covers several types that range from unstructured to structured interviews. Unstructured interviews do not rely on a formal agenda and pursue specific interview goals. In contrast, structured interviews rely on a formal agenda. Both unstructured and structured interviews have different advantages and disadvantages so that their usefulness depends on several factors, e.g. specific domain characteristics or availability of domain experts (Meyer and Booker 2001; Waldron 1986).

Protocol analysis (think aloud method) summarises different approaches to study the problem-solving behaviour of a human expert in a specific domain. Different forms of protocols (e.g. video, audio, text) record the problem-solving behaviour with regard to a specific task description as the basis for carrying out further analysis. The main use case of protocol analysis is the acquisition of procedural knowledge (Ericsson and Simon 1993; Wright and Ayton 1987).

Concept sorting (card sorting) categorises concepts through uncovering different viewpoints. It distinguishes between two different types: free format concept sorting (clustering) and guided concept sorting (classification). Free format concept sorting clusters a given set of cards, which represent domain concepts. On the contrary, guided concept sorting classifies the cards according to predetermined categories. Especially in unfamiliar domains, concept sorting supports the discovery and initial categorisation of classes and properties (Gammack 1987; Maiden and Hare 1998).

Repertory grid is similar to concept sorting since it aims at disclosing concepts and their relationships in a rather unfamiliar domain. It defines a procedure that starts with the selection of three concepts of the domain so that two concepts are similar to each other but different from the third concept. Making explicit the reasons for this differentiation, the first step provides a basis for distinguishing other concepts in the course of further repetitions of this procedure (Gutierrez 1987; Shaw and Gaines 1987).

Text analysis comprises several techniques to exploit textual knowledge sources (e.g. textbooks, technical articles). For example, text analysis relies on a keyword-based search to support the identification of concepts and their relationships, whereas abstracting fosters narrowing huge bodies of textual knowledge sources to enable focusing on central domain concepts and relationships (Cleal and Heaton 1988; Tang et al. 1994).

Based on that categorisation, predominantly text analysis and interview techniques are used to support the knowledge acquisition activities and sub-activities. The degree of utilisation largely overlaps with the degree of the involvement of the respective roles (Figure 39).

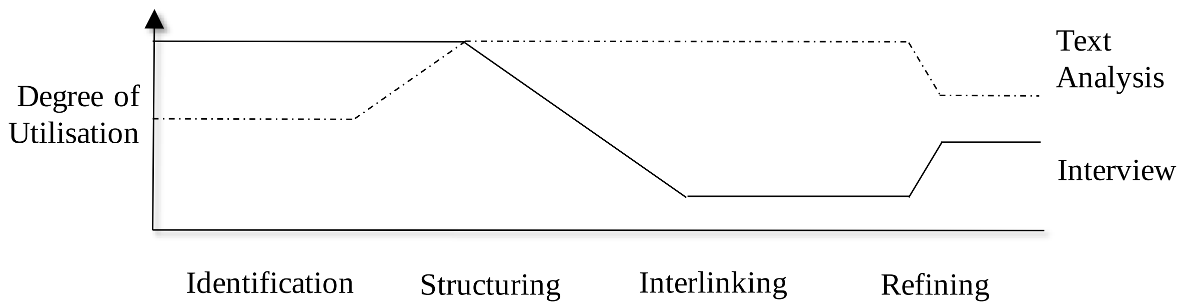

Figure 39: Utilisation of Techniques in Knowledge Acquisition

Figure 39 depicts that the technique interview has a higher degree of utilisation during the sub-activity of identification and during large parts of structuring than text analysis. Similar to the rationale for role involvement, interview techniques intend to align knowledge acquisition with the ontology’s scope and purpose prematurely. For example, an initial high utilisation of interview techniques contributes to a more concrete and comprehensive understanding of the domain. In contrast, the utilisation of text analysis increases during the subsequent sub-activities, whereas the degree of interview technique decreases. Within the sub-activity of refining, the utilisation of the interview technique increases due to a target/actual comparison of knowledge acquisition. Potentially, the respective result of this comparison could trigger further iterations of knowledge acquisition.

The pattern of technique utilisation is mainly due to the characteristics of the knowledge acquisition (sub-)activities. It holds across multiple iterations of knowledge acquisition. Nonetheless, this pattern could be adapted based on the particular requirements of ontology development (e.g. availability of knowledge sources).

In accordance to the techniques, knowledge acquisition differentiates between two types of knowledge sources: textual and human knowledge sources. Textual knowledge sources cover unstructured, semi-structured, and structured types of informal, semi-formal, or formal documents. Human knowledge sources refer to as domain experts. For exploiting such knowledge sources, the method additionally recommends the use of dedicated tools.

5 Evaluation

This chapter reports the evaluation of the proposed method for theory-based knowledge acquisition. First, approaches to evaluation are discussed. Second, the criteria-based evaluation is presented. Third, the scenario-based evaluation is reported.

5.1 Approach

Evaluation concerns the purposeful and systematic assessment of material or immaterial objects based on justified criteria and methods (House 1993, p. 1). It constitutes a central component of design science research by not only assessing the design artefact but also providing valuable feedback for its iterative and incremental development. However, evaluating a design artefact might lead to potential difficulties, for instance, when selecting justified evaluation criteria (Frank 2000, p. 36; Hevner et al. 2004, pp. 85-87).

For attenuating potential difficulties, the evaluation of a design artefact requires defining its subject area, objectives, methods, and criteria. The *subject of evaluation* constitutes the method for theory-based knowledge acquisition in ontology development as described in Chapter 4. The *objective of evaluation* covers the design artefact's utility (Hevner et al. 2004, p. 85). It requires demonstrating that the proposed artefact fulfils the requirements of method design and reduces the problem of knowledge acquisition. Additionally, design science research proposes a multiplicity of *methods for evaluating* the design artefact. Evaluation methods can be divided into observational, analytical, experimental, testing and descriptive methods (Hevner et al. 2004, p. 86). Next, Table 22 summarises the applicability of available methods for the artefact at hand.

Category of Evaluation Method	Evaluation Method and Applicability
Observational	*Case study* would study the use of the proposed method in depth in a real business environment. Such a study must consider the qualification, skills, background knowledge, and prior experience of the involved persons with knowledge acquisition in ontology development, as well as their attitude towards applying the new method in the working environment. Therefore, evaluating the practical utility is a difficult and time-consuming endeavour, whose results are subject to many confounding factors that can hardly be controlled (for lack of internal validity). This context, however, does not preclude empirical studies in the future when a large enough user base might be available (Cardoso 2007, pp. 86-87; Simperl et al. 2010). *Field study* would monitor the use of the method in multiple projects; thus, its applicability is dependent on successful case studies.
Analytical	All these methods such as *static*, *architecture,* and *dynamic analysis* can only be applied to formally specified artefacts such as axioms and algorithms; however, the proposed method is targeted at persons that carry out activities and interact with others.
Experimental	*Controlled experiment* would study the use of the proposed method in a controlled environment (laboratory). While experimentation is of course the proper way to maximise internal validity, using the proposed method for a realistic problem domain requires not only qualified participants but also a quite long time span of several days or weeks, which exceeds that of laboratory settings (restricted to a few hours). Considering a small-scale problem domain only would incur the risk of 'toy experiments' of very low external validity. *Simulation* would execute the method with artificial data/machines instead of persons; hence, it cannot be applied to the method (which is necessarily targeted at persons and their social interactions).
Testing	*Functional* and *structural testing* are only applicable to instantiations (software systems) but not to method artefacts.
Descriptive	*Informed argument* would rely on information from the knowledge base, e.g. criteria that allow for assessing the characteristics of the designed artefact, to demonstrate the artefact's utility. *Scenarios* would construct a detailed scenario around the proposed artefact to demonstrate its utility.

Table 22: Applicability of Evaluation Method

In summary, descriptive methods are applicable for those artefacts that are particularly innovative and hardly accessible for other evaluation methods. Descriptive evaluation is satisfactory because it provides feasible means with regard to the subject of evaluation while satisfying the evaluation objectives.

5.2 Criteria-based Evaluation

The criteria-based evaluation compares the design artefact against its requirements. The fulfilment of the requirements is both necessary and sufficient. Determining to which extent the method satisfies a particular requirement is straightforward for requirements that could be examined like a simple method characteristic. Otherwise, evaluation needs corresponding assumptions, which are then made explicit. Based on that, the criteria-based evaluation covers the design science, design-related, functional, and component-related requirements.

5.2.1 Evaluation of Design Science Research Requirements

The development of the knowledge acquisition method subscribes to the paradigm of design science research, which incorporates four basic principles to ensure quality design artefacts (Frank 2000, pp. 43-45; Hevner et al. 2004, p. 87; Oesterle et al. 2010, pp. 668-669). Table 23 describes these principles for contrasting them with the designed artefact.

Principle	Description	Design Artefact
Abstraction	The design artefact should contain generic statements that address a class of real situations (problems).	The method contains generic statements about theory-based knowledge acquisition in ontology development to reduce the problem of know-ledge acquisition based on the example domain of transport chains.
Originality	The design artefact should provide a novel contribu-tion to the existing know-ledge base.	The method contributes to the field of ontology engineering since research in ontologies does not yet report on theory-based knowledge acquisition methods for ontology development.
Reason	The design artefact should be intersubjective repea-table and should allow for validation.	The method is founded on a systematic and rigorous usage of the knowledge base and has been specified through formal models, which then allow for validation.
Utility	The design artefact should create a benefit for the stakeholder groups now or in the future.	The method reduces the problem of knowledge acquisition to advance the development of use-ful ontologies.

Table 23: Evaluation of Design Science Requirements

5.2.2 Evaluation of Design-related Requirements

The evaluation of the design-related requirements assesses to which extent the proposed method fulfils these requirements (rd1 - rd14), i.e. the general (rd1 - rd7) and specific (rd8 - rd14) requirements. Additionally, the designed method is contrasted with prior work on knowledge acquisition methods in ontology development (Section 2.3).

With regard to the requirements rd1 - rd7, the evaluation provides the following results.

The method focuses on the activities of identification, structuring, interlinking, and refinement. Its outcomes are based on categories of the semantic spectrum and the main constructs stem from the Contingency Approach. The activities, outcomes, and constructs are essential for theory-based knowledge acquisition (rd1).

The activity model defines all the relationships between the activities and their corresponding sub-activities (rd2). It also points out relevant relationships to the remaining method components, i.e. outcomes, roles, and techniques (rd3).

The method and outcome metamodels are specified using ERM and OWL 2 DL, respectively. Particularly, the outcome metamodel allows for ensuring consistency and coherence within knowledge acquisition and throughout ontology development (rd4). Using ERM as the method metamodelling language enables method instantiations to be checked for syntactical correctness (rd5).

Decomposition is used for the activities and outcomes of knowledge acquisition to support understandability. The graphical notation provides a clear and consistent layout design of the proposed method to support readability (rd6).

Determining method efficiency is objectively hardly feasible even when conducting empirical studies. Considering the research gap and assuming the fulfilment of this requirement, the method reduces the problem of knowledge acquisition and, thus, advances ontology development with regard to quality and time (rd7). In addition, the fulfilment of rd6 indirectly contributes to increased efficiency.

Next, the requirements rd8 - rd14 are evaluated.

The method incorporates an incremental and iterative activity model, which supports a gradual approach for knowledge acquisition (rd8).

The activity model is grounded on a theory in business economics, which supports capturing the key concepts, relationships, and dependencies of the domain of interest. Particularly, the activity model builds on the Contingency Approach, which guides the acquisition of the domain knowledge (rd9). Based on that and through adopting OWL 2 DL as the outcome metamodelling language, the terminology of the domain of interest, e.g. transport chains, can be represented (rd10). Hereby, the knowledge representation language is the primary source for potential restrictions. For instance, OWL 2 DL does not allow for n-ary relations (e.g. ternary relations) but it is possible to represent them by using multiple binary relations (Noy and Rector 2006).

The outcomes of the knowledge acquisition activities are based on the categorisation of the semantic spectrum, i.e. controlled vocabulary, glossary, thesaurus, taxonomy, and ontology. In addition, OWL 2 DL represents the outcome metamodelling language (rd11).

The designed method requires a preceding activity of ontology specification and a succeeding activity of formalisation and/or implementation. Both activities are mandatory in ontology development. Apart from this, no other restrictions apply on the proposed method (rd12).

Knowledge acquisition mainly relies on text analysis and interview techniques but equally allows for applying other techniques according to the particular needs of ontology development (rd13).

The method does not impose any requirements and restrictions on the use of specific tools for knowledge acquisition and ontology development (e.g. Protégé, NeOn Toolkit) (rd14).

This evaluation demonstrates that the knowledge acquisition method fulfils the design-related requirements apart from requirement rd7. This requirement does not allow for making a justified statement.

Table 24 contrasts the proposed method (denoted by DKAM) with existing methods in ontology development (Section 2.3).

	DKAM	CYC	GFM	UKM	MET	NMM	OTK	HSM	UPON	GODF
rd1	+	+	+	+	+	+	+	+	+	+
rd2	+	+	+	+	+	+	+	+	+	+
rd3	+	-	-	-	-	-	-	-	-	-
rd4	+	o	o	o	o	o	o	o	o	o
rd5	+	o	o	o	o	o	o	o	o	o
rd6	+	+	+	+	+	+	+	+	+	+
rd7	o	o	o	o	o	o	o	o	o	o
rd8	+	+	+	+	+	+	+	+	+	+
rd9	+	o	o	o	o	o	o	o	o	o
rd10	+	+	+	+	+	+	+	+	+	+
rd11	+	+	+	+	+	+	+	+	+	+
rd12	+	-	-	-	-	-	-	-	-	-
rd13	+	-	o	o	o	+	o	o	o	o
rd14	+	-	o	+	-	+	+	+	o	-
	+ requirement fulfilled - requirement not fulfilled o no statement possible									

Table 24: Evaluation of Design-related Requirements

With regard to fulfilling the requirements rd1 – rd14, the existing methods depict a rather homogeneous picture but fall short when comparing them to DKAM. These shortcomings become evident when considering the requirements rd3 – rd5, rd9, rd12 – rd14. It is worth mentioning that this comparison suffers from difficulties in assessing these requirements objectively due to a lack of information about the existing methods. For instance, it is hardly feasible to assess the requirements rd1 and rd6 objectively and to make justified statements about rd8, rd9, and rd10 due to information deficiencies.

5.2.3 Evaluation of Functional Requirements

The evaluation of the functional requirements assesses to which extent the designed method satisfies the functional requirements (rf1 – rf4.3). These requirements reflect the results from the deductions of the Contingency Approach. Since it is novel to ground knowledge acquisition on these deductions for guiding knowledge acquisition in ontology development, contrasting the designed method with prior work on knowledge acquisition methods (Section 2.3) is not possible. Instead, the focus is on the method's activity model, i.e. on the knowledge acquisition activities as they respectively implement the corresponding functional requirements rf1 – rf1.5, rf2 – rf2.3, rf3 – rf3.5, and rf4 – rf4.3.

The first activity deals with acquiring the goals (rf1) and, thus, accounts for the requirements rf1.1 – rf1.5. The first sub-activity identifies various types of goals (e.g. material, economic goals) (rf1.1), whereas the second sub-activity focuses on different types of relationships between these goals by establishing taxonomic relations (rf1.2). The third sub-activity constructs non-taxonomic relations to capture goal conflicts and complementarities (rf1.3, rf1.4). The fourth sub-activity refines the goal ontology (coherence, consistency, syntactic, semantic correctness), and extends it by proposing (unidirectional) relationships to consider the connections with the remaining functional requirements (rf1.5).

The second activity focuses on the acquisition of the structure (rf2) with regard to fulfilling the requirements rf2.1 – rf2.3. That is, the first sub-activity identifies various types of the structural and process organisation, whereas the second sub-activity establishes corresponding taxonomic relations between and within these structural types. Additionally, building non-taxonomic relations, e.g. to address various types of constraints, are subject of the third sub-activity (rf2.1, rf2.2). The fourth sub-activity refines the ontology about the structure as described in the previous activity. It extends the ontology by elaborating on already proposed (unidirectional) relationships and introducing new relationships and dependencies to other functional requirements (rf2.3)

The third activity concentrates on acquiring actions and roles (rf3) for satisfying the requirements rf3.1 – rf3.5. Covering the requirements rf3.1 – rf3.4 requires that the first sub-activity focuses on the identification of actions and roles (e.g. primary, secondary actions) with regard to their purpose and constitution. These actions and roles are structured by means of taxonomic relations within the second sub-activity. Then, interlinking them through establishing non-taxonomic relations is subject of the third sub-activity. Concerning requirement rf3.5, the fourth sub-activity refines the ontology of the actions and roles and extends it by considering the relationships and dependencies to the remaining functional requirements.

The fourth activity acquires knowledge about the influence factors (rf4), which demand for the implementation of the requirements rf4.1 – rf4.3. Therefore, the first sub-activity identifies various types of influences factors according to their viewpoint (e.g. internal, external) and source (e.g. historical, present). Based on that, taxonomic relations are established between and within influence factors by the second sub-activity. Non-taxonomic relations are added during the third sub-activity (fr4.1, rf4.2). For considering the relationships and dependencies to other functional requirements, the ontology about the influence factors is refined as well as extended (rf4.3).

In summary, the evaluation results demonstrate that the designed method fulfils all functional requirements.

5.2.4 Evaluation of Component-related Requirements

The evaluation of the component-related requirements depicts to which extent the method satisfies these requirements rc1 – rc5. Therefore, a comparison between the proposed method and prior work on methods for knowledge acquisition (Section 2.3) is performed.

The method design includes the specification of a method and outcome metamodels. These metamodels respectively define the modelling constructs, relationships, dependencies between these constructs, and the corresponding consistency conditions for representing the method and its outcomes (rc1).

The activity model defines the temporally and factually logical sequence of a finite set of activities for knowledge acquisition (rc2).

Each of the knowledge acquisition activities and sub-activities has specified outcomes, which follow the categories described in the semantic spectrum (rc3).

For carrying out these activities, the roles of the ontology developer and domain expert are specified as well as their respective degree of involvement during knowledge acquisition (rc4).

The knowledge acquisition activities are supported by text analysis and interview techniques (rc5).

Having shown that the proposed method fulfils the component-related requirements rc1 – rc5, Table 25 compares the characteristics of the proposed knowledge acquisition method with existing knowledge acquisition methods in ontology development (Section 2.3).

	DKAM	CYC	GFM	UKM	MET	NMM	OTK	HSM	UPON	GODF
rc1	+	-	-	-	-	-	-	-	-	-
rc2	+	+	+	+	+	+	+	+	+	+
rc3	+	+	+	+	+	+	+	+	+	+
rc4	+	+	+	+	+	+	+	+	+	+
rc5	+	+	-	-	-	o	o	o	+	+

+ requirement fulfilled - requirement not fulfilled o no statement possible

Table 25: Evaluation of Component-related Requirements

The results show that existing methods predominantly fall short in specifying method and outcome metamodels (rc1) as well as defining techniques (rc5). Across these methods, however, the description of the respective method components such as activity model and outcomes (rc2, rc3, and rc4) spans a wide array specifically with regard to their level of granularity, comprehensiveness, and extent to which implicit information are made explicit. For instance, when information about roles is missing, the method review assumed at least the role of the ontology developer involved.

5.3 Scenario-based Evaluation

The scenario-based evaluation constructs a detailed scenario around the designed method. Prior to reporting on method application, the underlying scenario is briefly described as well as preliminaries and assumptions are made explicit.

5.3.1 Scenario Overview

The scenario covers the domain of intermodal transport chains (ITC). ITC constitute the backbone of global trade as they are in charge to match supply (production of goods) with demand (consumption of goods) on a global scale. Therefore, globally dispersed and heterogeneous (logistics) actors provide specific and complex logistics services for realising the flow of logistics objects from source to destination according to individual customer demands. The object transcends the borders of different organisations (inter-organisational) and typically requires changing modes of transport (e.g. road, air, sea). For reasons of effective and efficient logistics service operations, it is important to coordinate logistics service provision and consumption along the entire intermodal transport chain. Such coordination significantly benefits from the application of advanced IT (Singh 2003).

From an IT viewpoint, the challenge of coordinating logistics services in intermodal transport chains could be addressed by Semantic Web Services. They represent a combination of Service-oriented Computing and Semantic Technologies in the form of Web Services and ontologies respectively (Breslin et al. 2010; Cardoso et al. 2006; McIlraith et al. 2001). Applying Semantic Web Services on the domain of intermodal transport chains means to map the concept of Web Services to logistics services for creating logistics web services. This mapping is complemented by developing an ontology of intermodal transport chains for providing the semantics to design semantic logistics web services. Particularly, this ontology not only structures and organises but also formally and explicitly represents the relevant domain knowledge for providing a sharable, reusable, and common terminology. For instance, such a terminology allows for semantically annotating and reasoning about logistics web services to enable their (semi-)automated discovery, ranking, composition, and execution (Hoxha et al. 2010; Scheuermann and Hoxha 2012).

5.3.2 Preliminaries and Assumptions

The application of the theory-based knowledge acquisition method is based on the following preliminaries and assumptions:

- Applying the proposed method focuses on a single iteration. This iteration covers all relevant issues of ontology development but does not represent an ontology construction project aiming at the construction of a complete ontology. Accomplishing such a goal would additionally require the selection of an ontology development method.
- Method application presupposes that the preceding activity of ontology specification, i.e. the definition of the ontology’s scope and purpose is completed. Based on the described scenario, the scope covers intermodal transport chains. The purpose is to provide an interlingua for enabling semantic annotation of logistics web services.
- The method requires a succeeding activity of formalisation and/or implementation, which is typically incorporated when using dedicated knowledge acquisition tools. Deciding on such tools induces the need for selecting an ontology language, which itself is chosen based on reasons of expressiveness and decidability.
- The ontology developer performs the knowledge acquisition activities because of scarcity of human domain experts, subject contingency, and feasibility.

– The knowledge acquisition activities are supported by text analysis. Using this technique means that textual knowledge sources are used predominantly. For instance, Table 26 depicts some textbooks and specifications being relevant for the domains.

Knowledge Source	**Author(s)**
DIN 30781 Teil 1, Transportkette: Grundbegriffe	DIN 1989
21st Century Logistics: Making Supply chain Integration a Reality. Council of Logistics Management	Bowersox et al. 1999
Designing & Managing the Supply Chain	Simchi-Levi et al. 2003
Logistics and Supply Chain Management. Creating Value-Added Networks	Christopher 2005
Transportwirtschaft. Einzelwirtschaftliche und gesamtwirtschaftliche Grundlagen	Aberle 2009
Logistik. Grundlagen, Strategien, Anwendungen	Gudehus 2010
Logistiksysteme. Betriebswirtschaftliche Grundlagen	Pfohl 2010
Supply Chain Operations Reference Model (SCOR®)	APICS Supply Chain Council 2010
Supply Chain Management Terms and Glossary	CSCMP 2013

Table 26: Example Knowledge Sources

– For each knowledge acquisition activity and sub-activity, the outcomes are presented exemplarily in an appropriate representational form, i.e. tabular forms and graphical notations.

– Knowledge acquisition uses the ontology language OWL 2 DL as defined in the outcome metamodel. When deemed appropriate, a reuse of existing controlled vocabularies, glossaries, thesauri, taxonomies, and ontologies are recommended.

– Knowledge acquisition uses the tool NeOn Toolkit as it provides state-of-the-art support for ontology development.

These preliminaries and assumptions represent the framing conditions for performing the knowledge acquisition activities and sub-activities including the respective outcomes, roles, and techniques.

5.3.3 Knowledge Acquisition Method Application

5.3.3.1 ITC Goals Acquisition

The first activity concerns the acquisition of the goals inherent to the ITC domain for constructing a respective ontology. This goal ontology should be complete with regard to the overall scope and purpose (Figure 40).

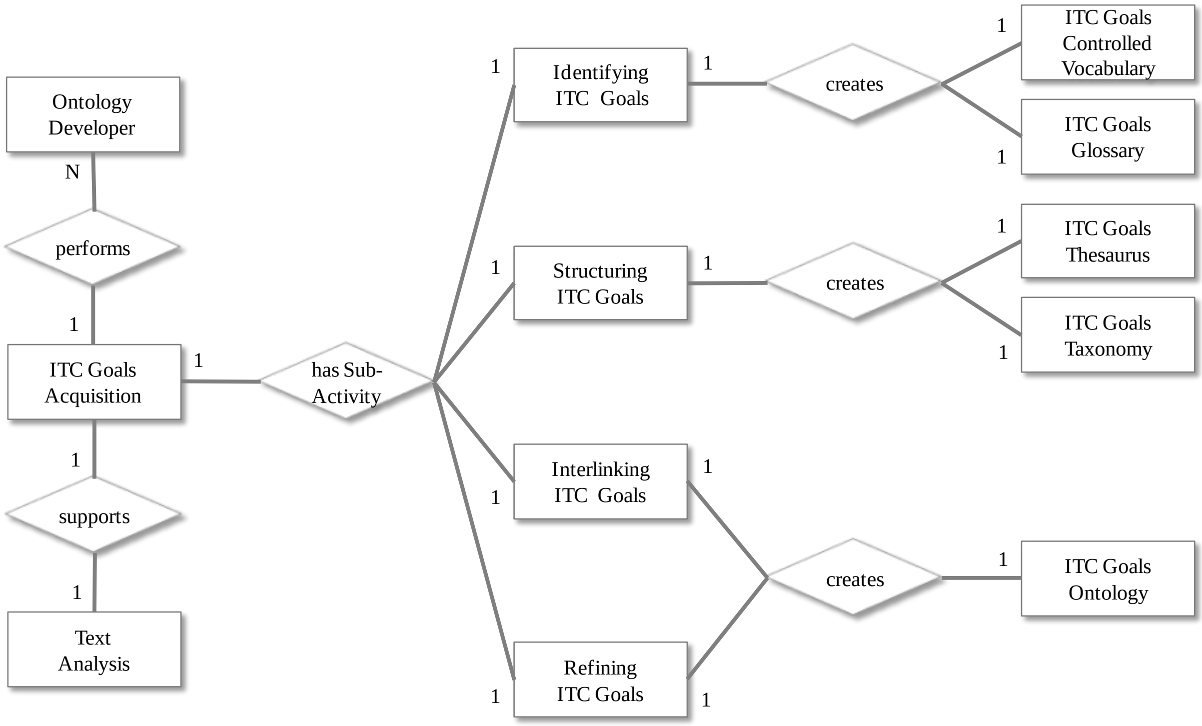

Figure 40: ITC Goals Acquisition

For acquiring the knowledge, the ontology developer relies on text analysis and carries out the following four sub-activities:

1. Identifying ITC Goals
2. Structuring ITC Goals
3. Interlinking ITC Goals
4. Refining ITC Goals

5.3.3.1.1 Identifying the ITC Goals

This sub-activity identifies and interprets the ITC goals based on the constructs of the economic and material goals. The initial result is a controlled vocabulary about the goals represented in tabular form (Table 27).

Economic Goals	Material Goals
Transport chain costs	Delivery time
Transport costs	Delivery reliability
Handling costs	Delivery quality
Storage costs	Delivery flexibility

Table 27: Part of the ITC Goals Controlled Vocabulary

This controlled vocabulary is extended by a glossary. The glossary specifies the meaning of the terms contained in the controlled vocabulary through adding natural language definitions. For that purpose, it is possible to extend the glossary’s tabular form by adding a new column. Otherwise, Neon Toolkit and OWL 2 DL Annotation Properties could be used. As an example, Figure 41 depicts the use of OWL 2 DL Annotation Properties with regard to the term *delivery flexibility*.

Figure 41: Part of the ITC Goals Glossary

5.3.3.1.2 Structuring the ITC Goals

The second sub-activity concentrates on structuring the ITC goals. It adds taxonomic relations to the glossary for constructing a taxonomy. Beforehand, synonyms, antonyms, narrower terms, and broader terms should be considered for creating a thesaurus as the intermediate result. Figure 42 shows OWL 2 DL Annotation Properties to represent synonyms of *delivery time* (modelled as the OWL 2 DL ObjectProperty *hasDeliveryTime*).

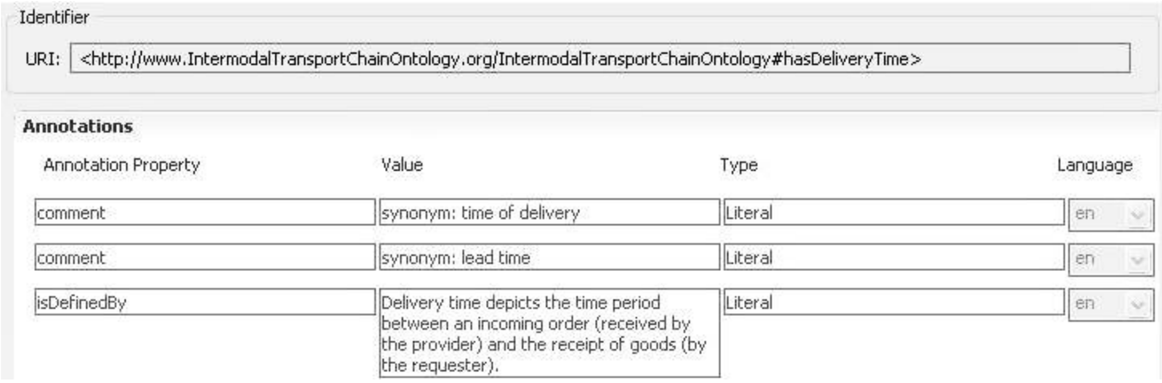

Figure 42: Synonyms of the ITC Goals Thesaurus

With regard to the term *delivery quality*, Figure 43 presents two narrower terms in the form of *orders delivered damage free conformance* and *orders delivered defect free conformance*.

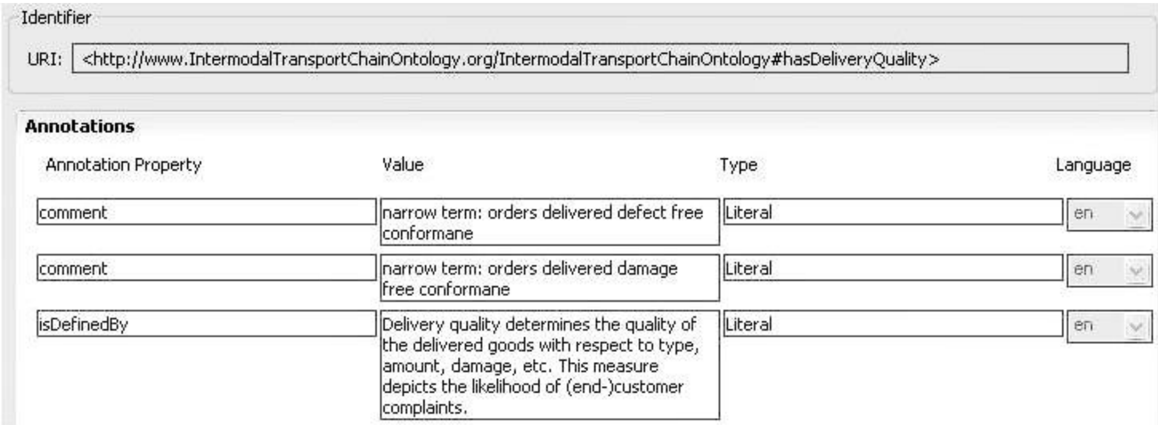

Figure 43: Narrower Terms of the ITC Goals Thesaurus

Further developing this thesaurus towards a taxonomy requires establishing taxonomic relations. Accordingly, a part of this taxonomy in the form of OWL 2 DL Object Properties in is shown in (Figure 44).

Figure 44: Part of the ITC Goals Taxonomy

5.3.3.1.3 Interlinking the ITC Goals

Based on the taxonomy of goals, the third sub-activity interlinks these goals by modelling additional relationships and dependencies beyond taxonomic relations for constructing an ontology. It considers various types of conflicts, complementarities, and constraints of intermodal transport chain goals. Figure 45 presents a part of this ontology with a particular focus on the term *delivery time* and its corresponding OWL 2 DL Axioms.

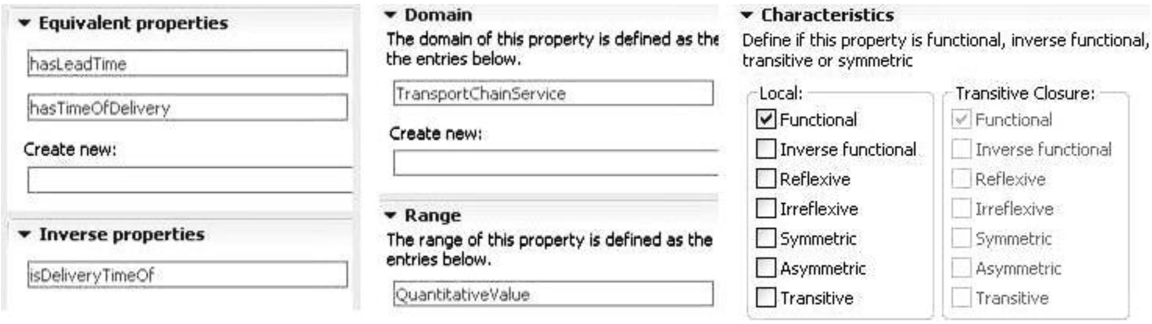

Figure 45: Axioms and Expressions of the ITC Goals Ontology

5.3.3.1.4 Refining the ITC Goals

Refining the ITC goals ontology is subject of the fourth sub-activity. It checks the ontology's coherence, consistency, syntactical errors, and modelling errors. For instance, Neon Toolkit allows for automatically checking the coherence and consistency of the ontology (Figure 46).

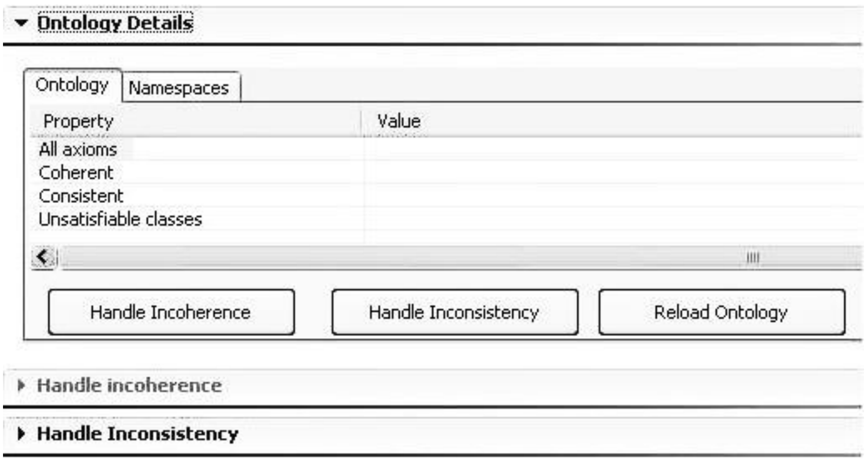

Figure 46: Coherence and Consistency of the ITC Goals Ontology

NeOn Toolkit additionally provides means for visualising the ontology (e.g. KC-Viz), supporting the formalisation and/or implementation for minimising syntactical errors, and using different formats (e.g. Functional Syntax, Manchester Syntax) for serialisation.

Further refining the ontology is about constructing unidirectional relationships to the other ontologies, which are supposed to represent further constructs of the Contingency Approach, i.e.

the structure, actions and roles, as well influence factors. Accordingly, the following enumeration not exhaustively lists potentially relevant relations:

– ITC goals explicitly characterise intermodal transport chain actions and implicitly define the corresponding capabilities of intermodal transport chain roles.

– ITC goals explicitly determine the organisation of intermodal transport chains, i.e. their structural and process organisation.

– ITC goals are subject of influence factors of intermodal transport chains and, in particular, of task-specific influence factors.

5.3.3.2 ITC Structure Acquisition

For developing an ontology about the ITC structure, the second activity acquires the relevant knowledge. This knowledge should be complete with regard to the overall ontology's scope and purpose (Figure 47).

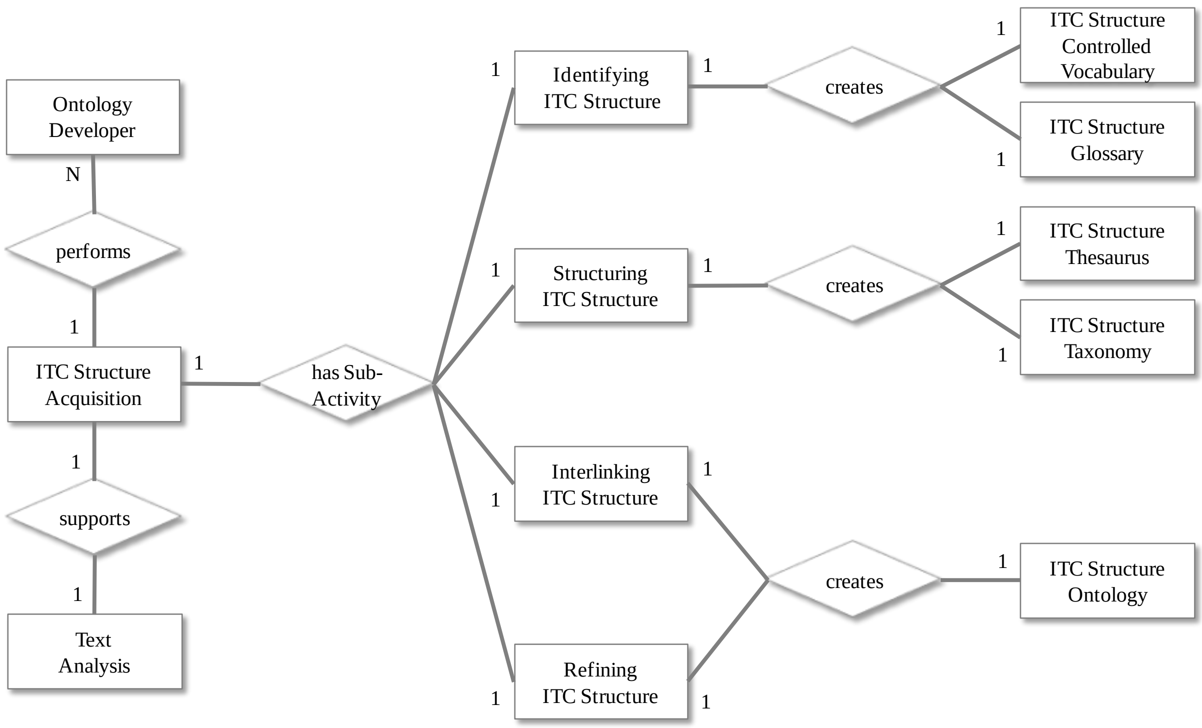

Figure 47: ITC Structure Acquisition

Figure 47 depicts that the ontology developer and text analysis are central to this activity, which is decomposed in the following four sub-activities:

1. Identifying ITC Structure
2. Structuring the ITC Structure

3. Interlinking ITC Structure
4. Refining ITC Structure

5.3.3.2.1 Identifying the ITC Structure

Based on the constructs of the structural and process organisation, the first sub-activity identifies and interprets the ITC structure to build a controlled vocabulary. This controlled vocabulary lists the relevant terms in a tabular form (Table 28).

Structural Organisation	Process Organisation
Strategic organisation	Serial
Regional organisation	Convergent
Operational organisation	Divergent
Virtual organisation	Main carriage

Table 28: Part of the ITC Structure Controlled Vocabulary

Natural language definitions are used to enrich the terms of the controlled vocabulary for developing a glossary of the ITC structure. For example, Figure 48 shows the term *strategic organisation* contained in this glossary with its defined meaning in natural language by means of OWL 2 Annotation Properties.

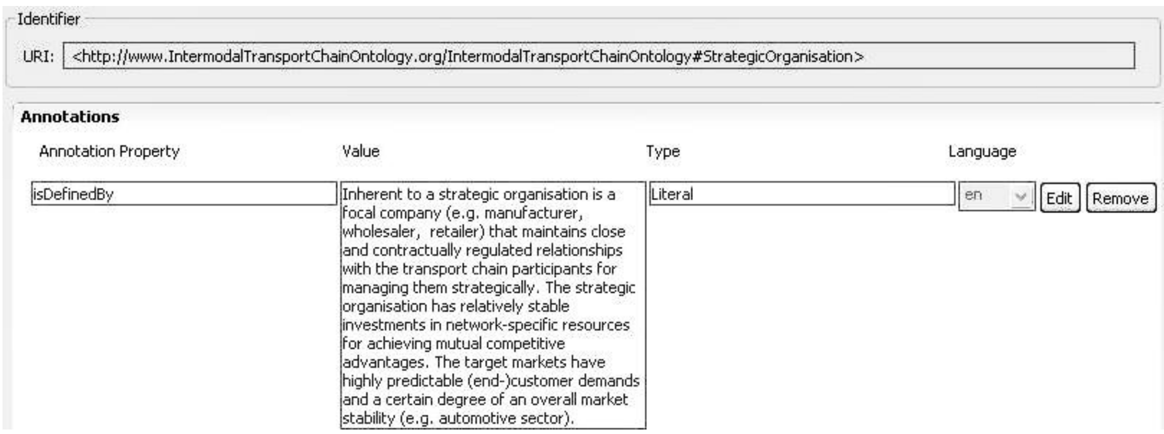

Figure 48: Part of the ITC Structure Glossary

5.3.3.2.2 Structuring the ITC Structure

For constructing a taxonomy of the ITC structure, the second sub-activity deals with taxonomic relations between the terms in the glossary. It is necessary to consider synonyms, antonyms, narrower terms, and broader terms for building a thesaurus as a basis for further developing the taxonomy. In this context, Figure 49 represents synonyms of the term *serial* (modelled as the OWL 2 Class *SerialObjectFlow*) with OWL 2 Annotation Properties.

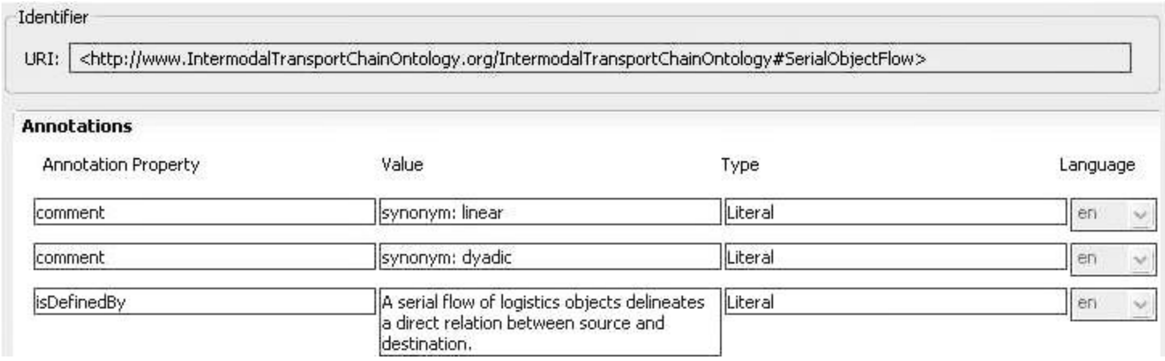

Figure 49: Synonyms of the ITC Structure Thesaurus

The term *TypeOfObjectFlow* of the process organisation represents a broader term for the terms *divergent*, *convergent*, and *serial*. Further, the terms *convergent* and *divergent* are antonymous (Figure 50).

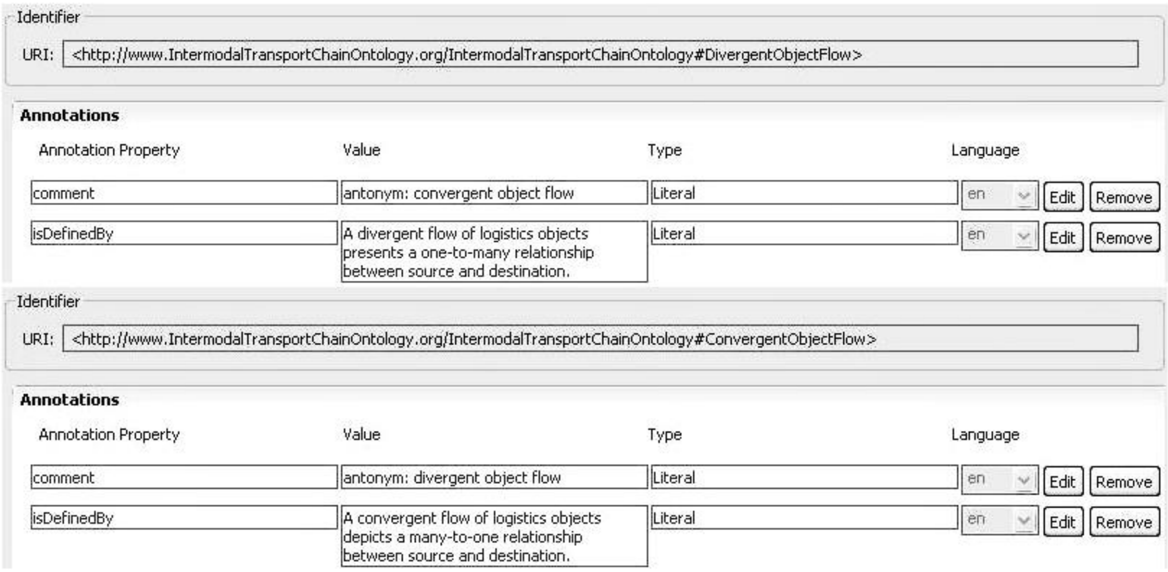

Figure 50: Antonyms of the ITC Structure Thesaurus

By establishing taxonomic relations, this thesaurus is further developed towards a taxonomy of the ITC structure (Figure 51).

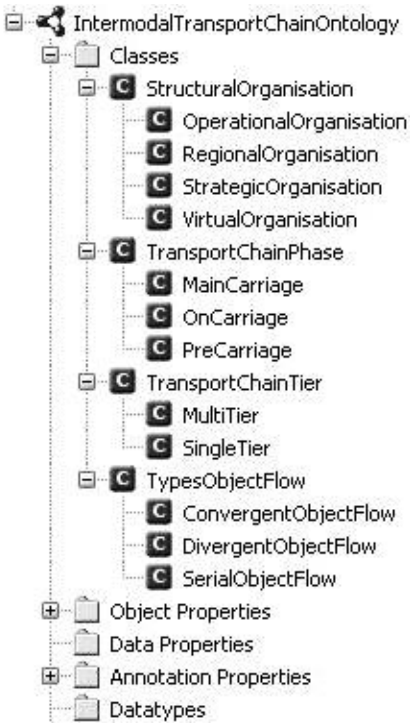

Figure 51: Part of the ITC Structure Taxonomy

5.3.3.2.3 Interlinking the ITC Structure

This sub-activity focuses on relationships and dependencies beyond taxonomic relations for building an ontology about the ITC structure. Non-taxonomic relations concern different types of constraints placed on the components, relationships, and feasibility of the structural and the process organisation. For instance, the OWL 2 Class *ConvergentOb-jectFlow* and its corresponding OWL 2 Axioms and Expressions are presented (Figure 52).

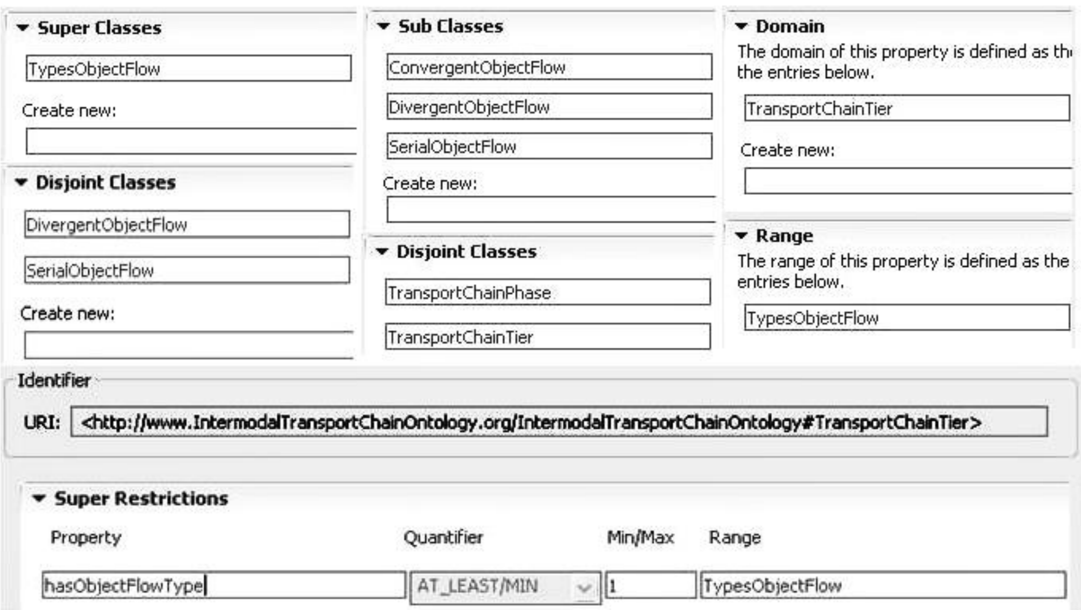

Figure 52: Axioms and Expressions of the ITC Structure Ontology

5.3.3.2.4 Refining the ITC Goals

The fourth sub-activity refines the ontology by checking for coherence, consistency, syntactical errors, and modelling errors. As such, it is analogous to the fourth sub-activity of the previous knowledge acquisition activity.

Next, ontology refinement establishes unidirectional relationships to the not yet developed ontologies about actions and roles as well as influence factors. Potentially relevant relationships deal with the following issues (not exhaustively listed):

- ITC structure determines the ITC actions as well as implicitly defines the corresponding competences and responsibilities, which are reflected by ITC roles.
- ITC structure is subject of various types of ITC factors, which are particularly task-specific influence factors.

Considering and mutually aligning the potential unidirectional relationships between the ontologies about the ITC goals and ITC structure means to remove, modify, or confirm them. Within this method iteration, the alignment of such relationships needs to be postponed to the subsequent knowledge acquisition activities or further iterations. This postponement is because the ITC goals could characterise the ITC structure either directly, indirectly through the actions and roles, or even both. Based on the scenario description as well as the ontology's scope and purpose, it could be sufficient to characterise the ITC structure indirectly through ITC actions and roles.

5.3.3.3 ITC Actions and Roles Acquisition

The third activity deals with acquiring knowledge about actions and roles that are involved in intermodal transport chains. Based on this knowledge, this activity develops an ontology about ITC actions and roles, which should be complete with regard to the overall scope and purpose (Figure 53).

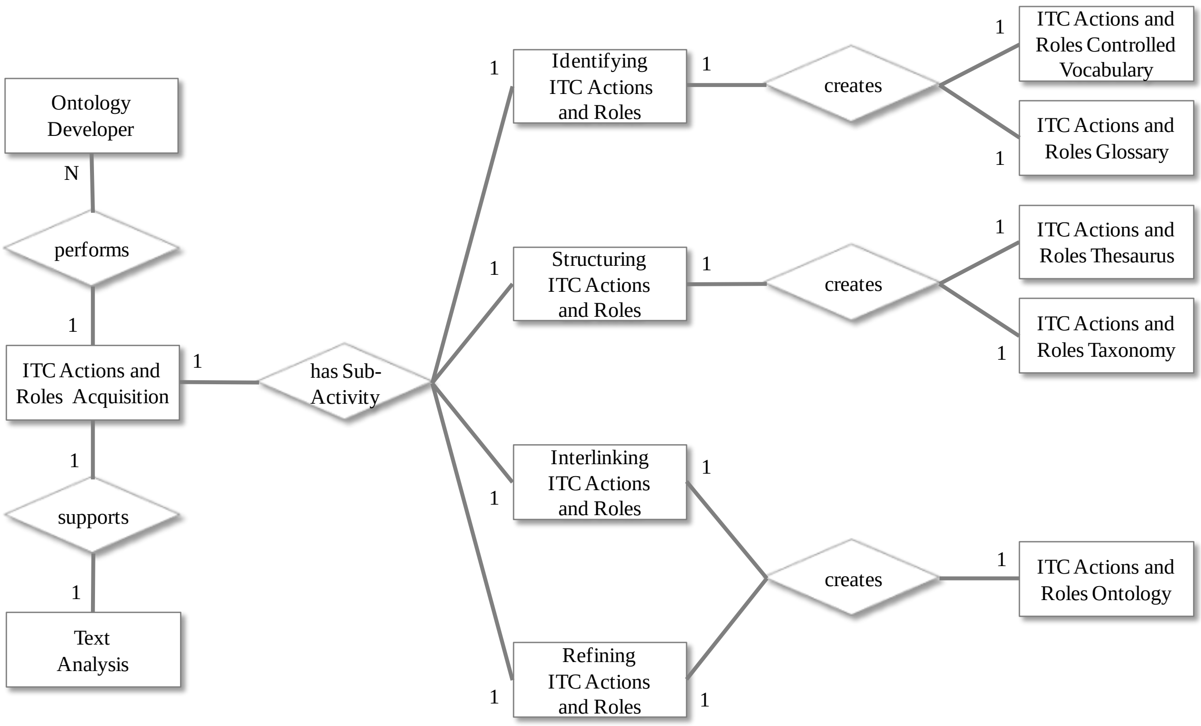

Figure 53: ITC Actions and Roles Acquisition

For acquiring such knowledge, this activity involves the ontology developer, applies text analysis, and comprises four sub-activities:

1. Identifying ITC Actions and Roles
2. Structuring ITC Actions and Roles
3. Interlinking ITC Actions and Roles
4. Refining ITC Actions and Roles

5.3.3.3.1 Identifying the ITC Actions and Roles

The first sub-activity identifies and interprets the ITC actions based on the construct of the primary and secondary actions as well as their corresponding competences and responsibilities (ITC roles). Table 29 shows the resulting controlled vocabulary.

Primary Actions	Secondary Actions	Roles
Transport	Packaging	Second-party logistics provider
Handling	Picking	Third-party logistics provider
Storage	Signing	Fourth-party logistics provider
Planning	Customs clearance	Manufacturer

Table 29: Part of the ITC Actions and Roles Controlled Vocabulary

For extending this controlled vocabulary, it is necessary to specify the meaning of the respective terms by adding natural language definitions. These definitions are implemented by means of OWL 2 Annotation Properties. Figure 54 shows the term *transport* as an example.

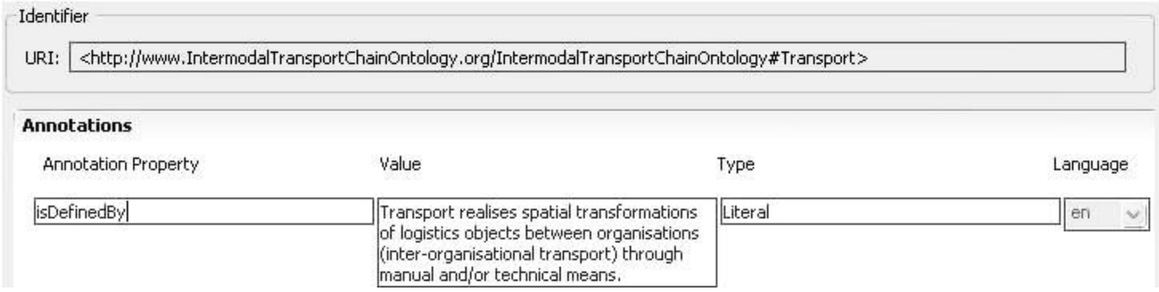

Figure 54: Part of the ITC Actions and Roles Glossary

5.3.3.3.2 Structuring the ITC Actions and Roles

The following sub-activity concerns taxonomic relations between the terms in the glossary. Dealing with such relations initially requires considering synonyms, antonyms, narrower terms, and broader terms for constructing a thesaurus. With regard to the term *transport* (modelled as the OWL 2 Class *Transport)*, Figure 55 presents the synonymous terms *shipping*, *haulage*, and *carriage* as well as the antonym *storage* by means of OWL 2 Annotation Properties.

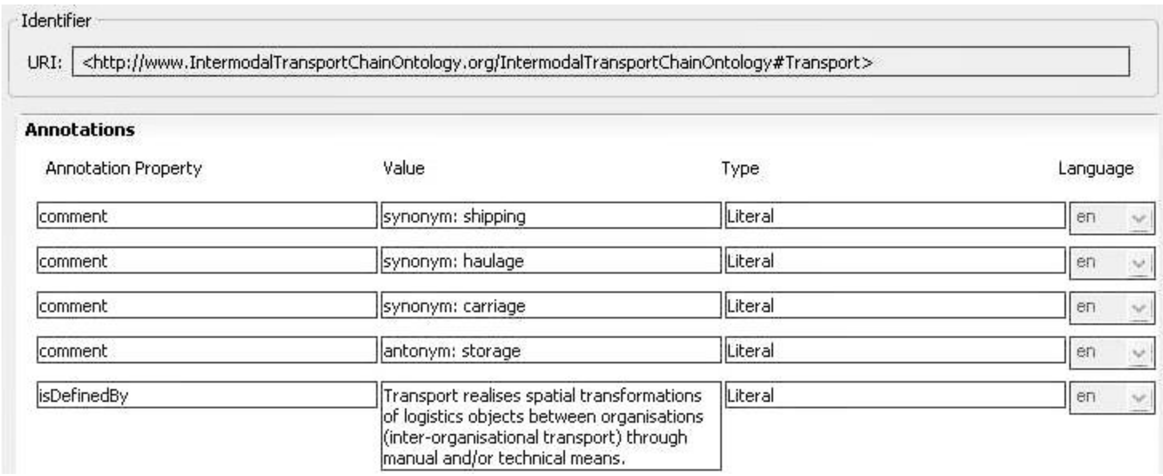

Figure 55: Part of the ITC Actions and Roles Thesaurus

As part of the construct primary actions, the term *transport* corresponds to a broader term, which encompasses, inter alia, the narrower terms *road transport*, *sea transport*, *air transport*, *individual cargo transport*, *special cargo transport*, and *bulk haulage*. These narrower terms reflect potential taxonomic relations. In this context, a part of the actions and roles taxonomy is presented with a particular focus on the OWL 2 Classes *TransportChainAction*, *TransportChainSupportAction*, and *TransportChainRoles* (Figure 56).

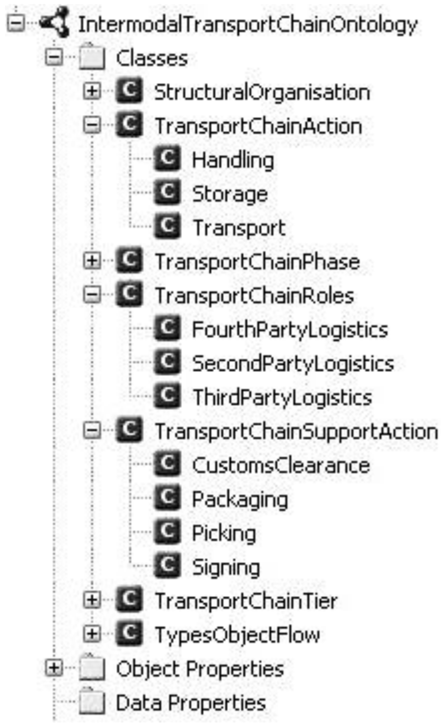

Figure 56: Part of the ITC Actions and Roles Taxonomy

5.3.3.3.3 Interlinking the ITC Actions and Roles

Besides taxonomic relations, the third sub-activity interlinks the ITC actions and roles, i.e. acquires non-taxonomic relations to model further relationships and dependencies. For constructing an ontology about ITC actions and roles, the focus is on various types of actions with respect to their purpose and constitution, types of roles with respect to these types, and occurring constraints on the feasibility of the actions and roles. As an example, Figure 57 points to the OWL 2 Class *TransportChainService* and its corresponding OWL 2 Axioms and Expressions as a part of the ITC actions and roles ontology.

Figure 57: Axioms and Expressions of the ITC Actions and Roles Ontology

5.3.3.3.4 Refining the ITC Actions and Roles

Similar to the other sub-activities, refinement checks the coherence and consistency as well as syntactical and modelling errors of the ITC actions and roles ontology.

Additionally, ontology refinement proposes unidirectional relationships to the not yet developed ontology about influence factors. Corresponding relationship mainly address the fact that these actions and roles are subject of influence factors and, in particular, of task-specific influence factors. The previously established relationships between the ITC goals, ITC structure, as well as ITC actions and roles can be mutually aligned by removing, modifying, or confirming them. For instance, there is a modification, which results in introducing the transitive OWL 2 Object Property *realises* and its inverse OWL 2 Object Property *isRealisedBy*. As such, the OWL 2 Object Property *realises* relates

– the OWL 2 Class *TransportChainService* with the OWL 2 Class *TypesObjectFlow*,

– the OWL 2 Class *TypesObjectFlow* with the OWL 2 Class *TransportChainTier*, and

– the OWL 2 Class *TransportChainTier* with the OWL 2 Class *TransportChainPhase*.

Attaching the characteristics of transitive and inverse to the OWL 2 Object Property *realises* allows for navigating across the corresponding OWL 2 Classes *TransportChainService*, *TypesObjectFlow*, *TransportChainTier*, and *TransportChainPhase* (Figure 58).

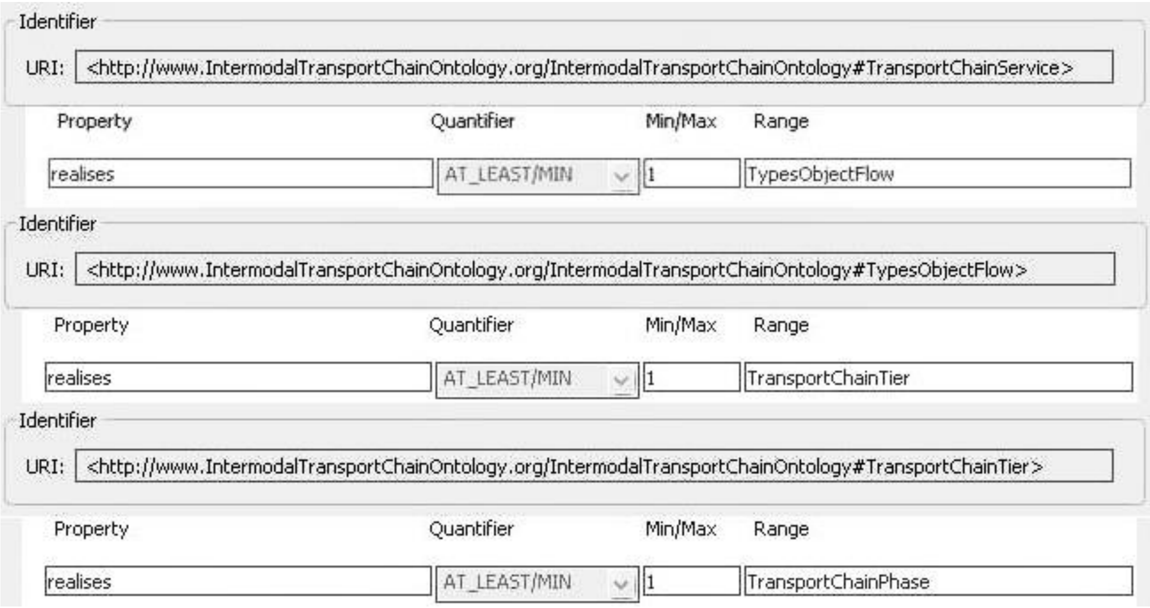

Figure 58: Part of the ITC Actions and Roles Ontology

5.3.3.4 ITC Influence Factors Acquisition

The fourth activity considers knowledge about ITC influence factors for building a corresponding ontology. The ontology about influence factors should be complete with regard to its overall scope and purpose (Figure 59).

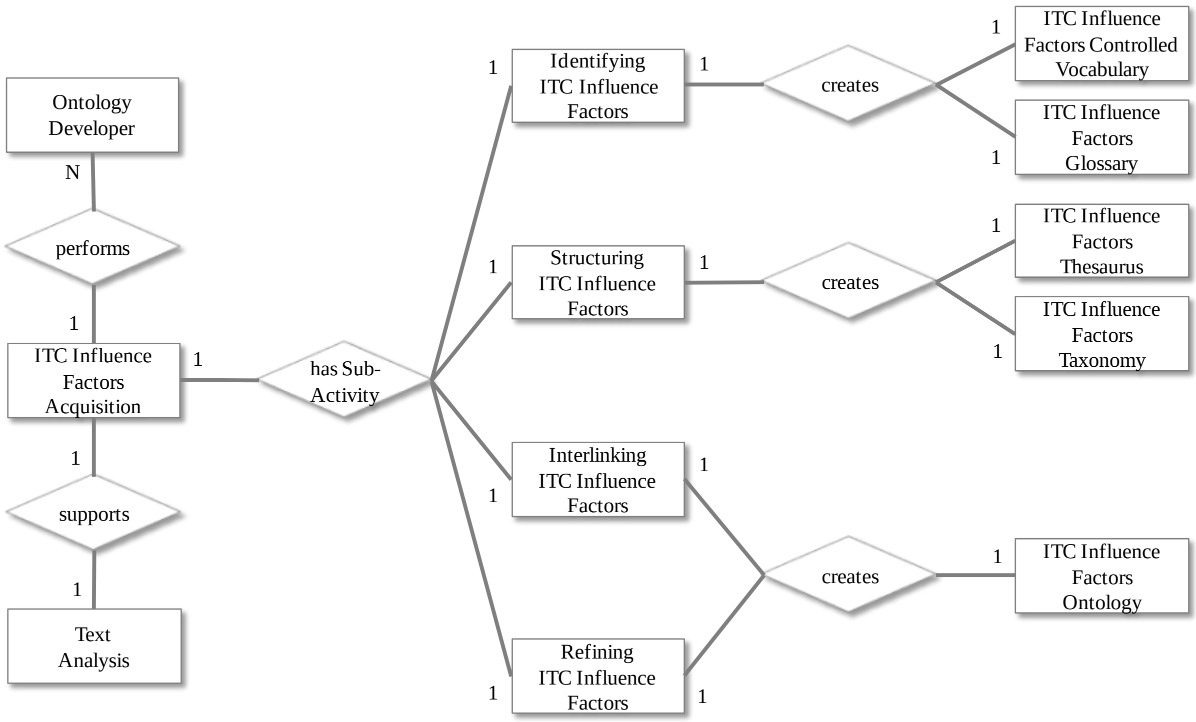

Figure 59: ITC Influence Factors Acquisition

This activity is performed by the ontology developer, supported by the technique text analysis, and decomposed into the following four sub-activities:

1. Identifying ITC Influence Factors
2. Structuring ITC Influence Factors
3. Interlinking ITC Influence Factors
4. Refining ITC Influence Factors

5.3.3.4.1 Identifying the ITC Influence Factors

Based on the constructs of internal (historical and present) and external influences (global and task-specific), the goal of this sub-activity is to identify and interpret the ITC influence factors. For instance, Table 29 exemplarily depicts a part of a controlled vocabulary, which pays special attention to global and task-specific influence factors.

Global Influence Factors	Task-Specific Influence Factors
Ecological (e.g. climate change, sustainability, carbon foot print, emission certificate trade)	Order variety
Legal (e.g. customs regulations, data protection issues, and toll)	Order volume
Economical (e.g. globalisation, competition, entry or exit of major firms)	Express cargo
Political (e.g. liberalisation of markets, transport infrastructure)	Hazardous cargo
Cultural (e.g. individualisation, changing societal concerns, attitudes, lifestyles)	Special packaging

Table 30: Part of the ITC Influence Factors Controlled Vocabulary

This enumeration points out two issues. On the one hand, these ITC influence factors constrain the ITC structure as well as the ITC actions and roles. On the other hand, such influences correspond to requirements (e.g. individual customer demands) that intermodal transport chains need to fulfil. Taking into account these two issues means to understand ITC influence factors in terms of characteristics of the ITC structure as well as ITC action and roles. For instance, the task-specific influence factor *express cargo* not only affects the structure in terms of the number of tiers, types of object flow, and intermodal transport chain phases but also determines the actions and roles with regard to the types of primary and secondary actions.

Further developing the controlled vocabulary towards a glossary of influence factors requires adding natural language definitions to the respective terms. Such a term represents *hazardous cargo* as defined by means of OWL 2 Annotation Properties (Figure 60).

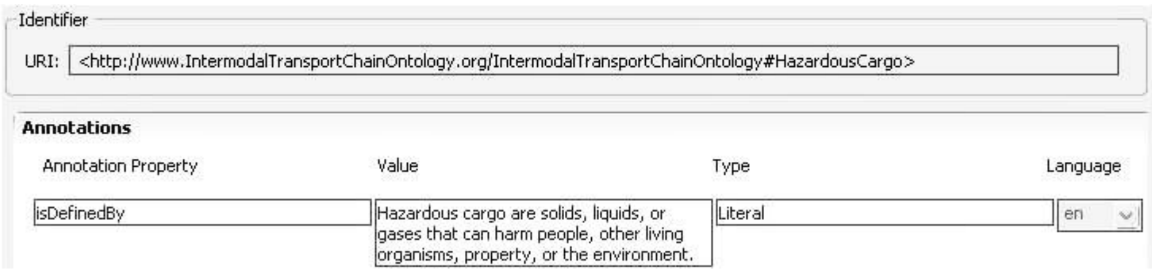

Figure 60: Part of the ITC Influence Factors Glossary

This glossary partly reuses the *hazardous cargo ontology*[2] through adapting particular terms and definitions in accordance to the ontology's scope and purpose.

[2] http://www.daml.org/2002/10/hazardous/hazardous-cargo-ont (last accessed: 2015-11-14)

5.3.3.4.2 Structuring the ITC Influence Factors

The subsequent sub-activity constructs taxonomic relations between the terms in the glossary to structure them for building a taxonomy of influence factors. This taxonomy requires a priori the development of a thesaurus, which induces a need for adding synonyms, antonyms, narrower terms, and broader terms of the influence factors. As an example, Figure 61 focuses on the term *hazardous cargo* (modelled as the OWL 2 Class *HazardousCargo*) to present synonyms such as *hazardous substance*, *hazardous material*, *hazardous good*, *dangerous substance*, *dangerous material*, and *dangerous good* as well as its antonym *standard cargo*.

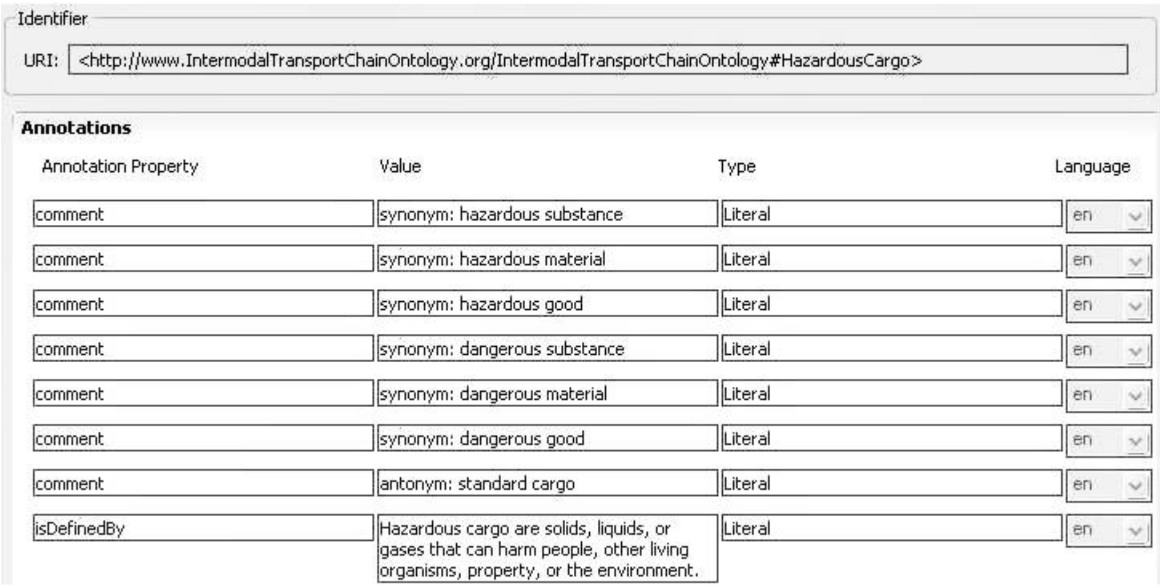

Figure 61: Part of the ITC Influence Factors Thesaurus

The term *hazardous cargo* has the broader terms *special cargo* and, more generally *cargo* as well as the narrower terms *liquid hazardous cargo*, *gaseous hazardous cargo*, and *solid hazardous cargo*. These terms indicate taxonomic relations, which are reflected as a part of the taxonomy of influence factors (Figure 62).

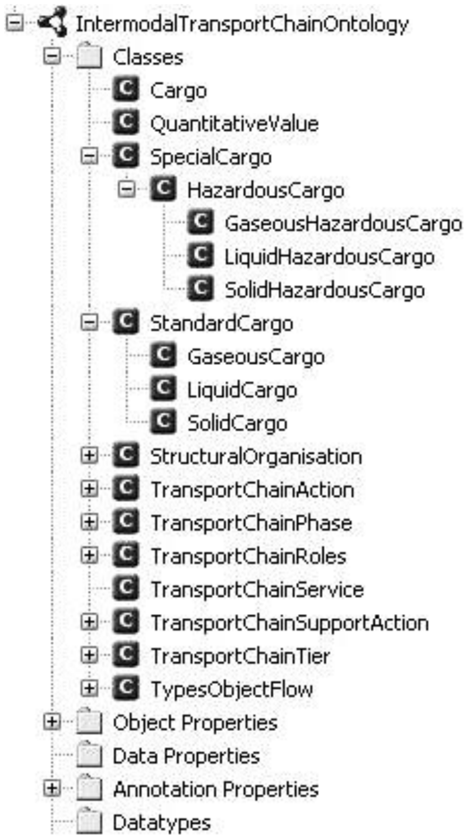

Figure 62: Part of the ITC Influence Factors Taxonomy

5.3.3.4.3 Interlinking the ITC Influence Factors

In addition to taxonomic relations, the third sub-activity builds non-taxonomic relations, i.e. additional relationships and dependencies beyond taxonomic relations to interlink the influence factors for ontology construction. Non-taxonomic relations concern different types of constraints that affect the feasibility as well as the relations and dependencies between ITC influence factors. With regard to the OWL 2 Class *HazardousCargo*, Figure 63 depicts a part of the ontology focussing on corresponding OWL 2 Axioms and Expressions.

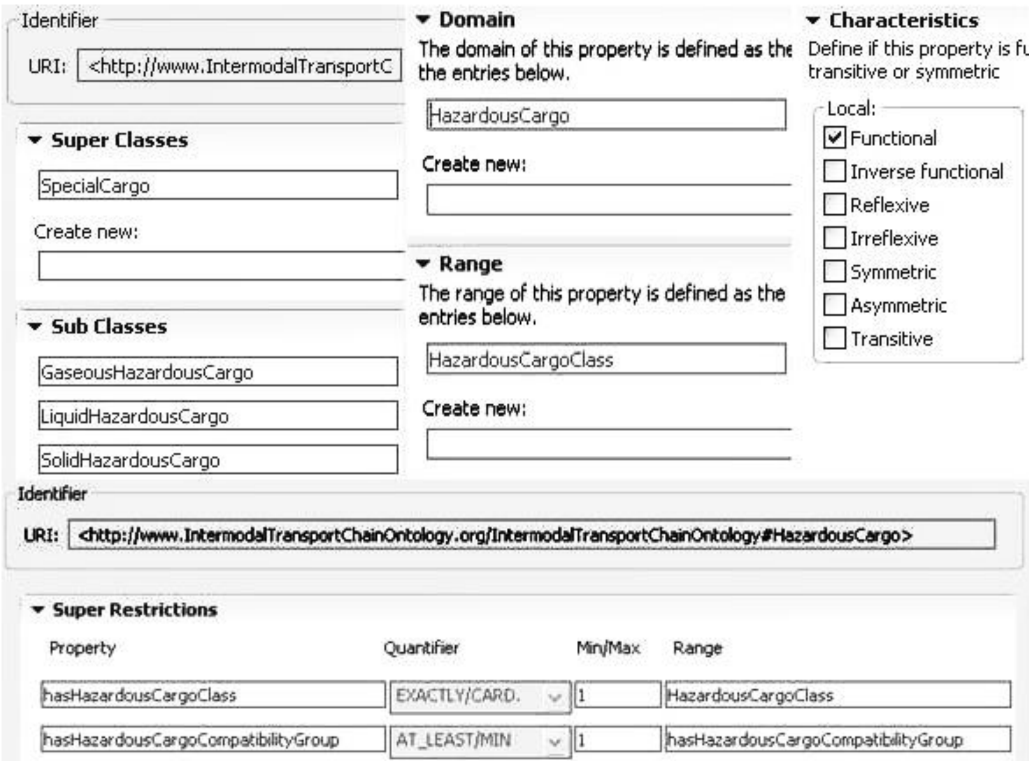

Figure 63: Axioms and Expressions of the ITC Influence Factors Ontology

5.3.3.4.4 Refining the ITC Influence Factors

The fourth sub-activity refines the ontology of influence factors. First refinement checks the ontology for its coherence, consistency, syntactical errors, and modelling errors analogous to previous sub-activities.

Subsequently, ontology refinement establishes unidirectional relationships and mutually aligns them, i.e. removes, modifies, and confirms pre-existing relationships to the ontologies about ITC goals, ITC structure, as well as ITC actions and roles. For reasons of illustration, the following example is provided:

– The OWL 2 Class *VirtualOrganisation* as a specialisation of the OWL 2 Class *Structural-Organisation* is related to the OWL 2 Class *TransportChainService* through the OWL 2 Object Property *involves*. The OWL 2 Class *TransportChainService* represents the union of the OWL 2 Classes *TransportChainAction* and *TransportChainSupport-Action*.

– The OWL 2 Class *TransportChainService* has a relation to the OWL 2 Class *Cargo* through the OWL 2 Object Property *transforms*. The OWL 2 Class *Cargo* represents the union of the OWL 2 Classes *StandardCargo* and *SpecialCargo*, which depicts the generalisation of the OWL 2 Class *HazardousCargo*.

Thus, the ontologies about the ITC structure, ITC actions and roles, as well as ITC influence factors are interlinked (Figure 64).

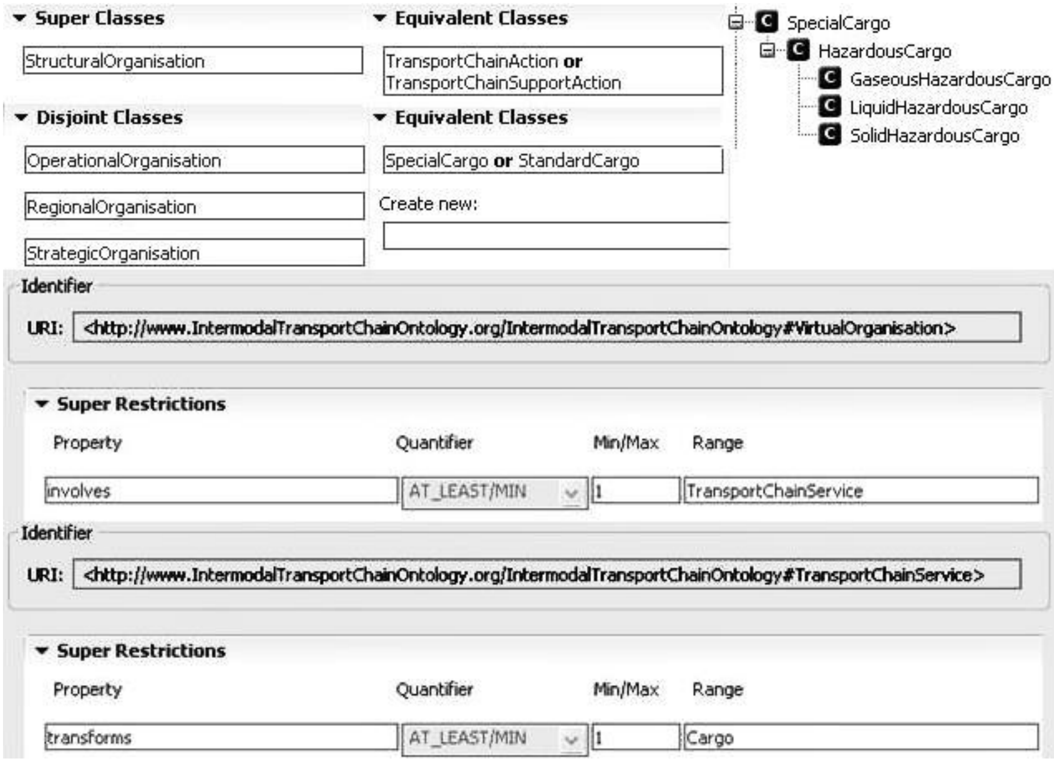

Figure 64: Part of the ITC Influence Factors Ontology

This sub-activity completes the first iteration of the method for theory-based knowledge acquisition method. The results are ontologies about the ITC goals, ITC structure, ITC actions and roles, as well as ITC influence factors. Further developing these ontologies according to the scenario description would require further iterations and potentially the involvement of domain experts and application of interview techniques.

5.3.4 Summary

The scenario-based evaluation takes the example domain of intermodal transport chains as the basis for applying the proposed knowledge acquisition method. Method application requires assumptions, e.g. it comprises a single iteration, presupposes ontology specification, centres on the ontology developer and mainly relies on text analysis. Based on that, the knowledge acquisition activities and sub-activities have been adapted to the domain of intermodal transport chains to produce exemplary outcomes in terms of controlled vocabularies, glossaries, thesauri, taxonomies, and ontologies for ITC goals, structure, actions and roles as well as influence factors. Thereby, existing knowledge bodies have been reused as well as design principles and design patterns have been applied where deemed appropriate.

6 Conclusion

6.1 Summary

The contribution of this thesis is a theory-based knowledge acquisition method for ontology development. This method proposes the use of theories in business economics in terms of partial knowledge models to guide knowledge acquisition. The rationale is to reduce the problem of knowledge acquisition by mitigating its linguistic, cognitive, modelling, and methodical difficulties. The development and evaluation of the theory-based knowledge acquisition method was based on the example domain of transport chains and the Contingency Approach.

Specifically, the following results have been achieved:

- Definition of the problem of knowledge acquisition in terms linguistic, cognitive, modelling, and methodical difficulties and empirical evidence for the difficulties in 15 transport chain ontologies.
- Demonstration of the gap in the literature concerning knowledge acquisition methods form the areas of ontology engineering and knowledge engineering.
- Deduction of requirements for method design according to an analysis of the domain of transport chains based on the partial knowledge model provided by the Contingency Approach.
- Design of the theory-based knowledge acquisition method in form of method and outcome metamodel, activity model, outcomes, roles, and techniques.
- Evaluation of the utility of the proposed method based on a criteria-based and a scenario-based evaluation method.

6.2 Limitations and Future Research

The theory-based knowledge acquisition method offers various starting points for directing future avenues of research in knowledge acquisition as well as in the more general area of ontology engineering and the discipline of IS research.

Within this specific area of research, theory-based knowledge acquisition could be subject of further studies with regard to further types of ontologies (e.g. top-level ontologies), ontology development methods (e.g. alignment, merging), theories in business economics (e.g. New Institutional Economics) and evaluation methods (e.g. scenarios, user experiments).

The field of ontology engineering could provide a fertile ground for adopting the basic idea underpinning theory-based knowledge acquisition to the areas of ontology design patterns (e.g. Gangemi 2005), ontology integration, alignment, merging, re-engineering, learning methods (e.g. Gómez-Pérez et al. 2004), ontology modularisation (Abbès et al. 2012) as well as ontology evaluation (Vrandecic 2010) and evolution (Zablith et al. 2013) based on a domain-centred viewpoint.

Within the realm of IS research, the designed artefact could be subject of behavioural science research. This research paradigm originates in natural science methods and aims at developing and justifying theories in the form of principles or laws. Such theories allow for explaining or predicting human and organisational phenomena in the context of analysing, designing, implementing, managing, and using IS (Hevner et al 2004, pp. 75-81). In other words, theory-based knowledge acquisition could be studied with regard to its perceived usefulness on an empirical basis by using IS theories such as Cognitive-Load Theory (Sweller 1998), Cognitive-Fit Theory (Vessey 1991) and Task-Technology Fit Theory (Goodhue and Thompson 1995). More ambitiously, pursuing this direction could contribute to a theory of ontology engineering, which amalgamates elements of design science and behavioural science research.

Bibliography

Aamodt A and Nygard M (1995) Different roles and mutual dependencies of data, information, and knowledge – An AI perspective on their integration. *Data and Knowledge Engineering* 16 (3): 191-222

Abbès S, Meilender T, Scheuermann A and d'Aquin M (2012) Characterizing Modular Ontologies. In: Proceedings of the Workshop 6th International Workshop on Modular Ontologies (FOIS '12), Graz, Austria

Aberle G (2009) Transportwirtschaft. Einzelwirtschaftliche und gesamtwirtschaftliche Grundlagen. 5. ed., Oldenbourg, Munich, Germany

Ahmad A, Mollaghasemi M and Rabelo L (2003) Ontologies for Supply Chain Management. In: Proceedings of the IIE Annual Conference (IERC'03), Portland, Oregon, USA

Alavi M and Leidner D (2001) Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *Management Information Systems Quarterly* 25 (1): 107-136

Anand N, Yang M, van Duin J and Tavasszy L (2012) GenCLOn: An ontology for city logistics. *Expert Systems with Applications* 39 (15): 11944–11960

Angele J, Fensel D, Landes D and Studer R (1998) Developing Knowledge-Based Systems with MIKE. *Journal of Automated Software Engineering* 5 (4): 389-418

APICS Supply Chain Council (2010) Supply Chain Operations Reference Model (SCOR®). Version 10.0, URL: http://www.supply-chain.org (last accessed: 2015-11-14)

Arpírez J, Corcho O, Fernández-López M and Gómez-Pérez A (2003) WebODE in a Nutshell. *AI Magazine* 24 (3): 37-47

Arpírez J, Gómez-Pérez A, Lozano A and Pinto H (1998) Reference Ontology and $(ONTO)^2$ Agent: The Ontology Yellow Pages. In: Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods (ECAI'98), Brighton, UK

Arpírez-Vega J, Gómez-Pérez A, Lozano-Tello A and Pinto H (1998) $(ONTO)^2$ Agent: An ontology-based WWW broker to select ontologies. In: Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods (ECAI'98), Brighton, UK

Autry C, Zacharia Z and Lamb C (2008) A Logistics Strategy Taxonomy. *Journal of Business Logistics* 29 (2): 27-51

Baader F, Calvanese D, McGuiness D, Nardi D and Patel-Schneider P (Eds) (2003) The Description Logic Handbook. Theory, Implementation and Applications. 2. ed., University Press, Cambridge, UK

Balzert H (2000) Lehrbuch der Software-Technik. 2. ed., Spektrum, Heidelberg, Germany

Baresi L, Garzotto F and Paolini P (2001) Extending UML for Modeling Web Applications. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS'01), Maui, Hawaii, USA

Barnard C (1938) The Functions of the Executive. Harvard University Press, Cambridge, Massachusetts, USA

Barney J (1991a) Firm Resources and Sustained Competitive Advantage. *Journal of Management* 17 (1): 99-120

Barney J (1991b) The Resource-based View of Strategy: Origins, Implications, and Prospects. *Editor of Special Theory Forum in Journal of Management* 17: 97-211

Baumgarten H, Klinkner R and Stommel H (2002) Integrationsaspekte des Supply Chain Management. *Logistik Management* 4 (4): 34-46

Bea F and Göbel E (2010) Organisation. Theorie und Gestaltung. 4. ed., Lucius & Lucius, Stuttgart, Germany

Becerra-Fernandez I and Sabherwal R (2001) Organization Knowledge Management: A Contingency Perspective. *Journal of Management Information Systems* 18(1): 23-55

Becker J, Rosemann M and Schütte R (1995) Grundsätze ordnungsmäßiger Modellierung. *Wirtschaftsinformatik* 37 (5): 435-445

Beckett D (2004) RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004, URL: http://www.w3.org/TR/rdf-syntax-grammar/ (last accessed: 2015-11-14)

Beckett D and Berners-Lee T (2008) Turtle – Terse RDF Triple Language. W3C Team Submission, URL: http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/ (last accessed: 2015-11-14)

Benjamins R and Aben M (1997) Structure-preserving knowledge-based system development through reusable libraries: a case study in diagnosis. *International Journal of Human-Computer Studies* 47 (2): 259-288

Bernaras A, Laresgoiti I and Corera J (1996) Building and Reusing Ontologies for Electrical Network Applications. In: Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96), Budapest, Hungary

Berners-Lee T, Hendler J and Lassila O (2001) The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* 284 34-43

Birmingham W and Klinker G (1993) Knowledge acquisition tools with explicit problem-solving models. *Knowledge Engineering Review* 8 (1): 5-25

Blázquez M, Fernández-López M, García-Pinar J and Gómez-Pérez A (1998) Building Ontologies at the Knowledge Level using the Ontology Design Environment. In: Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98), Banff, Alberta, Canada

Blomqvist E and Sandkuhl K (2005) Patterns in Ontology Engineering: Classification of Ontology Patterns. In: Proceedings of the 7th International Conference on Enterprise Information Systems (ICEIS'05), Miami Beach, Florida, USA

Boehm B (1988) A Spiral Model of Software Development and Enhancement. *IEEE Computer* 21 (5): 61-72

Boose J (1989) A survey of knowledge acquisition techniques and tool. *Knowledge Acquisition* 1 (1): 3-37

Borgo S, Guarino N and Masolo C (1996) Stratified Ontologies: The Case of Physical Objects. In: Proceedings of the Workshop on Ontology Engineering (ECAI'96), Budapest, Hungary

Borst W (1997) Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD Thesis, University of Enschede, The Netherlands

Bowersox D, Closs D and Stank T (1999) 21st Century Logistics: Making Supply Chain Integration Reality. Council of Logistics Management, Oak Brook, Illinois, USA

Braun C, Wortmann F, Hafner M and Winter R (2005) Method Construction – A Core Approach to Organizational Engineering. In: Proceedings of the 2005 ACM Symposium on Applied Computing (SAC'05), Santa Fe, New Mexico, USA

Breslin J, O'Sullivan D, Passant A and Vasiliu L (2010) Semantic Web computing in industry. *Computers in Industry* 61(8): 729-741

Brinkkemper S (1996) Method Engineering: Engineering of Information Systems Development Methods and Tools. *Information and Software Technology* 38 (4): 275-280

Brinkkemper S, Saeki M and Harmsen A (1998) Assembly Techniques for Method Engineering. In: Proceedings of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98), Pisa, Italy

Brockhaus (2005) Der Brockhaus. In zehn Bänden. F.A. Brockhaus Leipzig

Buchanan B, Barstow D, Bechtal R, Benett J, Clancey W, Kulikowsky C, Mitchell T and Waterman D (1983) Constructing an Expert System. In: Hayes-Roth F, Waterman D and Lenat D (Eds) Building expert systems. Addison Wesley, London, 127-167

Burns T and Stalker G (1961) The Management of Innovation. Tavistock Publications, London, UK

Bylander T and Chandrasekaran B (1987) Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies* 26 (2): 231-243

Byrd T, Cossick K and Zmud R (1992) A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. *Management Information Systems Quarterly* 16 (1): 117-138

Cardoso J (2007) The Semantic Web Vision: Where are We? *IEEE Intelligent Systems* 22 (5): 84-88

Cardoso J and Sheth A (Eds) (2006) Semantic Web Services, Processes and Applications (Semantic Web and Beyond). Springer, New York, USA

Caves R (1980) Industrial Organization, Corporate Strategy and Structure. *Journal of Economic Literature* 18 (1): 64-92

Caves R and Porter M (1977) From Entry Barriers to Mobility Barriers: Conjectural Decisions and Contrived Deterrence to New Competition. *The Quarterly Journal of Economics* 91 (2): 241-262

Chandra C and Tumanyan A (2007) Organisation and problem ontology for supply chain information support system. *Data and Knowledge Engineering* 61 (2): 263-280

Chandrasekaran B (1986) Generic Tasks in Knowledge-based Reasoning: High-Level Building Blocks for Expert System Design. *IEEE Expert* 1 (3): 23-30

Chandrasekaran B, Johnson T and Smith J (1992) Task Structure Analysis for Knowledge Modeling. *Communications of the Association of Computing Machinery* 35 (9): 124-137

Chandrasekaran B, Josephson J and Benjamins R (1999) What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems* 14 (1): 20-26

Chen P (1976) The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems* 1 (1): 9–36

Chi Y-L (2010) Rule-based ontological knowledge base for monitoring partners across supply networks. *Expert Systems with Applications* 37 (2): 1400–1407

Chorafas D (1990) Knowledge Engineering: Knowledge Acquisition, Knowledge Representation, the Role of the Knowledge Engineer, and Domains Fertile for Al Implementations. Van Nostrand Reinhold, New York, USA

Christopher M (2000) The Agile Supply Chain: Competing in Volatile Markets. *Industrial Marketing Management* 29 (1): 37-44

Christopher M (2005) Logistics and Supply Chain Management. Creating Value-adding Networks. Pearson Education, Edinburgh, UK

Cleal D and Heaton N (1988) Knowledge-Based Systems: Implications for Human-Computer Interfaces. John Wiley and Sons, New York, USA

Cockburn A (2008) Using both incremental and iterative development. *Software Engineering Technology* 21 (5): 27-30

Cooper M, Lambert D and Pagh J (1997) Supply Chain Management: More Than a New Name for Logistics. *International Journal of Logistics Management* 8 (1): 1-14

Corcho O, Fernández-López M and Gómez-Pérez A (2003) Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data and Knowledge Engineering* 46 (1): 41-64

Corcho O, Fernandéz-López M, Gómez-Pérez A and Vicente O (2002) WebODE: An integrated workbench for ontology representation, reasoning and exchange. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW’02), Siguenza, Spain

Council of Supply Chain Management Professionals (CSCMP) (2013) Supply Chain Management Terms and Glossary. URL: https://cscmp.org/research/glossary-terms (last accessed: 2015-11-14)

Cranefield S and Purvis M (1999) UML as an Ontology Modelling Language. In: Proceedings of the IJCAI Workshop on Intelligent Information Integration (IJCAI'99), Stockholm, Sweden

Cyert R and March J (1963) A Behavioral Theory of the Firm. Prentice Hall, Eaglewood Cliffs, New York, USA

Dabrowska E (2004) Language, Mind and Brain. Some Psychological and Neurological Constraints on Theories of Grammar. University Press, Edinburgh

d'Aquin M, Motta E, Sabou M, Angeletou S, Gridinoc L, Lopez V and Guidi D (2008) Toward a New Generation of Semantic Web Applications. *IEEE Intelligent Systems* 23 (3): 20-28

De Hoog R (1998) Methodologies for Building Knowledge-Based Systems: Achievements and Prospects. In: Liebowitz J (Ed) The Handbook of Applied Expert Systems. CRC Press, Boca Raton, Florida, Chapter 1-1

De Nicola A, Missikoff M and Navigli R (2009) A software engineering approach to ontology building. *Information Systems* 34 (2): 258-275

DeLone W and McLean E (1992) Information Systems Success: The Quest for the Dependent Variable. *Information Systems Journal* 3 (1): 60-95

Deutsches Institut für Normung e.V. (DIN) (Ed) (1989) Deutsche Norm, DIN 30781 Teil 1, Transportkette: Grundbegriffe. Beuth, Berlin, Germany

Donaldson L (2001) The Contingency Approach of Organizations. Sage Publications, Thousand Oaks, California, USA

Eckhardt A (1979) Strategien der organisatorischen Gestaltung. Konzeptionelle und empirische Grundlagen einer Gestaltungstheorie unter besonderer Berücksichtigung der Gestaltungsträger und der Formen ihrer Zusammenarbeit. Peter Lang, Frankfurt, Germany

Enterprise Project (1997) Enterprise Ontology Project. In: Artificial Intelligence Applications Institute URL: http://www.aiai.ed.ac.uk/project/enterprise/ (last accessed: 2015-11-14)

Ericsson K and Simon H (1993) Protocol Analysis – Verbal Reports as Data. Revised Edition. MIT Press, Cambridge, Massachusetts, USA

Eriksson H, Shahar Y, Tu S, Puerta A and Musen M (1995) Task modeling with reusable problem-solving methods. *Artificial Intelligence* 79 (2): 293-326

Etzioni A (1961) A Comparative Analysis of Complex Organizations: On Power, Involvement, and Their Correlates. Free Press, New York, USA

Euzenat J and Shvaiko P (2013) Ontology Matching. 2 ed., Springer, Heidelberg, Germany

Fadel F, Fox M and Grüninger M (1994) A Generic Enterprise Resource Ontology. In: Proceedings of the 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'94), West Virginia, USA

Farquahr A, Fikes R and Rice J (1997) The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies* 46 (6): 707-727

Fayez M, Rabelo L and Mollaghasemi M (2005) Ontology for supply chain simulation modeling. In: Proceedings of the 37th Winter Simulation Conference, Phoenix, Arizona, USA

Fayol H (1919) Administration Industrielle et Générale; prévoyance, organisation, commandement, coordination, controle. Dunod, Paris

Feigenbaum E (1977) The art of artificial intelligence: Themes and case studies of knowledge engineering. In: Proceedings of the 5th International Joint Conference on AI (IJCAI'77), Cambridge, Massachusetts, USA

Fensel D (2004) Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. 2. ed., Springer, Berlin, Germany

Fernandéz-López M, Gómez-Pérez A and Juristo N (1997) METHONTOLOGY: From Ontological Art Towards Ontology Engineering. In: Proceedings of the Spring Symposium Series on Ontology Engineering, Stanford, California, USA

Fernández-LópezM, Gómez-Pérez A, Pazos J and Pazos A (1999) Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. *IEEE Intelligent Systems* 14 (1): 37-46

Fisher M (1997) What is the Right Supply Chain for Your Product? *Harvard Business Review* 75 (2): 105-116

Fox M (1992) The TOVE Project: Towards a Common-Sense Model of the Enterprise. In: Belli F and Radermacher F (Eds) Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Springer, Berlin, 25-34

Fox M and Grüninger M (1998) Enterprise Modeling. *AI Magazine* 19 (3): 109-121

Fox M, Barbuceanu M and Grüninger M (1996) An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour. *Computers in Industry* 29 (1-2): 123-134

Fox M, Barbuceanu M, Grüninger M and Lin J (1998) An Organization Ontology for Enterprise Modelling. In: Prietula M, Carley K and Gasser L (Eds) Simulating Organisations: Computational Models of Institutions and Groups. AAAI/MIT, Menlo Park, 131–152

Fox M, Chionglo J and Fadel F (1993) A Common-Sense Model of the Enterprise. In: Proceedings of the 2nd Industrial Engineering Research Conference, Los Angeles, California, USA

Frank U (2000) Evaluation von Artefakten in der Wirtschaftsinformatik. In: Heinrich L and Häntschel I (Eds) Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. Oldenbourg, Munich, 35-48

Frank U (2005) Empirical Research Strategies in Conceptual Modeling – Silver Bullet or Academic Toys? *Wirtschaftsinformatik* 47 (2): 153-154

Freiling M, Alexander J, Messick S, Rehfuss S and Shulman S (1985) Starting a Knowledge Engineering Project. A Step-by-Step Approach. *AI Magazine* 6 (3): 150-164

Gammack J (1987) Different Techniques and Different Aspects on Declarative Knowledge. In: Kidd A (Ed) Knowledge Acquisition for Expert Systems: A Practical Guide. Plenum Press, New York, 137-163

Gangemi A (2005) Ontology Design Patterns for Semantic Web Content. In: Proceedings of the 4th International Semantic Web Conference (ISWC'05), Galway, Ireland

Gangemi A and Presutti V (2009) Ontology Design Patterns. In: Staab S and Studer R (Eds) Handbook on Ontologies. 2 ed., Springer, Heidelberg, 221-243

Gehlert A, Braun R and Esswein W (2004) Temporal Aspects in Business Processes – An Application to E-Government. In: Proceedings of Informationssysteme im E-Business und E-Government (EMISA'04), Luxemburg

Genesereth M and Fikes R (1992) Knowledge Interchange Format. Version 3.0. Reference Manual. In: Technical Report Logic-92-1, Computer Science, Stanford University, California, USA

Gilchrist A (2003) Thesauri, taxonomies and ontologies – an etymological note. *Journal of Documentation* 59 (1): 7-19

Glass R (2000) Process Diversity and a Computing Old Wives'/Husbands' tale. *IEEE Software* 17 (4): 127-128

Gómez-Pérez A (2004) Ontology Evaluation. In: Staab S and Studer R (Eds) Handbook on Ontologies Springer, Berlin, 251-274

Gómez-Pérez A and Rojas-Amaya M (1999) Ontological Reengineering for Reuse. In: Proceedings of 11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'99), Dagstuhl Castle, Germany

Gómez-Pérez A, Fernandez A and De Vicente M (1996) Towards a Method to Conceptualize Domain Ontologies. In: Working notes of the workshop on Ontology Engineering (ECAI'96), Budapest, Hungary

Gómez-Pérez A, Fernández-López M and Corcho O (2004) Ontological Engineering with examples from the area of Knowledge Management, e-Commerce, and Semantic Web. Springer, London, UK

Gonnet S, Vegetti M, Leone H and Henning G (2007) SCOntology: A Formal Approach toward a Unified and Integrated View of the Supply Chain. In: Cunha M, Cortes B and Putnik G (Eds) Adaptive technologies and business integration: social, managerial and organizational dimensions. IGI Global, Hershey, 137–158

Goodhue D and Thompson R (1995) Task-Technology Fit and Individual Performance. *MIS Quarterly* 19 (2): 213-236

Grau B, Horrocks I, Motik B, Parsia B, Patel-Schneider P and Sattler U (2008) OWL 2: The Next Step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (4): 309-322

Gregor S (2006) The Nature of Theory in Information Systems. *Management Information Systems Quarterly* 30 (3): 611-642

Gregor S and Jones D (2007) The Anatomy of a Design Theory. *Journal of the Association of Information Systems* 8 (5): 312-335

Gregory R and Muntermann J (2011) Theorizing in Design Science Research: Inductive versus Deductive Approaches. In: Proceedings of the 32nd International Conference on Information Systems (ICIS'11), Shanghai, China

Grochla E (1995) Grundlagen der organisatorischen Gestaltung. Schäffer-Poeschel, Stuttgart

Grosso W, Eriksson H, Fergerson R, Gennari J, Tu S and Musen M (1999) Knowledge Modeling at the Millennium (The Design and Evolution of Protégé-2000). In: Proceedings of the Knowledge Acquisition Workshop (KAW'99), Banff, Canada

Gruber T (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5 (2): 199-220

Gruber T (1995) Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies* 43 (5/6): 907-928

Gruber T and Cohen P (1987) Design for acquisition: principles of knowledge-system design to facilitate knowledge acquisition. *International Journal of Man-Machine Studies* 26 (2): 143-159

Gruber TR (2009) Ontology. In: Liu L and Öszu T (Eds) Encyclopaedia of Database Systems Springer, Berlin, 1963-1965

Grubic T, Veza I and Bilic B (2011) Integrating process and ontology to support supply chain modelling. *International Journal of Computer Integrated Manufacturing* 24 (9): 847-863

Grüninger M and Fox M (1995) Methodology for the Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI'95), Montreal, Canada

Grüninger M and Lee J (2002) Ontology Applications and Design. *Communications of the Association of Computing Machinery* 45 (2): 39-41

Grüninger M and Pinto J (1995) A Theory of Complex Actions for Enterprise Modelling. In: Proceedings of the AAAI Spring Symposium Series 1995 – Extending Theories of Action: Formal Theory and Practical Applications, Stanford, California, USA

Guarino N (1995) Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies* 43 (5-6): 625-640

Guarino N (1997) Semantic Matching: Formal Ontological Distinctions for Information Organisation, Extraction, and Integration. In: Pazienza M (Ed) Information Extraction: A

Multidisciplinary Approach to an Emerging Information Technology. Springer, London, 139-170

Guarino N (1998) Formal Ontology and Information Systems. In: Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS'98), Trento, Italy

Guarino N and Giaretta P (1995) Ontologies and Knowledge Bases - Towards a Terminological Clarification. In: Mars N (Ed) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam, 25-32

Guarino N, Oberle D and Staab S (2009) What is an Ontology? In: Staab Sand Studer R (Eds) Handbook on Ontologies. 2. ed., Springer, Berlin, Germany

Gudehus T (2010) Logistik. Grundlagen – Strategien - Anwendungen. 4. ed., Springer, Heidelberg, Germany

Gupta D and Prakash N (2001) Engineering Methods from Method Requirements Specifications. *Requirements Engineering* 6 (3): 135-160

Gutierrez O (1987) Some Aspects of Information Requirements Analysis Using A Repertory Grid Technique. In: Galliers R (Ed) Information Analysis: Selected Readings. Addison-Wesley, Reading, Massachusetts, 347-362

Gutzwiller T (1994) Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. PhD Thesis, Hochschule St. Gallen für Wirtschafts-, Rechts- und Sozialwissenschaften, Switzerland

Harland C (1996) Supply Chain Management: Relationships, Chains and Networks. *British Journal of Management* 7 (Special Issue): 63-80

Harmon P and King D (1985) Expert Systems: Artificial Intelligence in Business. Wiley, New York, USA

Hayes P (2004) RDF Semantics. W3C Recommendation 10 February 2004, URL: http://www.w3.org/TR/rdf-mt/ (last accessed: 2015-11-14)

Hayes-Roth F, Waterman D and Lenat D (1983) Building Expert Systems. Addison-Wesley, New York, USA

Haynes S and Carroll J (2010) The Range and Role of Theory in Information Systems Design Research: From Concepts to Construction. In: Proceedings of the 31st International Conference on Information Systems (ICIS'10), St. Louis, USA

Hazmann M, El-Beltagy S and Rafea A (2011) A Survey of Ontology Learning Approaches. *International Journal of Computer Applications* 22 (9): 36-43

Heinrich L, Heinzl A and Roithmayr F (2004) Wirtschaftsinformatik-Lexikon. 7. ed., Oldenbourg, Munich, Germany

Hendersson-Sellers B and Ralyté J (2010) Situational Method Engineering: State-of-the-Art Review. *Journal of Universal Computer Science* 10 (3): 424-478

Hepp M (2007) Ontologies: State of the Art, Business Potential, and Grand Challenges. In: Hepp M, De Leenheer P, De Moor A and Sure Y (Eds) Ontology Management: Semantic Web, Semantic Web Services, and Business Applications. Springer, Berlin, 3-22

Hevner A, March S, Park J and Ram S (2004) Design Science in Information System Research. *Management Information Systems Quarterly* 28 (1): 75-105

Hill W, Fehlbaum R and Ulrich P (1994a) Organisationslehre 1. Ziele, Instrumente und Bedingungen der Organisation sozialer Systeme. 5. ed., Paul Haupt, Bern, Switzerland

Hill W, Fehlbaum R and Ulrich P (1994b) Organisationslehre 2. Theoretische Ansätze und praktische Methoden der Organisation sozialer Systeme. 5. ed., Paul Haupt, Bern, Switzerland

Hitzler P, Krötzsch M, Parsia B, Patel-Schneider P and Rudolph S (2012) OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation 11 December 2012, URL: http://www.w3.org/2012/pdf/REC-owl2-primer-20121211.pdf (last accessed: 2015-11-14)

Holsapple C and Joshi K (2002) A Collaborative Approach to Ontology Design. *Communications of the Association of Computing Machinery* 45 (2): 42-47

Hori M, Euzenat J and Patel-Schneider P (2003) OWL Web Ontology Language XML Presentation Syntax. W3C Note 11 June 2003, URL: http://www.w3.org/TR/owl-xmlsyntax/ (last accessed: 2015-11-14)

Horridge M (2010) OWLViz – A visualisation plug-in for the Protégé OWL Plug-in. URL: http://protegewiki.stanford.edu/wiki/OWLViz (last accessed: 2015-11-14)

Horridge M (2011) A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. Edition 1.3. The University of Manchester, URL: http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/ (last accessed: 2015-11-14)

Horridge M and Patel-Schneider P (2009) OWL 2 Web Ontology Language Manchester Syntax. W3C Working Group Note 27 October 2009, URL: http://www.w3.org/TR/2009/NOTE-OWL2-manchester-syntax-20091027/ (last accessed: 2015-11-14)

Horrocks I, Kutz O and Sattler U (2006) The Even More Irresistible SROIQ. In: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06), Lake District of the UK

House E (1993) Professional Evaluation: Social Impact and Political Consequences. Sage, Newbury Park, California, USA

Hoxha J, Scheuermann A, Bloehdorn S (2010) An Approach to Formal and Semantic Representation of Logistics Services. In: Proceedings of the ECAI'10 Workshop on Artificial Intelligence and Logistics (AILOG 2010), pp. 73-78

Huang G, Lau S and Mak K (2003) The impacts of sharing production information on supply chain dynamics: A review of the literature. *International Journal of Production Research* 41 (7): 1483-1517

IEEE (1990) IEEE Standard Glossary of Software Engineering Terminology. In: IEEE Computer Society, New York, USA

IEEE (1997) IEEE Standard for Developing Software Life Cycle Processes. In: IEEE Computer Society, New York, USA

Ihde G (2001) Transport, Verkehr, Logistik. Gesamtwirtschaftliche Aspekte und einzelwirtschaftliche Handhabung. 3. ed., Vahlen, Munich, Germany

IS Theory (2012) Theories Used in IS Research Wiki. Brigham Young University (BYO), Provo, Utah. URL: http://istheory.byu.edu/wiki/Main_Page (last accessed: 2015-11-14)

Jacobson I, Booch G and Rumbaugh J (1999) The Unified Software Development Process. Addison Wesley, Amsterdam, The Netherlands

Jarrar M and Meersman R (2002) Formal Ontology Engineering in the DOGMA Approach. In: Proceedings of the 1st International Conference on Ontologies, Databases and Applications of Semantics (ODBase'02), Irvine, California, USA

Jarrar M and Meersman R (2008) Ontology Engineering – The DOGMA Approach. In: Dillon TS, Chang E, Meersman R and Sycara K (Eds) Advances in Web Semantics I. Ontologies, Web Services and Applied Semantic Web. Springer, Berlin, 7-34

Kaczmarek M (2006) Modellbasierte Gestaltung von Supply Chains. Ein prozess- und simulationsorientierter Ansatz. Verlag Dr. Kovac, Hamburg, Germany

Karbach W and Linster M (1990) Wissensakquisition für Expertensysteme. Techniken, Modelle und Softwarewerkzeuge. Carl Hanser, Munich, Germany

Kastens U and Kleine Büning K (2008) Modellierung. Grundlagen und formale Methoden. Hanser, Munich, Germany

Katifori A, Halatsis C, Lepouras G, Vassilakis C and Giannopoulou E (2007) Ontology Visualization Methods - A Survey. *Association of Computing Machinery Computing Surveys* 39 (4): Article 10

Ketchen Jr. D and Hult G (2007) Bridging organization theory and supply chain management: The case of best value supply chains. *Journal of Operations Management* 25 (2): 573-580

Kettinger W, Teng JT and Guha S (1997) Business Process Change: A Study of Methodologies, Techniques, and Tools. *Management Information Systems Quarterly* 21 (1): 55-98

Khazanchi D (2005) Information Technology (IT) Appropriateness: The Contingency Theory of "Fit" and IT Implementation in Small and Medium Enterprises. *Journal of Computer Information Systems* 45(3): 88-95

Kidd A (1987) Knowledge Acquisition – An Introductory Framework. In: Kidd A (Ed) Knowledge Acquisition for Expert Systems: A Practical Handbook. Plenum Press, New York, USA

Kieser A (2006) Der Situative Ansatz. In: Kieser A and Ebers M (Eds) Organisationstheorien. Kohlhammer, Stuttgart, 215-245

Kieser A and Kubicek H (1992) Organisation. 3. ed., de Gruyter, Berlin, Germany

Kieser A and Walgenbach P (2007) Organisation. 5. ed., Schäffer-Poeschel, Stuttgart, Germany

Kifer M, Lausen G and Wu J (1995) Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the Association of Computing Machinery* 42 (4): 741-843

Kim H and Fox M (1995) An Ontology of Quality for Enterprise Modelling. In: Proceedings of the 4th Workshop on Enabling Technologies-Infrastructures for Collaborative Enterprises, Morgantown, West Virginia, USA

Kim H, Fox M and Grüninger M (1999) An Ontology for Quality Management - Enabling Quality Problem Identification and Tracing. *BT Technology Journal* 17 (4): 131-140

Kirn S, Anhalt C, Bieser T, Jacob A and Klein A (2008) Individualisierung von Sachgütern und Dienstleistungen durch Adaptivität von Wertschöpfungssystemen in Raum, Zeit und Ökonomie. In: Kirn S (Ed) Individualization Engineering - Gestaltung adaptiver Wertschöpfungsketten für individualisierte Sachgüter und Dienstleistungen. Cuvillier Verlag, Göttingen, 3-60

Kishore R, Sharman R and Ramesh R (2004a) Computational Ontologies and Information Systems I: Foundations. *Communications of the Association for Information Systems* 14 (1): 158-183

Kishore R, Zhang H and Ramesh R (2004b) A Helix-Spindle Model for Ontological Engineering. *Communications of the Association of Computing Machinery* 47 (2): 69-75

Klaas T (2002) Logistik-Organisation. Ein konfigurationstheoretischer Ansatz zur logistikorientierten Organisationsgestaltung. Deutscher Universitätsverlag, Wiesbaden, Germany

Kosiol E (1962) Organisation der Unternehmung. Gabler, Wiesbaden, Germany

Kuhlen R (1995) Informationsmarkt. Chancen und Risiken der Kommerzialisierung von Wissen. Universitätsverlag, Konstanz, Germany

Kühne T (2006) Matters of (Meta-) Modeling. *Journal on Software and Systems Modeling* 5 (4): 369-385

Lambert D and Cooper W (2000) Issues in Supply Chain Management. *Industrial Marketing Management* 29 (1): 65-83

Lemoine Q and Dagnaes L (2003) Globalisation strategies and business organisation of a network of logistics service providers. *International Journal of Physical Distribution & Logistics Management* 33 (3): 209-228

Lenat D and Guha R (1990) Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison-Wesley, Boston, Massachusetts, USA

Leukel J and Kirn S (2008) A Supply Chain Management Approach to Logistics Ontologies in Information Systems. In: Proceedings of the 11th International Conference on Business Information Systems (BIS’08), Innsbruck, Austria

Li Z, Yang M and Ramani K (2009) A methodology for engineering ontology acquisition and validation. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23 (1): 37-51

Lin H and Harding J (2007) A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry* 58 (5): 428-437

Lin H, Harding J and Shahbaz M (2004) Manufacturing system engineering ontology for semantic interoperability across extended project teams. *International Journal of Production Research* 42 (24): 5099-5118

Lin J, Fox M and Bilgic T (1996) A Requirement Ontology for Engineering Design. *Concurrent Engineering: Research and Application* 4 (4): 279-291

Lin J, Wang G, Hu Z and Long Q (2012) A Supply Chain Architecture Based on Ontology for Distributed Simulation and Modeling. *International Journal of Digital Content Technology and its Applications* 6 (5): 225-234

Litwak E (1961) Models of Bureaucracy Which Permit Conflict. *American Journal of Sociology* 67 (2): 177-184

Lorenz K (1995) Methode. In: Mittelstrass J (Ed) Enzyklopadie Philosophie und Wissenschaftstheorie Metzler, Stuttgart, 876-879

Lu Y, Panetto H and Gu X (2010) Ontology Approach for the Interoperability of Networked Enterprises in Supply Chain Environment. In: Proceedings of the 5th IFAC/IFIP Workshop on Enterprise Integration, Interoperability and Networking (EI2N'10), Hersonissou, Crete, Greece

MacGregor R (1991) Inside the LOOM Description Classifier. *SIGART Bulletin* 2 (3): 70-76

Madni A, Lin W and Madni C (2001) IDEON^{TM}: An extensible ontology for designing, integrating and managing collaborative distributed enterprises. *Systems Engineering* 4 (1): 35-48

Maiden N and Hare M (1998) Problem domain categories in requirements engineering. *International Journal of Human-Computer Studies* 49 (3): 281-304

March J and Olsen J (1976) Ambiguity and Choice in Organizations. Universitetsforlaget, Bergen, Norway

March J and Simon H (1958) Organizations. John Wiley & Sons, New York, USA

March S and Smith G (1995) Design and natural science research on information technology. *Decision Support Systems* 15 (4): 251-266

Marcus S (1988) SALT: A Knowledge Acquisition Tool for Propose-and-revise Systems. In: Marcus S (Ed) Automating Knowledge Acquisition for Experts Systems. Kluwer Academic Publisher, Boston 80-123

Marcus S and McDermott J (1989) SALT: a knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence* 39 (1): 1-38

Marcus S, Stout J and McDermott J (1988) VT: An Expert Elevator Designer That Uses Knowledge-Based Backtracking. *AI Magazine* 9 (1): 95-112

McGuiness D (2003) Ontologies Come of Age. In: Fensel D, Hendler J, Liebermann H and Wahlster W (Eds) Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential, MIT Press, 171-194

McGuiness D and van Harmelen F (2004) OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004, URL: http://www.w3.org/TR/2004/REC-owl-features-20040210/ (last accessed: 2015-11-14)

McIlraith S, Son T and Zeng H (2001) Semantic Web Services. *Intelligent Systems* 16 (2): 46-53

Mentzer J, DeWitt W, Keebler J, Min S, Nix N, Smith C and Zacharia Z (2001) Defining Supply Chain Management. *Journal of Business Logistics* 22 (2): 1-25

Meyer M and Booker J (2001) Eliciting and Analyzing Expert Judgment: A Practical Guide. Academic Press, London, UK

Mintzberg H (1992) Die Mintzberg-Struktur: Organisationen effektiver gestalten. Moderne Industrie, Landsberg am Lech, Germany

Mizoguchi R (2003) Tutorial on ontological engineering. Part 1: Introduction to Ontological Engineering. *New Generation Computing* 21 (4): 365-384

Mizoguchi R, Vanwelkenhuysen J and Ikeda M (1995) Task Ontology for Reuse of Problem Solving Knowledge. In: Mars N (Ed) Towards Very Large Knowledge Bases, IOS Press, Amsterdam, 46-59

Morik K (1991) Underlying assumptions of knowledge acquisition and machine learning. *Knowledge Acquisition* 3 (2): 137-156

Motik B, Patel-Schneider P and Grau B (2012) OWL 2 Web Ontology Language Direct Semantics (Second Edition). W3C Recommendation 11 December 2012, URL: http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/ (last accessed: 2015-11-14)

Motik B, Patel-Schneider P and Parsia B (2009) OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation 27 October 2009, URL: http://www.w3.org/TR/2009/REC-OWL2-syntax-20091027/ (last accessed: 2015-11-14)

Motta E (1999) Reusable Components for Knowledge Modeling. Case Studies in Parametric Design. IOS Press, Amsterdam, The Netherlands

Motta E (Ed) (2013) 25 Years of Knowledge Acquisition. *International Journal of Human-Computer Studies* 71 (2)

Motta E, Mulholland P, Peroni S, D'Aquin M, Gómez-Pérez J, Mendez V and Zablith F (2011) A Novel Approach to Visualizing and Navigating Ontologies. In: International Semantic Web Conference (ISWC'11), Bonn, Germany

Musen M (1993) An Overview of Knowledge Acquisition. In: David J-M, Krivine J and Simmons R (Eds) Second Generation Experts Systems. Springer, New York, 405-427

Network Ontologies Toolkit (NeOn) (2012) NeOn Toolkit. URL: http://neon-toolkit.org/wiki/Main_Page.html (last accessed: 2015-11-14)

Newell A (1982) The knowledge level. *Artificial Intelligence* 18 (1): 87-127

Nickerson R, Varshney U and Muntermann J (2012) A method for taxonomy development and its application in information systems. *European Journal of Information Systems* 22(3): 336-359

Nonaka I (1991) The Knowledge-Creating Company. *Harvard Business Review* 69 (November-December): 96-104

Nonaka I (1994) A Dynamic Theory of Organisational Knowledge Creation. *Organisation Science* 5 (1): 14-37

Nordsieck F (1934) Grundlagen der Organisationslehre. Carl Ernst Poeschel, Stuttgart, Germany

Noy N and Klein M (2004) Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems* 6 (4): 428-440

Noy N and McGuiness D (2001) Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report, Stanford Medical Informatics Report SMI-2001-0880, Stanford University, Stanford, California, USA

Noy N and Rector A (2006) Defining N-ary Relations on the Semantic Web. In: W3C Working Group Note, URL: http://www.w3.org/TR/swbp-n-aryRelations/ (last accessed: 2015-11-14)

Obrst L, Ceusters W, Mani I, Ray S and Smith B (2007) The Evaluation of Ontologies. Toward Improved Semantic Interoperability. In: Baker C and Cheung K-H (Eds) Revolutionizing Knowledge Discovery in the Life Sciences. Springer, Berlin, 139-158

Oesterle H, Becker J, Frank U, Hess T, Karagiannis D, Krcmar H, Loos P, Mertens P, Oberweis A and Sinz E (2010) Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Zeitschrift für betriebswirtschaftliche Forschung* 62 (6): 664-672

Ogden C and Richards I (1923) The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. 10. ed., Routledge & Kegan Paul Ltd., London, UK

Pawlaszczyk D, Dietrich A, Timm I, Otto S and Kirn S (2004) Ontologies Supporting Cooperations in Mass Customization – A Pragmatic Approach. In: Proceedings of the International Conference on Mass Customization and Personalization – Theory and Practice in Central Europe, Rzeszów, Poland

Payne P, Mendonca E, Johnson S and Starren J (2007) Conceptual knowledge acquisition in biomedicine: A methodological review. *Journal of Biomedical Informatics* 40 (5): 582-602

Peffers K, Tuunanen T, Rothenberger M, Chatterjee S (2008) A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24 (3): 45–77

Pfohl H-C (2010) Logistiksysteme. Betriebswirtschaftliche Grundlagen. 8. ed., Springer, Berlin, Heidelberg, Germany

Poek K and Gappa U (1993) Making Role-Limiting Shells More Flexible. In: Proceedings of the 7th Knowledge Acquisition Workshop (EKAW'93), Toulouse, France

Porter M (1981) The Contributions of Industrial Organization To Strategic Management. *The Academy of Management Review* 6 (4): 609-620

Prahalad C and Hamel G (1990) The Core Competence of the Corporation. *Harvard Business Review* 68 (3): 79-91

Presutti V and Gangemi A (2008) Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In: Proceedings of the 27th International Conference on Conceptual Modeling (ER'08), Barcelona, Spain

Probst G, Raub S and Romhardt K (2006) Wissen Managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. 5. ed., Gabler, Wiesbaden, Germany

Protégé (2012) The Protégé Ontology Editor and Knowledge Acquisition System. Stanford Center for Biomedical Informatics Research, Stanford University School of Medicine. URL: http://protege.stanford.edu/ (last accessed: 2015-11-14)

Puerta A, Egar J, Tu S and Musen M (1992) A multiple-method knowledge acquisition shell for the automatic generation of knowledge acquisition tools. *Knowledge Acquisition* 4 (2): 171-196

Pugh D (1981) The Aston program perspective. The Aston program of research. Retrospect and prospect. In: Van De Ven A and Joyce WF (Eds) Perspectives on Organization Design and Behavior. Wiley, New York, 155-166

Pugh D and Hickson D (1971) Eine dimensionale Analyse bürokratischer Strukturen. In: Mayntz R (Ed) Bürokratische Organisation. 2. ed., Kiepenhauer & Witsch, Köln, 82-93

Pugh D and Hickson D (Eds) (1976) Organizational Structure in Its Context. The Aston Programme I. Saxon House, Westmead, Farnborough, UK

Pugh D, Hickson D, Hinings C, MacDonald K, Turner C and Lupton T (1963) A Conceptual Scheme for Organizational Analysis. *Administrative Science Quarterly* 8 (3): 289-315

Puppe F, Gappa U, Poeck K and Bamberger S (1996) Wissensbasierte Diagnose- und Informationssysteme. Mit Anwendungen Des Expertensystem-Shell-Baukastens D3. Springer, Berlin, Germany

Rai A, Patnayakuni R and Seth N (2006) Firm performance impacts of digitally enabled supply chain integration capabilities. *MIS Quarterly* 30 (2): 225–246.

Rajpathak D and Chougule R (2011) A generic ontology development framework for data integration and decision support in a distributed environment. *International Journal of Computer Integrated Manufacturing* 24 (2): 154-170

Ralyté J, Rolland C and Deneckère R (2003) Towards a Generic Model for Situational Method Engineering. In: Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE'03), Klagenfurt/Velden, Austria

Ralyté J, Rolland C and Deneckère R (2004) Towards a Meta-tool for Change-Centric Method Engineering: A Typology of Generic Operators. In: Proceeding of the 16th International Conference on Advanced Information Systems Engineering (CAiSE'04), Riga, Lativa

Rehäuser J and Krcmar H (1996) Wissensmanagement im Unternehmen. In: Schreyögg G and Conrad P (Eds) Managementforschung 6: Wissensmanagement. DeGruyter, Berlin, 1-40

Roethlisberger F and Dickson W (1939) Management and the Worker. Harvard University Press, Cambridge, Massachusetts, USA

Ruggaber R (2006) ATHENA – Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications. In: Konstantas D, Bourrières J-P, Léonard M and Boudjilida N (Eds) Interoperability of Enterprise Software and Applications. Springer, London, 459-460

Scheuermann A (2011a) Semantic Technology for Intelligent Logistics Information Systems – An Application Ontology to Enhance Planning of Customer-Tailored Supply Chains. Doctoral Consortium 34th Annual German Conference on Artificial Intelligence (KI 2011). 04.10.2011, Berlin, Germany

Scheuermann A (2011b) Semantic Technology for Intelligent Logistics Information Systems – An Application Ontology to Enhance Planning Capabilities for Customer-Tailored Supply Chains. The 8th Summer School on Ontology Engineering and the Semantic Web (SSSW'11). 10.-16.07.2011, Cercedilla, Madrid, Spain

Scheuermann A and Hoxha J (2012) Ontologies for Intelligent Provision of Logistics Services. In: Proceedings of the 7th International Conference on Internet and Web Applications and Services (ICIW'12), Stuttgart, Germany, pp. 106-111

Scheuermann A und Leukel J (2013) Task Ontology for Supply Chain Planning – A Literature Review. International Journal of Computer Integrated Manufacturing 27 (8) 719-732

Scheuermann A und Leukel J (2014) Supply Chain Management Ontology from an Ontology Engineering Perspective. Computers in Industry 65 (6) 913-923

Scheuermann A und Obermann J (2014) Status Quo of Ontology Learning From Unstructured Knowledge Sources for Knowledge Management. Extended Version. In: E. Mercier-Laurent E. and Boulanger D. (Eds.): AI4KM 2012, IFIP AICT 422, IFIP International Federation for Information Processing, pp. 72-94

Scheuermann A, Motta E, Mulholland P, Gangemi A und Presutti V (2013) An Empirical View on Representing Temporal Information. In: Proceedings of the Seventh International Conference on Knowledge Capture (K-Cap 2013), Banff, Canada, pp. 89-96

Schneider M (2012) OWL 2 Web Ontology Language RDF-Based Semantics. W3C Recommendation 11 December 2012, URL: http://www.w3.org/TR/owl2-rdf-based-semantics/ (last accessed: 2015-11-14)

Scholz-Reiter B, Toonen C and Windt K (2008) Logistikdienstleistungen. In: Arnold D, Isermann H, Kuhn A, Tempelmeier H and Furmans K (Eds) Handbuch Logistik. 3. ed., Springer, Berlin, 581-607

Schreiber G, Akkermans H, Anjewierden A, de Hoog R, Shadbolt N, Van der Velde W and Wielinga B (2002) Knowledge Engineering and Management. The CommonKADS Methodology. MIT Press, Cambridge, Massachusetts, USA

Schreiber G, Wielinga B and Breuker J (Eds) (1993) KADS. A Principled Approach to Knowledge-Based System Development. Academic Press, London, UK

Schreiber G, Wielinga B, De Hoog R, Akkermans H and van de Velde W (1994) CommonKADS: A Comprehensive Methodology for KBS Development. *IEEE Expert* 9 (6): 28-37

Schreyögg G (2008) Organisation. Grundlagen moderner Organisationsgestaltung. 5. ed., Gabler, Wiesbaden, Germany

Schreyögg G and Geiger D (2003) Wenn alles Wissen ist, ist Wissen am Ende nichts?! *DBW - Die Betriebswirtschaft* 63 (1): 7-22

Schulte C (2009) Logistik. Wege zur Optimierung der Supply Chain. 5. ed., Vahlen, Munich, Germany

Schulte-Zurhausen M (2010) Organisation. 5. ed., Vahlen, Munich, Germany

Schütte R (1997) Die neuen Grundsätze ordnungsmäßiger Modellierung. In: Forschungsforum Leipzig, Leipzig, Germany

Schütte R (1998) Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsfähiger Modelle. Deutscher Universitätsverlag, Wiesbaden, Germany

Seddon P (1997) A Respecification and Extension of the DeLone and McLean Model of IS Success. *Information System Research* 8 (3): 240-253

Shamsfard M and Barforoush A (2003) The State of the Art in Ontology Learning: A Framework for Comparison. *The Knowledge Engineering Review* 18 (4): 293-316

Sharman R, Kishore R and Ramesh R (2004) Computational Ontologies and Information Systems II: Formal Specification. *Communications of the Association for Information Systems* 14 (1): 184-205

Shaw M and Gaines B (1987) An Interactive Knowledge Elicitation Technique Using Personal Construct Technology. In: Kidd A (Ed) Knowledge Acquisition for Expert Systems: A Practical Handbook. Plenum Press, New York, 109-136

Shvaiko P and Euzenat J (2012) Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering* 25 (1): 158-176

Simchi-Levi D, Kaminsky P and Simchi-Levi E (2003) Designing & Managing the Supply Chain: Concepts, Strategies & Case Studies. 2. ed., McGraw-Hill, Boston, Massachusetts, USA

Simon H (1957) Administrative Behavior. Macmillan, New York, USA

Simon H (1996) The Sciences of the Artificial. 4. ed., MIT Press, Cambridge, Massachusetts, USA

Simperl E (2009) Reusing ontologies on the Semantic Web: A feasibility study. *Data & Knowledge Engineering* 68 (10): 905-925

Simperl E, Mochol M and Bürger T (2010) Achieving Maturity: the State of Practice in Ontology Engineering in 2009. *International Journal of Computer Science and Applications* 7 (1): 45-65

Singh N (2003) Emerging Technologies to Support Supply Chain Management. *Communications of the Association of Computing Machinery* 46 (9): 243-247

Smith B (2003) Ontology. In: Floridi L (Ed) Blackwell Guide to the Philosophy of Computing and Information. Blackwell, Malden, Massachusetts, 155-166

Soares A, Azevedo A and De Sousa J (2000) Distributed planning and control systems for the virtual enterprise: organisational requirements and development life-cycle. *Journal of Intelligent Manufacturing* 11 (3): 253-270

Sowa J (2000) Knowledge Representation. Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove, California, USA

Speel P-H and Aben M (1998) Preserving conceptual structures in design and implementation of industrial KBSs. *International Journal of Human-Computer Studies* 49 (4): 547-575

Staab S and Studer R (Eds) (2009) Handbook on Ontologies. 2. ed., Springer, Berlin, Germany

Staab S, Schnurr H, Studer R and Sure Y (2001) Knowledge Processes and Ontologies. *IEEE Intelligent Systems* 16 (1): 26-34

Stachowiak H (1973) Allgemeine Modelltheorie. Springer, Wien

Stahlknecht P and Hasenkamp U (2005) Einführung in die Wirtschaftsinformatik. 11. ed., Springer, Berlin, Germany

Straube F (2004) e-Logistik. Ganzheitliches Logistikmanagement. Springer, Berlin, Germany

Studer R, Benjamins R and Fensel D (1998) Knowledge engineering: Principles and methods. *Data and Knowledge Engineering* 25 (1-2): 161-197

Suárez-Figueroa M (2010) NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. PhD Thesis, Universidad Politécnica De Madrid Facultad de Informática Departamento de Intelligigencia Artificial

Swartout W, Patil R, Knight K and Russ T (1997) Toward Distributed Use of Large-Scale Ontologies. In: Proceedings of the Spring Symposium Series on Ontology engineering, Stanford, California, USA

Sweller J (1988) Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science* 12 (2): 257-285

Sydow J (2010) Management von Netzwerkorganisationen – Zum Stand der Forschung. In: Sydow J (Ed) Management von Netzwerkorganisationen. 5. ed., Gabler, Wiesbaden, 373-458

Talaulicar T (2004) Wissen. In: Schreyögg G and von Werder A (Eds) Handwörterbuch Unternehmensführung und Organisation. Schaeffer-Poeschel, Stuttgart, 1640-1647

Tang Y, Yan C and Suen C (1994) Document Processing for Automatic Knowledge Acquisition. *IEEE Transactions on Knowledge and Data Engineering* 6 (1): 3-21

Taylor F (1911) The Principles of Scientific Management. Harper and Brothers, New York, USA

Tham K, Fox M and Grüninger M (1994) A Cost Ontology for Enterprise Modelling. In: Proceedings of the 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, West Virginia, USA

Thonemann U (2010) Operations Management. Konzepte, Methoden und Anwendungen. 2. ed., Pearson Education, Munich, Germany

TOVE (2002) TOVE Ontology Project. Enterprise Integration Laboratory, URL: http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html (last accessed: 2015-11-14)

Tu S, Eriksson H, Gennari J, Shahar Y and Musen M (1995) Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTÉGÉ-II to protocol-based decision support. *Artificial Intelligence in Medicine* 7 (3): 257-289

Uschold M and Grüninger M (1996) Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review* 11 (2): 93-155

Uschold M and Grüninger M (2004) Ontologies and Semantics for Seamless Connectivity. *SIGMOND Record* 33 (4): 58-64

Uschold M and King M (1995) Towards a Methodology for Building Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI'95), Montreal, Canada

Uschold M, King M, Moralee S and Zorgios Y (1998) The Enterprise Ontology. *The Knowledge Engineering Review* 13 (1): 31-89

van Heijst G, Schreiber G and Wielinga B (1997a) Roles are not classes: a reply to Nicola Guarino. *International Journal of Human-Computer Studies* 46 (2-3): 311-318

van Heijst G, Schreiber G and Wielinga B (1997b) Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies* 45 (2/3): 183-292

Vegetti M, Leone H and Henning G (2011) PRONTO: An ontology for comprehensive and consistent representation of product information. *Engineering Applications of Artificial Intelligence* 24 (8): 1305-1327

Vessey I (1991) Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature. *Decision Sciences* 22 (2): 219-240 von Krogh G and Grand S (2004) Wissensmanagement. In: Schreyögg G and von Werder A (Eds) Handwörterbuch Unternehmensführung und Organisation. Schaeffer-Poeschel, Stuttgart, col. 1648-1656

Vrandecic D (2010) Ontology Evaluation. PhD Thesis, Karlsruhe Institute of Technology (KIT), Germany

W3C OWL Working Group (2012) OWL 2 Web Ontology Language. Document Overview (Second Edition). W3C Recommendation 11 December 2012, URL: http://www.w3.org/TR/owl2-overview/ (last accessed: 2015-11-14)

Waldron V (1986) Interviewing for knowledge. *IEEE Transactions on Professional Communications* 29 (2): 31-34

Wang C and Chan C (2001) Ontology Modelling Using UML. In: Proceedings of the 7th International Conference on Object Oriented Information Systems Conference (OOIS'01), Geneva, Switzerland

Waterman D (1986) A Guide to Expert Systems. Addison-Wesley, Boston, Massachusetts, USA

Weber M (1922) Wirtschaft und Gesellschaft. Grundriss der verstehenden Soziologie. 1. ed., Mohr, Tübingen, Germany

Weber M (1972) Wirtschaft und Gesellschaft. Grundriss der verstehenden Soziologie. 5. ed., Mohr, Tübingen, Germany

Wendt O, Stockheim T, Grolik S and Schwind M (2003) Distributed Ontology Management. Prospects and Pitfalls on Our Way Towards a Web of Ontologies. In: Dagstuhl Workshop (Event Nr. 02212) – DFG-SPP 1083 Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien, Dagstuhl Castle, Germany

Wermke M, Kunkel-Razun K and Scholze-Stubenrecht W (2006) Duden. Die deutsche Rechtschreibung. 24. ed., Dudenverlag, Mannheim, Germany

Wernefelt B (1984) A Resource-based View of the Firm. *Strategic Management Journal* 5 (5): 171-180

Whitaker H and Stemmer B (Eds.) (1997) Handbook of Neurolinguistics. Academic Press, San Diego

Winograd T and Flores F (1995) Understanding Computers and Cognition: A New Foundation for Design. Addison Wesley, Reading, Massachusetts, USA

Wittmann W (1959) Unternehmung und unvollkommene Information. Westdeutscher Verlag, Cologne, Germany

Wright G and Ayton P (1987) Eliciting and modelling expert knowledge. *Decision Support Systems* 3 (4): 13-26

Xu D, Wijesooriya C, Wang Y-G and Beydoun G (2011) Outbound logistics exception monitoring: A multi-perspective ontologies' approach with intelligent agents. *Expert Systems with Applications* 38 (11): 13604-13611

Ye Y, Yang D, Jiang Z and Tong L (2008) An ontology-based architecture for implementing semantic integration of supply chain management. *International Journal of Computer Integrated Manufacturing* 21 (1): 1-18

Zablith F, Antoniou G, d'Aquin, M, Flouris G, Kondylakis H and Motta E (2013) Ontology Evolution: A Process Centric Survey. *The Knowledge Engineering Review* 28 (1): 1-31

Zdravkovic M, Panetto H, Trajanovic M and Aubry A (2011) An approach for formalising the supply chain operations. *Enterprise Information Systems* 5 (4): 401-421

Zelewski S (2008) Grundlagen. In: Corsten H and Reiss M (Eds) Betriebswirtschaftlehre. 4. ed., Oldenbourg, Munich, 1-99

Zhu Z (2002) Evaluating contingency approaches to information systems design. *International Journal of Information Management* 22(5): 343-356

Appendices

Appendix 1: OWL 2 DL Entities

Table 31 presents the OWL 2 DL entities used in the outcome metamodelling language in form of the OWL 2 DL Functional Syntax (Hitzler et al. 2012; W3C OWL Working Group 2012).

Name	Functional Syntax
Individual (e.g. a, b)	Declaration (Individual (:a)) Declaration (Individual (:b))
Class (e.g. C, D, E)	Declaration (Class (:C)) Declaration (Class (:D)) Declaration (Class (:E))
ObjectProperty (e.g. R, S)	Declaration (ObjectProperty (:R)) Declaration (ObjectProperty (:S))
DataProperty (e.g. D)	Declaration (DataProperty (:D))
AnnotationProperty (e.g. A)	Declaration (AnnotationProperty (:A))

Table 31: Entities Outcome Metamodelling Language

Appendix 2: OWL 2 DL Axioms

Table 32 depicts the OWL 2 DL axioms of the outcome metamodelling language in OWL 2 DL Functional Syntax and defines their respective set semantics. Axioms of the outcome metamodelling language types denoted with * may hold more than the given parameters (Hitzler et al. 2012; W3C OWL Working Group 2012).

Functional Syntax	Set Semantics
ClassAssertion (:C :a)	$a \in C$
PropertyAssertion (:R :a :b)	$(a\,, b) \in R$
NegativePropertyAssertion (:R :a :b)	$(a\,, b) \notin R$
SameIndividual (:a :b)*	$a = b$
DifferentIndividuals (:a :b)*	$a \neq b$
SubClassOf (:C :D)	$C \sqsubseteq D$
EquivalentClasses (:C :D)*	$C \equiv D$
DisjointClasses (:C :D)*	$(C \cap D) \equiv \bot$
DisjointUnion (:C :D :E)*	$C \equiv (D \cup E)$ $(D \cap E) \equiv \bot$
SubPropertyOf (:R :S)	$R \sqsubseteq S$
EquivalentProperties (:R :S)*	$R \equiv S$
DisjointProperties (:R :S)*	$(R \cap S) \equiv \bot$
InverseProperties (:R S)	$(a, b) \in R \leftrightarrow (b, a) \in S$
PropertyDomain (:R :C)	$(a, b) \in R \rightarrow a \in C$
PropertyRange (:R :C)	$(a, b) \in R \rightarrow b \in C$
FunctionalProperty (:R)	$(a, b) \in R \wedge (a, c) \in R \rightarrow b = c$
InverseFunctionalProperty (:R)	$(a, c) \in R \wedge (b, c) \in R \rightarrow a = b$
ReflexiveProperty (:R)	$a \in \top \rightarrow (a, a) \in R$
IrreflexiveProperty (:R)	$a \in \top \rightarrow (a, a) \notin R$
SymmetricProperty (:R)	$(a, b) \in R \leftrightarrow (b, a) \in R$
AsymmetricProperty (:R)	$(a, b) \in R \rightarrow (b, a) \notin R$
TransitiveProperty (:R)	$(a, b) \in R \wedge (b, c) \in R \rightarrow (a, c) \in R$
HasKey (:C :R :S)*	$(a, c) \in R \wedge (b, c) \in R \wedge (a, d) \in S \wedge (b, d) \in S \rightarrow a = b$

Table 32: Axioms Outcome Metamodelling Language

Appendix 3: OWL 2 DL Expressions

Table 33 shows the OWL 2 DL expressions by focusing object properties. Expression types denoted with * may hold more than the given parameters. Datatype property expressions are analogous to object property expressions (Hitzler et al. 2012; W3C OWL Working Group 2012).

Functional Syntax	**Set Semantics**
IntersectionOf (:C :D)*	$C \cap D$
UnionOf (:C :D)*	$C \cup D$
ComplementOf (:C)	$\neg C$
OneOf (:a)*	$\{a\}$
SomeValuesFrom (:R :C)	$\{x \mid \exists((x,y) \in R \land y \in C)\}$
AllValuesFrom (:R :C)	$\{x \mid \forall(x,y) \in R \rightarrow y \in C\}$
HasValue (:R :a)	$\{x \mid \exists(x,a) \in R\}$
HasSelf (:R)	$\{x \mid \exists(x,x) \in R\}$
MinCardinality (n :R)	$\{x \mid \#\{y \mid (x,y) \in R\} \geq n\}$
MaxCardinality (n :R)	$\{x \mid \#\{y \mid (x,y) \in R\} \leq n\}$
ExactCardinality (n :R)	$\{x \mid \#\{y \mid (x,y) \in R\} = n\}$
MinCardinality (n :R :C)	$\{x \mid \#\{y \mid (x,y) \in R \land y \in C\} \geq n\}$
MaxCardinality (n :R :C)	$\{x \mid \#\{y \mid (x,y) \in R \land y \in C\} \leq n\}$
ExactCardinality (n :R :C)	$\{x \mid \#\{y \mid (x,y) \in R \land y \in C\} = n\}$
PropertyChain (:R :S)*	$\{(a,b) \mid \exists(a,x) \in R \land (x,b) \in S\}$

Table 33: Expressions Outcome Metamodelling Language