FAMOUS-2 OVERALL ARCHITECTURE

PNA-F2-001

Issue 0, Rev. 4

12/05/2014

Document Change Record

Date	Version/Rev	Changes	Reason for Change
15/01/2014	0/1	All	First Issue
20/03/2014	0/2	Major update	Meeting of 16/01/2014 at ESA
21/03/2014	0/3	Introduction	Moved overall triangle model to the introduction as it forms the basis for the architecture of FAMOUS-2, while the other chapters are an instantiation of this triangle.
12/05/2014	0/4	Model update	Updated model: split fact types for reverse engineering into two: LM is based on CM; CM is result of reverse engineering of LM *Note: not added the triangle discussed in April with ESA (the orthogonal triangle)*

Table of Contents

Table of figures

1 Introduction

Developing and operating a space system implies complex activities involving many parties who may be widely-distributed, both geographically as well as in time. Successful development and operation of space systems therefore requires successful information sharing during the whole lifecycle of the space system. This is an integral part of the system engineering process and can only be achieved by realising *interoperability* between all involved parties.

Interoperability is often described in terms of data exchange according to specific data formats and communication protocols, i.e. from a *syntactical* viewpoint. While agreements on *how* to exchange the data are required for interoperability, syntactic agreements alone are not sufficient to guarantee a complete understanding of the information that is exchanged. For this understanding, *semantic* interoperability, addressing the "what" in the exchange is required.

1.1 Syntactic interoperability = exchange of data + schema

As mentioned, interoperability requires both syntactic agreements on how to exchange information (syntactic interoperability) as well as semantic agreements on how to interpret the information that is exchanged (semantic interoperability).

A standard manner for achieving syntactic interoperability is to specify a set of syntax rules that are to be obeyed by the data that is exchanged. That is, syntactic interoperability implies that the *structure* in which the data is exchanged is defined. This means that the domain-specific schema, for which the data is an instance shall be known by all parties involved. However, the generic schema that specifies the rules to which the domain-specific schema has to conform shall also be known such that there is no mismatch of interpretation of the domain-specific schema. In essence, for syntactic interoperability:

1) the data that is to be exchanged needs to be known,
2) the domain-specific schema to which the data complies needs to be known, and
3) the generic schema to which the domain-specific schema complies needs to be known (also known as the meta-schema).

In the context of the FAMOUS-2 research project ([RD2]) the Knowledge Triangle as described in [RD5] is used to differentiate between the three levels described above.

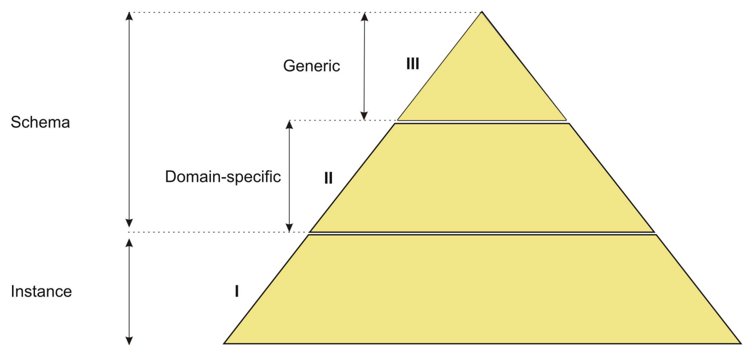

Figure 1: Exchange of data requires the data itself (level I), the domain-specific schema (level II) and the generic schema (level III).

The first level of the Knowledge Triangle is the instance level. This level is the most concrete level in any structured knowledge description or business communication. In case of a fact-based modelling method for example, the instance level is the level of the facts that do not contain any regulation. In case of an Oracle database, the instance level consists of the values for the columns of the different tables of the database. In general terms, the instance level of the Knowledge Triangle consists of the population that is regulated by the domain-specific schema specified at level II of the Knowledge Triangle.

The second level of the Knowledge Triangle is the domain-specific level, which contains the regulations for any specific domain of interest expressed in a specific language. That is, a domain-specific schema, expressed at level II of the Knowledge Triangle, specifies all regulations that must be satisfied by the elements of the instance level.

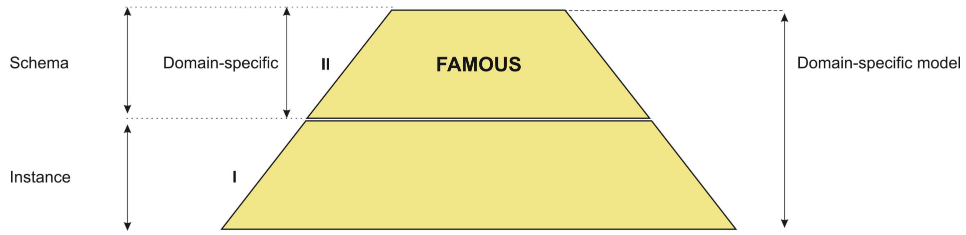

Figure 2: The domain-specific schema level of the Knowledge Triangle.

Figure 2 illustrates a specific Knowledge Triangle instance in which level II contains the regulation for a specific domain expressed in the FAMOUS language. An example of a possible regulation for a specific domain expressed in the FAMOUS language is:

`“Each launch attempt shall have exactly one flight number”.`

Since the domain-specific schema level specifies all regulations that need to be satisfied by (the combination of) the instances at the instance level for any domain-of interest, it contains the domain-specific schema for each domain of interest, expressed in a specific language. This is expressed in the model fragment of Figure 3.

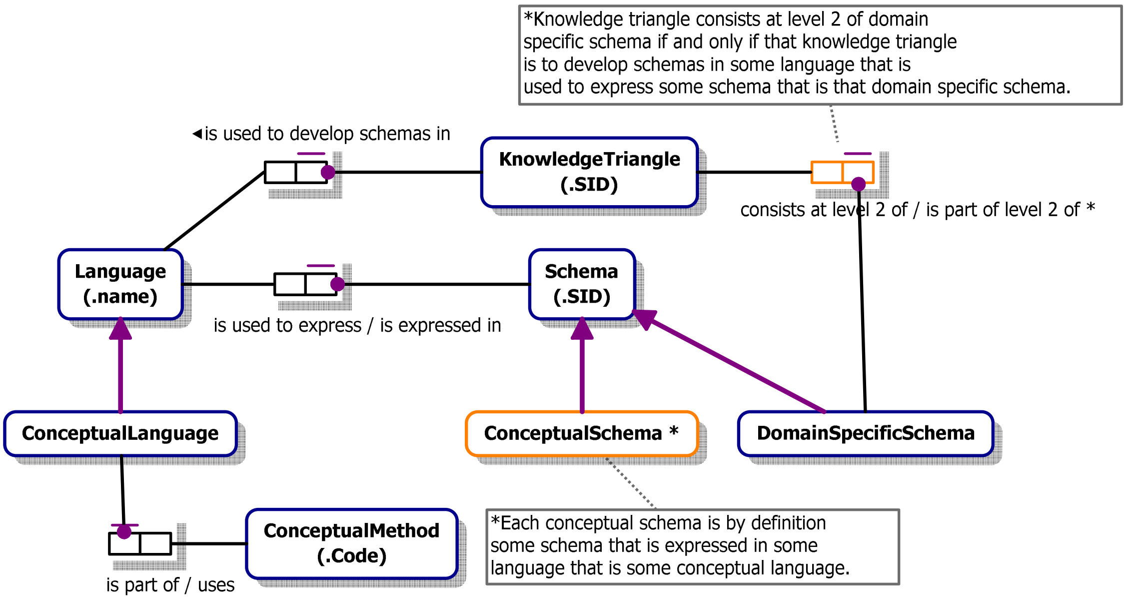

Figure 3: FAMOUS-2 conceptual schema of level II of the Knowledge Triangle.

In Figure 2, it is also indicated that the combination of a domain-specific schema and the associated population is called the *domain-specific model*. In Figure 4, the conceptual schema of "model" is given.

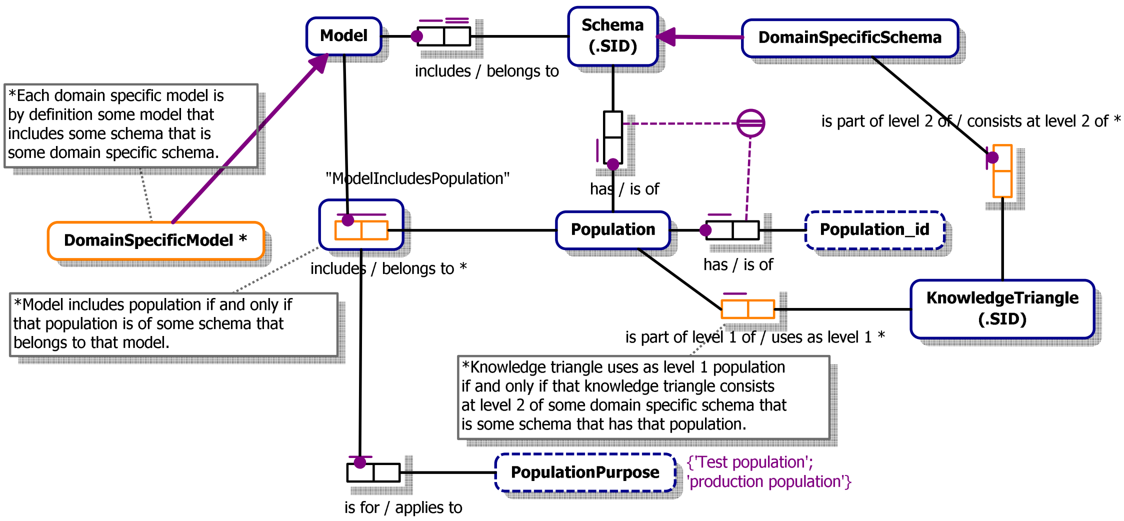

Figure 4: Conceptual schema of model.

As specified in the schema fragment above, each model includes exactly one schema and one or more populations for the schema, whereby each population serves a specific purpose.

The third level of the Knowledge Triangle is the *generic schema level*, as shown in Figure 5.

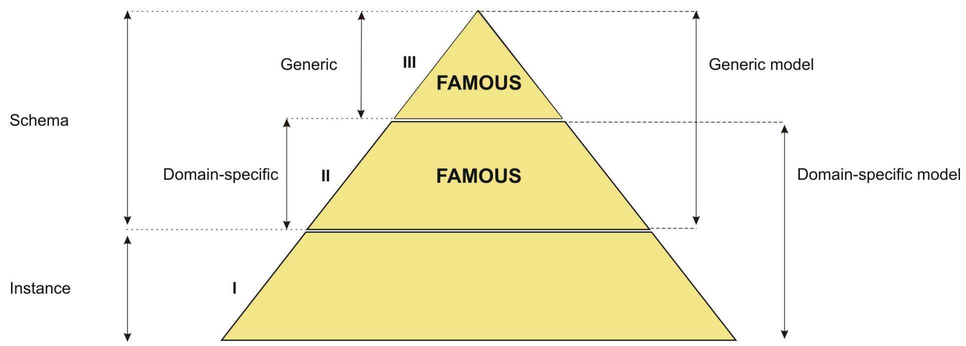

Figure 5: The generic schema level of the Knowledge Triangle.

The generic schema level, also called level III, contains the regulations for all schemas expressed in a *specific language* at domain-specific schema level. For the specification of these regulations the generic schema level itself consists of exactly one (generic) schema that is also expressed in a specific (possibly other) language[1]. Figure 5 indicates that in this case the schema at generic schema level is expressed in the FAMOUS language and contains the regulations for schemas at domain-specific schema level (also) expressed in the FAMOUS language. I.e. the Knowledge Triangle of Figure 5 is the Knowledge Triangle for the FAMOUS language (level II) expressed in the FAMOUS language (level III). Two examples of generic regulations expressed at level III in the FAMOUS language for domain-specific schemas at level II (also) expressed in the FAMOUS language are:

`“Each fact type shall contain at least one role.”`

`“Each role shall be played by exactly one object type.”`

In Figure 6, the conceptual schema with respect to level III of the Knowledge Triangle is illustrated.

[1] Within the context of the FAMOUS-2 research project, each schema at generic level shall be expressed in the FAMOUS language.

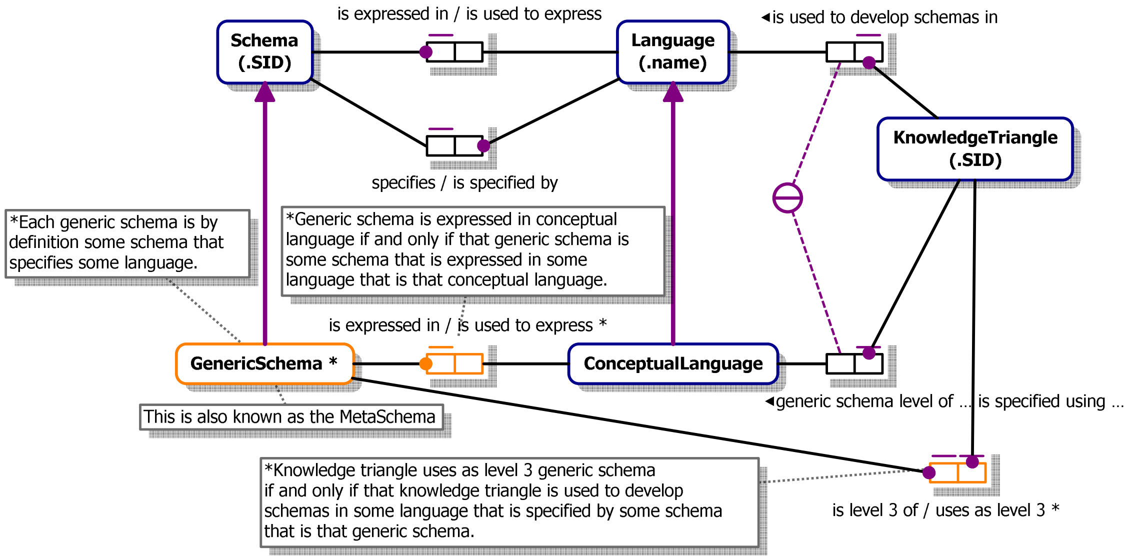

Figure 6: Conceptual schema of level III of the Knowledge Triangle as used in FAMOUS-2.

The model fragment above specifies that each generic schema is a schema that is expressed in a language and at the same time specifies a language.

The Knowledge Triangle introduced in Figure 5 also indicates that a *generic model* consists of a generic schema together with a domain-specific schema as population. This is depicted in the model fragment in Figure 7.

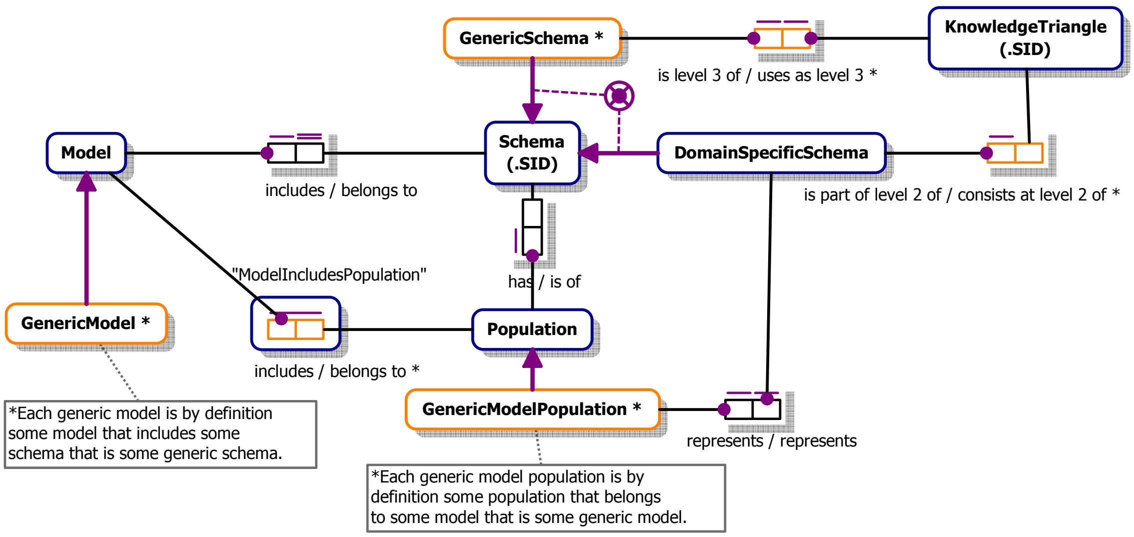

Figure 7: FAMOUS-2 conceptual schema with respect to level II as population of level III of the Knowledge Triangle.

1.2 Semantic interoperability

As specified before, for syntactic interoperability:

1) the data that is to be exchanged needs to be known,
2) the domain-specific schema to which the data complies needs to be known, and
3) the generic schema to which the domain-specific schema complies needs to be known (also known as the meta-schema).

Interoperability requires not only the syntactical agreements, what is also required is to understand *what* is exchanged. This aspect is called "semantic interoperability".

Semantic interoperability focuses on the "what", the requirements that have to be fulfilled when exchanging information, both by the receiver as well as the sender of the data. For this, it is required to model at *conceptual level*. Modelling at conceptual level should be independent of any design or implementation concerns. The aim of modelling at conceptual level is to express the *meaning* of the terms and concepts used by the different parties involved and to find the correct relationships between the different concepts. A conceptual model has as main aim to remove ambiguity as well as any risk of misunderstanding. Therefore, within the context of the FAMOUS-2 research project, each schema at the generic level shall be expressed in the FAMOUS language, which is a conceptual language.

In the FAMOUS-2 research project, we distinguish between conceptual languages, logical languages and physical languages. As specified above, a conceptual language is a modelling language that is used to convey meaning, the semantics of the domain of interest.

A logical language is a language which is used for a specific technology, and thus is used to express schemas that anticipate the implementation in a specific technology, like e.g. the relational technology. A physical model is the realisation (implementation) of a specific technology. A physical language therefore is used to express schemas that take into account the facilities and restrictions associated with the implementation of the technology. This principle is depicted in the model fragment of Figure 8.

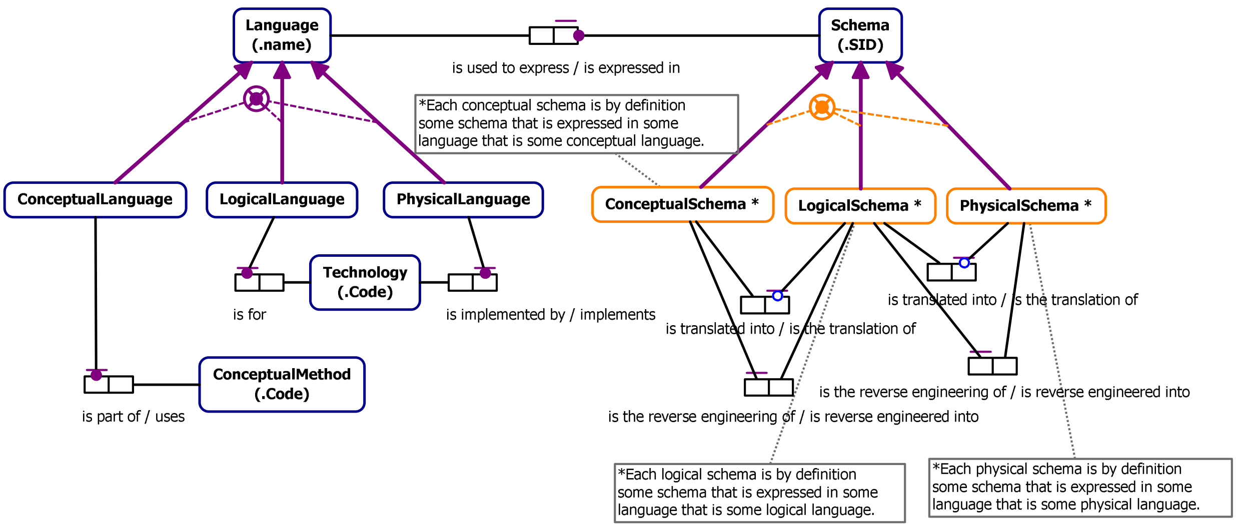

Figure 8: The different types of languages and associated schemas.

In the following, we elaborate on the different types of models that exist and how they are all related through a corresponding conceptual model, where a model is defined as a (domain-specific or generic) schema with associated population.

After introduction of the different types of models, it is described how semantic interoperability can be achieved using the concept of a global conceptual model.

1.3 Applicable and reference documents

Refer.	Title	Identifier/ISBN	Date	Issue
[RD1]	Space engineering – Space system data repository	ECSS-E-TM-10-23A	25/11/2011	1.0
[RD2]	ESA/ESTEC/Contract No. 4000107725/13/NL/GLC/al	4000107725/13/NL/GLC/al	04/03/2013	
[RD3]	Fact-Based Modelling Metamodel	FBM1002WD08	29/20/2013	WD08
[RD4]	ANSI/X3/SPARC Study Group on Data Base Management Systems: (1975), Interim Report. FDT, ACM SIGMOD bulletin. Volume 7, No. 2	ANSI/X3/SPARC	1975	
[RD5]	Kennis Gebaseerd Werken	ISBN 978-90-5540-013-3	2009	
[RD6]	Information technology – Database languages – SQL – Part 1 & 2	ISO/IEC 9075-1 & 2:2011	2011	
[RD7]	[Schmaal, 2010]: Data Exchange in ESA Space System Projects – Producing hierarchical data exchange formats from ORM conceptual schemas.		2010	
[RD8]	OMG Unified Modeling Language™ (OMG UML)	Ptc/2012-10-24	2010	2.5 Beta 1
[RD9]	http://www.oracle.com/pls/db112/homepage		2013	
[RD10]	www.w3.org/standards/xml/schema		2013	
[RD11]	www.omg.org/spec/XMI/2.4.1		2013	

[RD12]	AuGeMMI – statement of work	ESA/ESTEC/TEC-SWM/11-525/sva	05/07/2012	1.0

1.4 Acronyms and Abbreviations

A	
AuGeMMI	Automatic Generated MMI
E	
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
F	
FAMOUS	FAct based MOdelling Unifying System
FBM	Fact-based Modelling
M	
MMI	Man Machine Interface
U	
UML	Unified Modeling Language
X	
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XSD	XML Schema Definition

1.5 Terms and definitions

base-lining

identifying and selecting the subset of requirements within a global conceptual schema that constitute a local view

conceptual schema

schema that captures those aspects about the universe of discourse that are conceptually relevant

NOTE1 A{conceptual schema} specifies the semantics (the what) including the meaning of every term that could be misunderstood by the intended audience and ignoring aspects of data representation, physical data organisation and access (the how).

NOTE2 A {conceptual schema} declares the fact types, constraints, derivation rules, events and concept definitions relevant to the universe of discourse.

destination

schema with possible population that is output when transforming

domain of interest

synonym of universe of discourse

domain-specific schema level

knowledge level that specifies the regulations for any universe of discourse

NOTE1 The assertions at instance level for any universe of discourse should adhere the regulations for that universe of discourse specified at {domain-specific schema level}.

NOTE2 An example of a possible regulation at {domain-specific schema level} is: 'Each launch attempt took place on exactly one date'.

extracting

creating a local conceptual schema separate from a global conceptual schema that corresponds with the requirements determined by the corresponding local view within the context of that global conceptual schema

extraction rules

regulations that specify how requirements determined by a local view within the context of a global conceptual schema are converted into a corresponding local conceptual schema separate from that global conceptual schema

FAct-based MOdelling Unifying System

fact-based method for developing conceptual schemas that therefore contains a language for specifying conceptual schemas

NOTE {FAMOUS} also supports the development of logical schemas and physical schemas based on their corresponding conceptual schema.

generic schema level

knowledge level that specifies the regulations for a language and therefore the regulations to which any schema in that language for any universe of discourse (hence generic) shall adhere

NOTE Within the context of FAMOUS-2 all regulations at {generic schema level} shall be specified by a conceptual schema using the FAMOUS language.

global conceptual schema

conceptual schema for the complete universe of discourse, i.e. overall system, the complete lifecycle and all involved disciplines

hierarchical

language for specifying logical schemas

hierarchical schema

logical schema expressed in the Hierarchical language

instance level

knowledge level that only consists of facts, i.e. assertions that do not contain any regulation or perform any grammatical function

NOTE Two examples of facts at {instance level} are:

The launch attempt that was performed by the launcher Ariane 5 and has flight number VA212 took place on 7 February 2013.

The launch attempt that was performed by the launcher Ariane 5 and has flight number VA211 took place on 19 December 2012.

knowledge level

classification of knowledge based on corresponding properties of that knowledge

knowledge triangle

framework for knowledge and (business) communication about this knowledge

NOTE1 The {knowledge triangle} consists of a classification of knowledge allocated over the instance level, domain-specific schema level and generic schema level and therefore provides guidance for analysing, processing and working with knowledge.

NOTE2 The instance level forms the base of the {knowledge triangle} with the domain-specific schema level and generic schema level successively on top of that.

language

means to communicate knowledge

NOTE A schema (and a possible associated population) is specified in a {language}.

level I

synonym of instance level

level II

synonym of domain-specific schema level

level III

synonym of generic schema level

local conceptual schema

conceptual schema consisting of a part of the requirements of a global conceptual schema

NOTE1 A {local conceptual schema} may be updated but these updates are not repatriated to (the local view on) the global conceptual schema.

NOTE2 A {local conceptual schema} is for:

1) a part of the universe of discourse of the global conceptual schema,
2) a part of the lifecycle of the universe of discourse,
3) one or more of the involved disciplines;
4) or a combination of the above.

local view

view on (a part of) a global conceptual schema that contains the requirements of that global conceptual schema that correspond with the requirements of a local conceptual schema

NOTE A {local view} is extracted from a global conceptual schema in order to create a local conceptual schema.

logical schema

schema that anticipates implementation in a specific (software) technology

NOTE1 A {logical schema} is adjusted to achieve certain practical efficiencies in comparison to the conceptual data model.

NOTE2 A {logical schema} is derived from, i.e. based upon a conceptual schema.

method

body of techniques for investigating and acquiring new knowledge or correcting and integrating previous knowledge

methodology

the systematic and theoretical analysis of the methods applied to a field of study, or the theoretical analysis of the body of methods and principles associated with a branch of knowledge

NOTE Conceptual modelling is a {methodology} where e.g. FAMOUS, ORM and CogNIAM are possible methods for conceptual modelling.

model

schema with an associated population

Oracle

language for specifying physical schemas based upon a logical schema in Relational language

Oracle schema

physical schema expressed in the Oracle language

physical schema

schema which is a representation of a universe of discourse taking into account the facilities and constraints that are part of a given storage system or data management technology

NOTE A {physical schema} is derived from, i.e. based upon a logical schema.

relational

language for specifying logical schemas

relational schema

logical schema expressed in the Relational language

requirement

expression of a need of the business independent of an IT implementation

schema

structure that determines the regulations for a universe of discourse

NOTE The regulations for a universe of discourse consist of the possible and permitted states and the possible and permitted transitions within that universe of discourse.

source

schema with possible population that is input when transforming

structural part

specification of the terms and definitions with associated requirements that are part, i.e. of interest to a method

NOTE1 The FAMOUS method e.g. consist of fact types, integrity rules and derivation rules.

NOTE2 The {structural part} specifies *what* a method is about.

structuring part

specification of the protocol that is to be followed when a method is applied

NOTE The {structuring part} specifies *how* a method is to be applied.

tailoring

adjusting the selected requirements that constitute a local view on the global conceptual schema by strengthening them according to local product's needs

transformation rules

regulations that specify how a schema at [domain specific schema level] including a possible population at instance level is converted into another language

transforming

converting a schema at domain-specific schema level including a possible population at instance level from one language into another language

NOTE Within the context of FAMOUS-2 a conceptual schema is (generally) transformed into a logical schema and subsequently into a physical schema. The other way around from a physical schema via a logical schema into a conceptual schema is also supported.

UML class diagram

logical schema expressed in the UML language

Unified Modelling Language

language for specifying logical schemas

universe of discourse

the aspects of the world that the relevant community for the {universe of discourse} wishes to talk about

XMI schema

physical schema expressed in the XMI language

XML Metadata Interchange

language for specifying physical schemas based upon a logical schema in UML language

XML Schema definition language

language for specifying physical schemas based upon a logical schema in Relational or Hierarchical language

XSD schema

physical schema expressed in the XSD language

2 Conceptual model

In the context of the FAMOUS-2 research project, the FAMOUS modelling method is used as the means to develop domain-specific conceptual models. The result of the modelling process, i.e., applying the modelling protocol, is expressed in the FAMOUS language. The FAMOUS language is a fact-based modelling language, meaning that the conceptual model captures the semantics of the relevant domain by means of fact types together with the associated concept definitions (the terms and definitions), the communication patterns as well as the rules applying to these fact types.

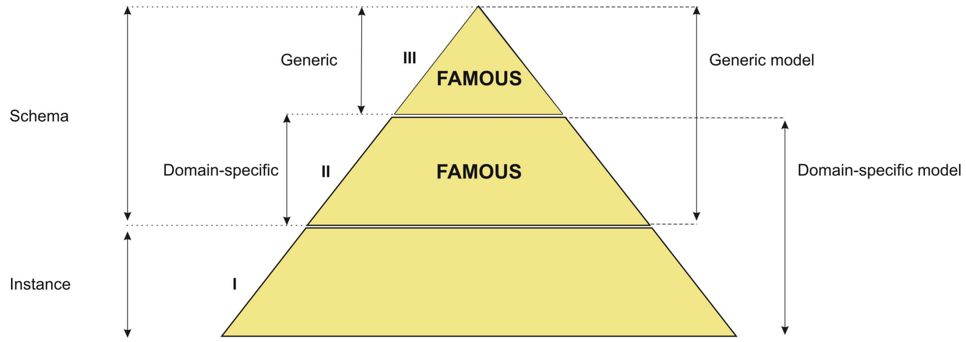

Figure 9: Conceptual models are expressed in the FAMOUS language.

Using FAMOUS to define the generic schema has as advantage that the generic schema is described implementation-independent; i.e. the requirements that need to be fulfilled by *any* domain-specific schema that is expressed in the FAMOUS language are described implementation-independent.

2.1.1 The FAMOUS method

The FAMOUS method is a method that consists of both a structural as well as a structuring part. The structural part specifies all the concepts that are of interest (the language) while the structuring part specifies how, when and by whom the different concepts are used and/or created.

As specified above, the FAMOUS language is a fact-based modelling language, meaning that the conceptual schema captures the semantics of the relevant domain by means of fact types together with the associated concept definitions (the terms and definitions), the communication patterns as well as the rules applying to these fact types. The core of the FAMOUS language is provided by the FBM WG in WD08([RD3]). In WP2100, for the FAMOUS-2 research project, the method is extended with means to:

1) formally specify derivation rules (WP2200),
2) identify existential dependency relationships between different object types (WP2300),
3) formally specify dynamic rules (WP2400),
4) provide extensive support for data types, units, quantities and dimensions in accordance to QUDV (WP2500),
5) identify possible compound conceptual definitions (assemblies) and their functional parts (WP2600) in relation to AuGeMMI ([RD12]), and
6) provide support for interoperability through a global conceptual model (WP2700).

The complete method, integrating the different work packages is the subject of WP2800. In this work package, both the structural as well as the structuring part is explained and the integrated generic schema of the FAMOUS language is provided.

2.2 Conceptual schema as the basis for logical schemas

A conceptual model specifies the semantics of the domain of interest. It is expressed in a conceptual language which is free of any implementation detail. A conceptual model can be used as the basis for logical models, which *anticipate* the implementation of the model in a specific technology. That is, a logical model is expressed in a logical language which takes into account the implementation in a specific technology by introducing structuring information to achieve practical efficiencies in implementation. The types of logical models that shall be supported by FAMOUS-2 are:

1) logical models expressed in the relational language (based on the SQL standard [RD6]),
2) logical models expressed in the hierarchical language (based on [RD7]), and
3) logical models expressed in the UML (based on the UML standard [RD8]).

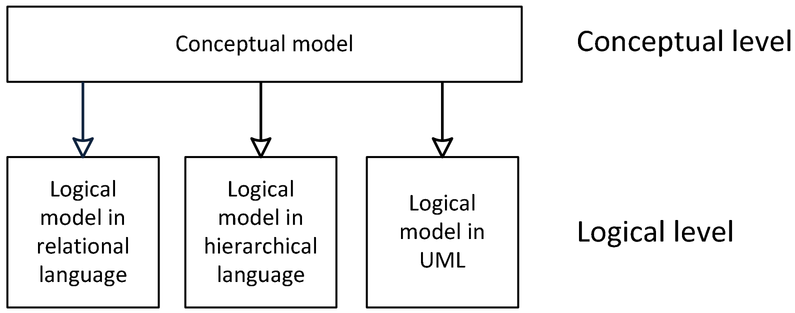

Figure 10: A conceptual model forms the basis for logical models.

For each of the types of logical models supported by FAMOUS-2, the generic schema (the meta-schema) shall be expressed in the FAMOUS language. This principle is depicted in Figure 11 to Figure 13.

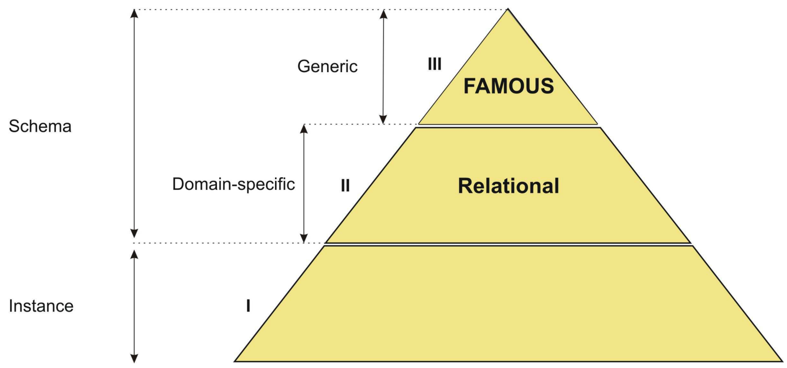

Figure 11: The generic schema of the relational language expressed in the FAMOUS language (WP 3100).

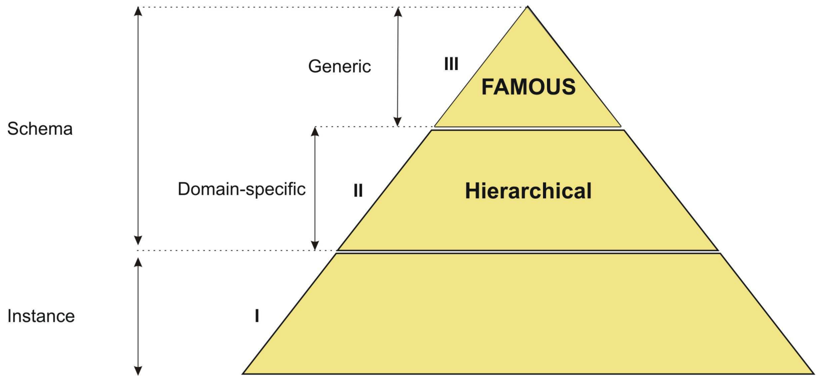

Figure 12: The generic schema of the hierarchical language expressed in the FAMOUS language (WP 3200).

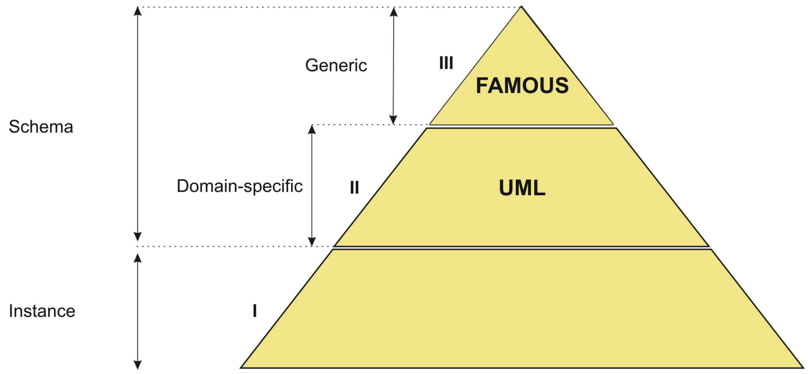

Figure 13: The generic schema of the UML expressed in the FAMOUS language (WP 3300).

2.3 Logical schemas as the basis for physical schemas

A physical schema is a representation of a data design taking into account the facilities and restrictions that are part of a storage system using a given technology. That is, while a logical language *anticipates* an implementation strategy (e.g., relational database, hierarchical database, object-oriented database), the physical language represents the actual implementation in a specific tool (e.g., Oracle). A physical model is, in the lifecycle of a project, typically derived from a logical model. The types of physical models that shall be supported by FAMOUS-2 are:

1) relational models expressed in the Oracle language (based on [RD9]),
2) hierarchical models expressed in the XSD language (based on [RD10]), and
3) UML models expressed in the XMI language (based on [RD11]).

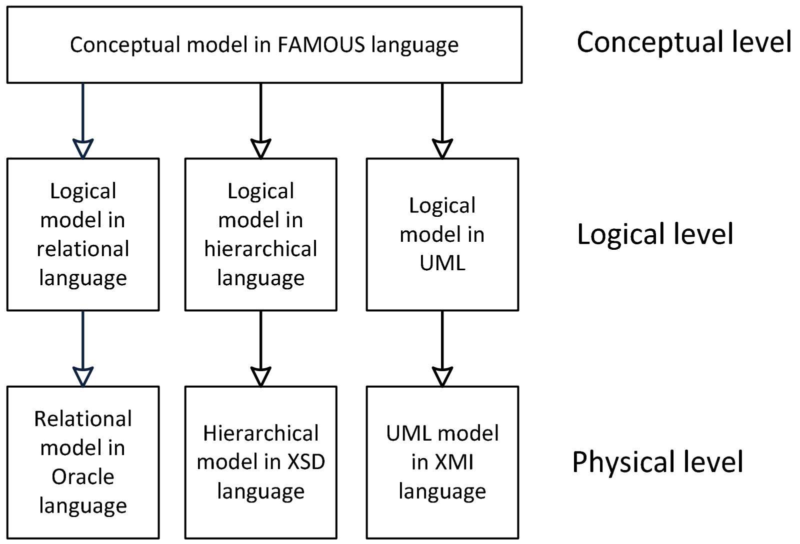

Figure 14: A physical model is typically based on a logical model.

As with the logical models, for each of the types of physical models supported by FAMOUS-2, the generic schema (the meta-schema) shall be expressed in the FAMOUS language. This principle is depicted in Figure 15 to Figure 17.

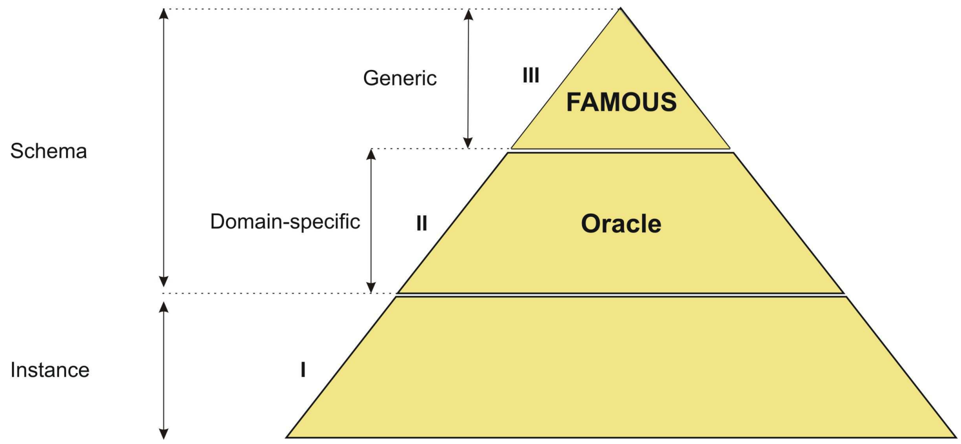

Figure 15: The generic schema of the Oracle language expressed in the FAMOUS language (WP 3100).

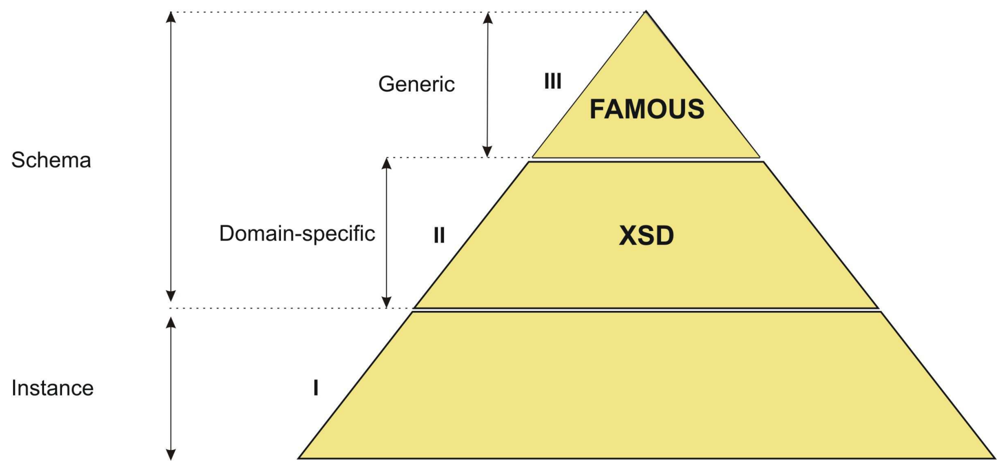

Figure 16: The generic schema of the XSD language expressed in the FAMOUS language (WP 3200).

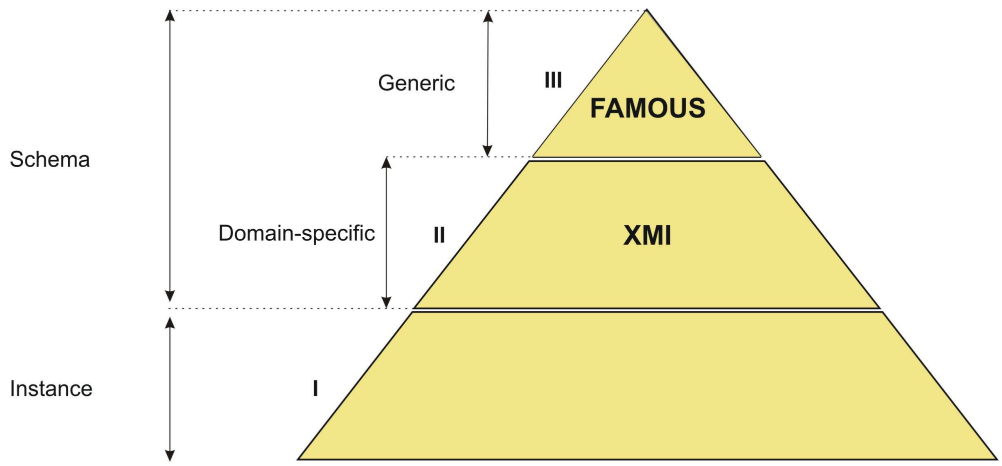

Figure 17: The generic schema of the XMI language expressed in the FAMOUS language (WP 3300).

3 The generic schemas expressed in FAMOUS – simplifying the transformations

3.1 From conceptual to logical

As discussed in the previous section, a logical model should be based on the structures identified in the preceding conceptual model since the conceptual model describes the semantics of the information that needs to be reflected in the logical model, while the logical model adjusts the representation by adding technology-specific information. That is, the logical models should be derivable from the conceptual model through transformation of the constructs of the conceptual language to the constructs of the logical languages.

Transformation of a conceptual model to a logical model is done at the generic level. That is, transforming a domain-specific model expressed in the FAMOUS language to a domain-specific model in a logical language e.g. the relational language, implies the transformation of the constructs of the conceptual language to the corresponding constructs of the relational language. In other words, the transformation rules specify how the constructs of the conceptual language are mapped to the corresponding constructs in the logical language. These mappings form the transformation rules which are used as the prescription for transforming a domain-specific FAMOUS model (schema and population) into a domain-specific logical model. In FAMOUS-2 the following transformations for conceptual models to logical models are supported:

1) from a conceptual model expressed in FAMOUS to the corresponding logical model expressed in the relational language (Figure 18),
2) from a conceptual model expressed in FAMOUS to the corresponding logical model expressed in the hierarchical language (Figure 19), and
3) from a conceptual model expressed in FAMOUS to the corresponding logical model expressed in the UML (Figure 20).

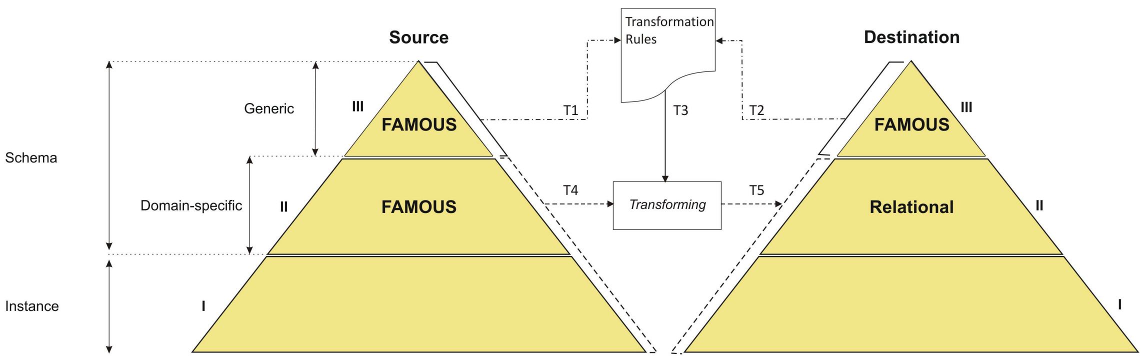

Figure 18: Transformation from a FAMOUS conceptual model to a logical model expressed in the relational language (WP 4100).

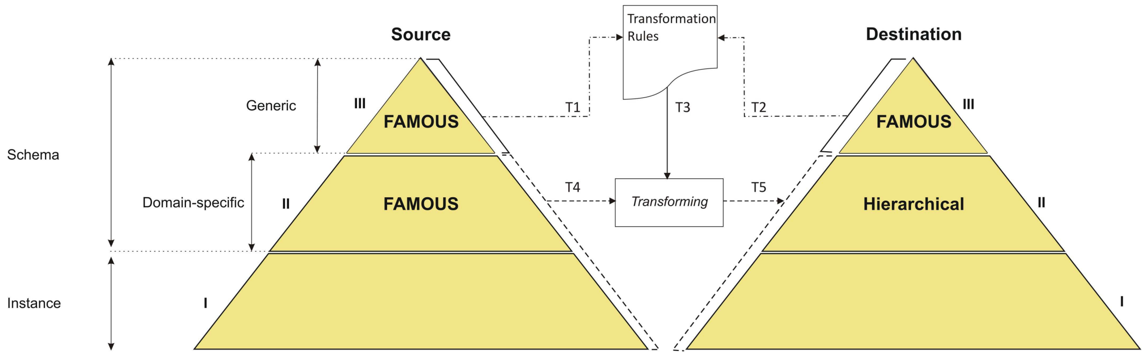

Figure 19: Transformation from a FAMOUS conceptual model to a logical model expressed in the hierarchical language (WP 4200).

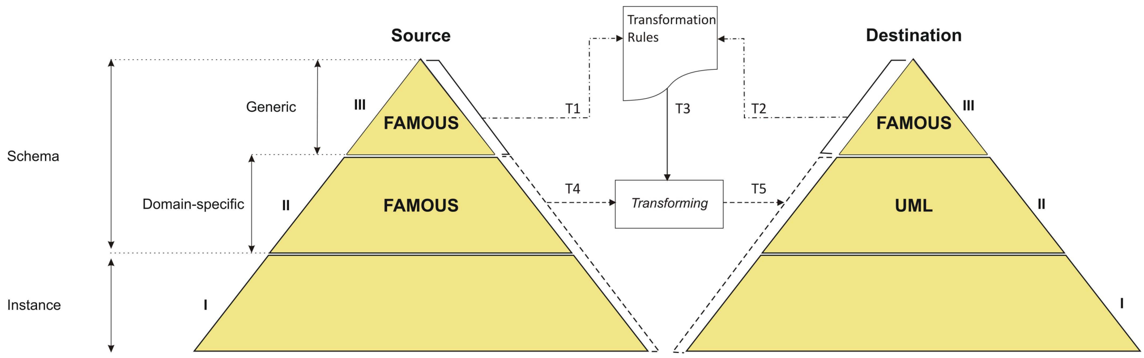

Figure 20: Transformation from a FAMOUS conceptual model to a logical model expressed in the UML (WP 4300).

3.2 From logical to physical

A physical model is typically derived from a logical model, whereby the physical model is an implementation-specific representation of the logical model. That is, a physical model does not change the structure of the representation in the logical language, but provides a transformation in the form of a translation to the associated implementation constructs.

In FAMOUS-2, the following transformations from logical models to physical models are supported:

1) from a logical model expressed in the relational language to the corresponding relational model expressed in the Oracle language (Figure 21),
2) from a logical model expressed in the hierarchical language to the corresponding hierarchical model expressed in the XSD language (Figure 22), and
3) from a logical model expressed in the UML to the corresponding UML model expressed in the XMI language (Figure 23).

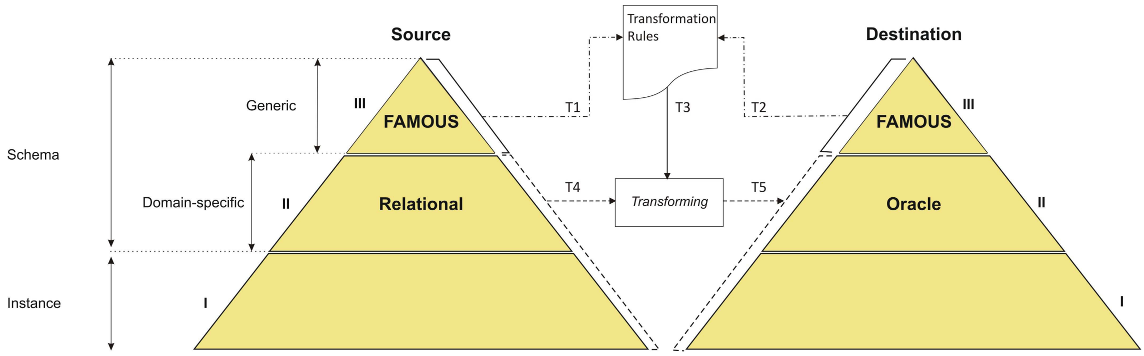

Figure 21: Transformation from a logical model expressed in the relational language to a relational model expressed in the Oracle language (WP 4100).

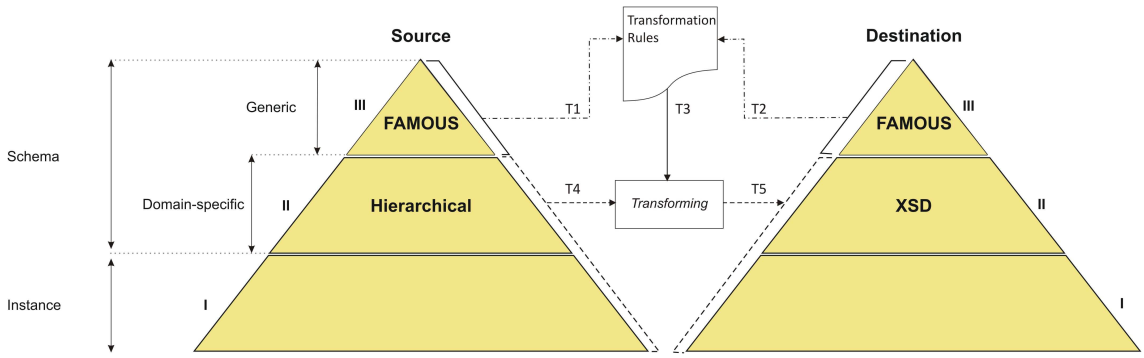

Figure 22: Transformation from a logical model expressed in the hierarchical language to a hierarchical model expressed in the XSD language (WP 4200).

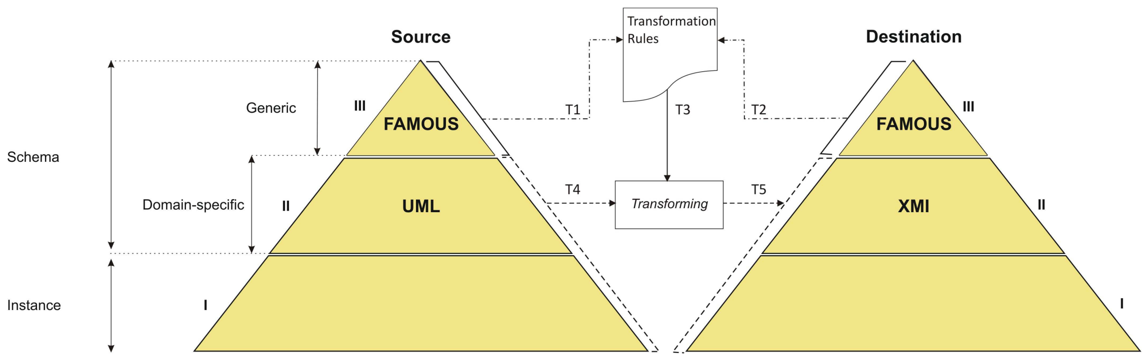

Figure 23: Transformation from a logical model expressed in the UML to a UML model expressed in the XMI language (WP4300).

3.3 Reverse transformations: from physical to logical to conceptual

Reverse engineering, the process of analysing a system to create representations of the systems that are understandable for communication, can be seen as "going backwards in the development cycle". That is,

while the development cycle dictates that a conceptual model should be formed as the basis for a logical model, which can then in the development lifecycle be transformed into the physical model, reverse engineering involves the backwards transformation, namely going from a physical model to an associated logical model and from thereon to the conceptual model.

Reverse engineering is not a trivial task, since it involves assessing and interpreting implementation-specific concepts, and adding stepwise semantic information for arriving at the conceptual level. Reverse engineering is moreover difficult since the same construct can often be implemented in many manners, depending on e.g. programmer's preferences, guidelines and best practices. Moreover, the further going back in the development cycle, the more semantics needs to be added.

The reverse engineering transformations that are supported by FAMOUS-2 are:

1) from a relational model expressed in the Oracle language to a logical model expressed in the relational language (Figure 24),
2) from a hierarchical model expressed in the XSD language to a logical model expressed in the hierarchical language (Figure 25),
3) from a UML model expressed in the XMI language to a logical model expressed in the UML (Figure 26).

These transformations, from physical level to logical level are rather straightforward with respect to the semantics to be added since it involves translations more than transformations.

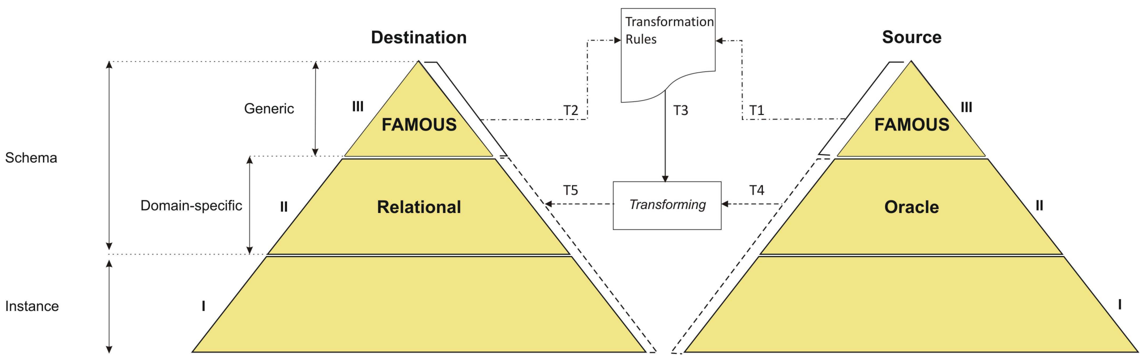

Figure 24: Transformation from a relational model expressed in the Oracle language to a logical model expressed in the relational language (WP 4100).

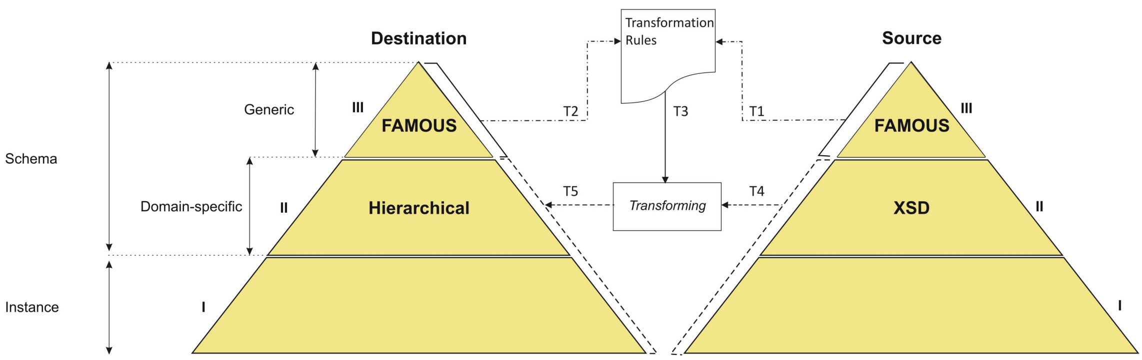

Figure 25: Transformation from a hierarchical model expressed in the XSD language to a logical model expressed in the hierarchical language (WP 4200).

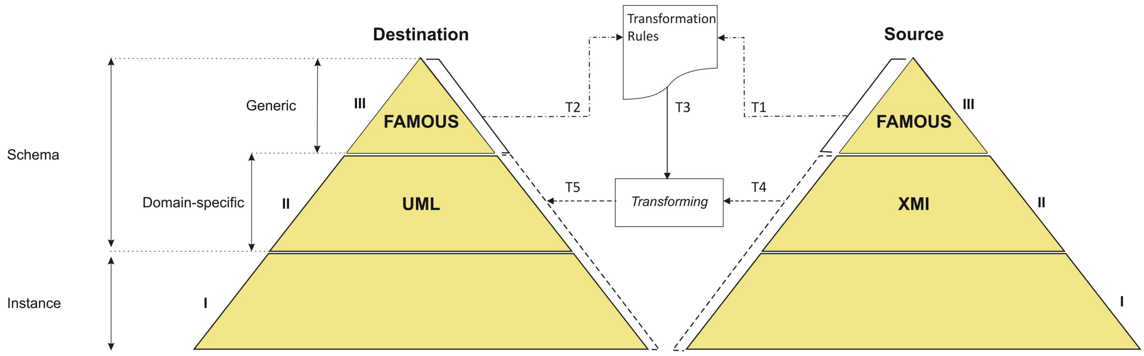

Figure 26: Transformation from a UML model expressed in the XMI language to a logical model expressed in the UML (WP 4300).

The transformations from logical to conceptual imply transforming logical structures into conceptual structures, implying the addition of semantics. That is, semantics needs to be added in order for the resulting conceptual model to make sense.

The reverse engineering transformations that are supported by FAMOUS-2 from logical to conceptual are:

1) from a logical model expressed in the relational language to a conceptual model expressed in the FAMOUS language (Figure 27),
2) from a logical model expressed in the hierarchical language to a conceptual model expressed in the FAMOUS language (Figure 28), and
3) from a logical model expressed in the UML to a conceptual model expressed in the FAMOUS language (Figure 29).

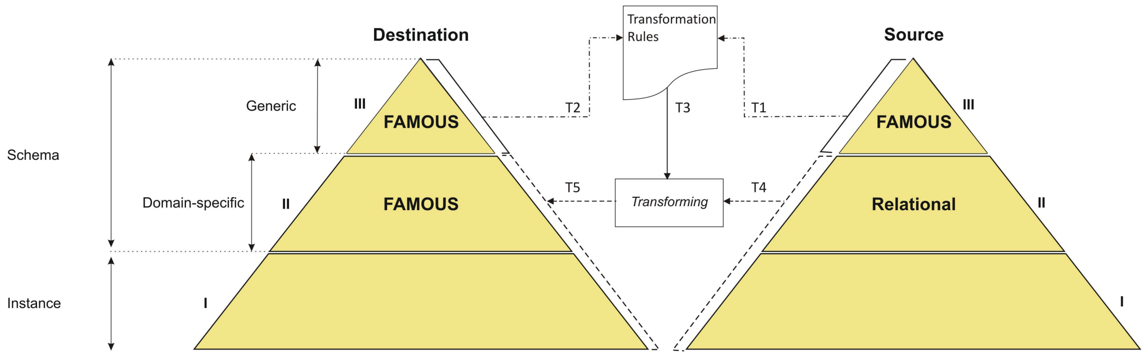

Figure 27: Transformation from a logical model expressed in the relational language to a conceptual model expressed in the FAMOUS language (WP 4100).

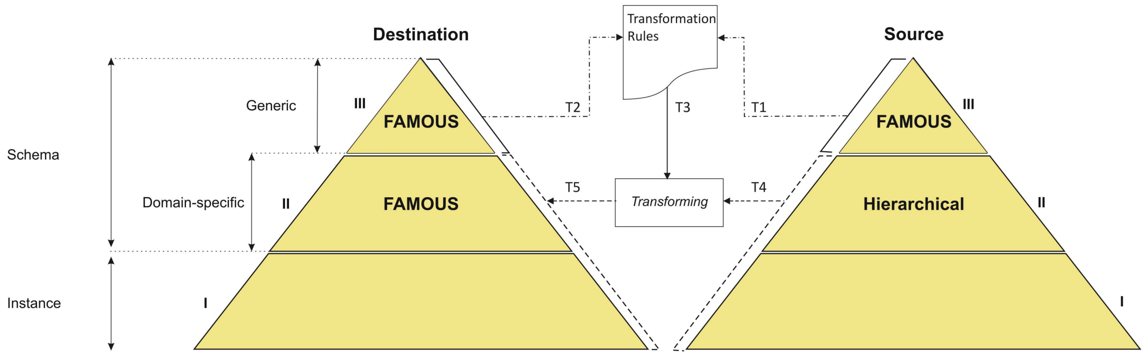

Figure 28: Transformation from a logical model expressed in the hierarchical language to a conceptual model expressed in the FAMOUS language (WP 4200).

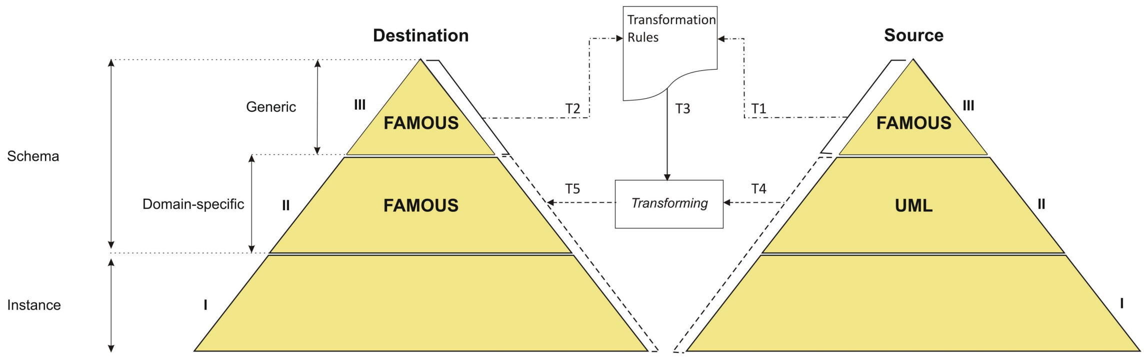

Figure 29: Transformation from a logical model expressed in the UML to a conceptual model expressed in the FAMOUS language (WP 4300).

4 Global versus local modelling – a means to realise semantic interoperability

In [RD1], the concept of modelling at global level has been introduced, in line with the single "ECSS system" development approach, i.e.:

1) standardizing at global level the "what" and
2) offering means to tailor that "what' recursively from projects to systems, elements, subsystems, whereby tailoring means adding specific needs and selecting the subset of the global conceptual model of interest.

Using the FAMOUS conceptual modelling approach to develop a global conceptual schema (at domain-specific schema level), each party involved in the development of the space system (and thus involved in the exchange) can extract its own "local" conceptual schema from this global conceptual schema by:

1) base-lining through identifying that subset of the global conceptual schema that applies in the local context,
2) tailoring the selected subset by potentially strengthening the requirements, by e.g. adding additional derivation rule for mapping from the global to the local view, and
3) extracting the tailored local view from the global conceptual model such that it can be used as the starting point for further development in the local context.

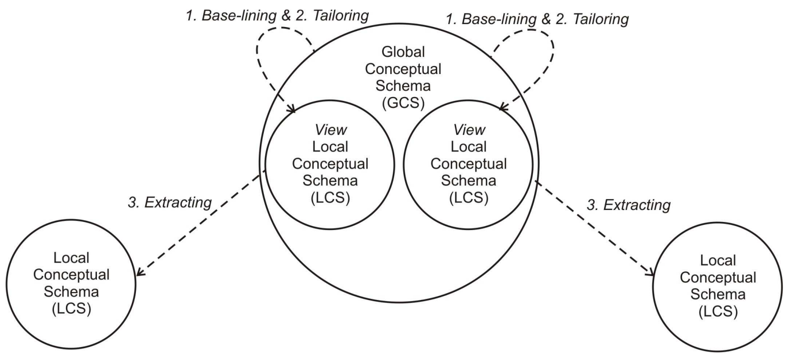

Figure 30: From global conceptual schema to local conceptual schemas.

Figure 31 indicates the principle explained above within the context of the Knowledge Triangle: within a global conceptual schema, local *views* (indicated with a dotted line outline) can be established (LCS A and B) which correspond to the requirements for the local conceptual schemas that will be extracted from the global conceptual schema based on these views.

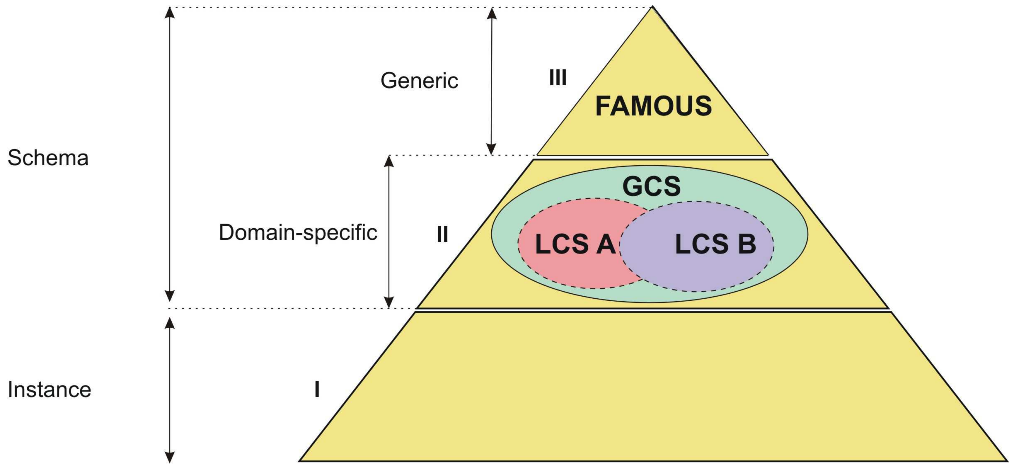

Figure 31: Local conceptual schemas as views on the global conceptual schema.

As indicated in Figure 32, a local view is extracted from the global conceptual schema to create the corresponding local conceptual schema. This local conceptual schema, expressed in the FAMOUS language, can form the basis for the logical schemas and, consequently the physical schemas conform the transformations explained in section 3 (see Figure 33).

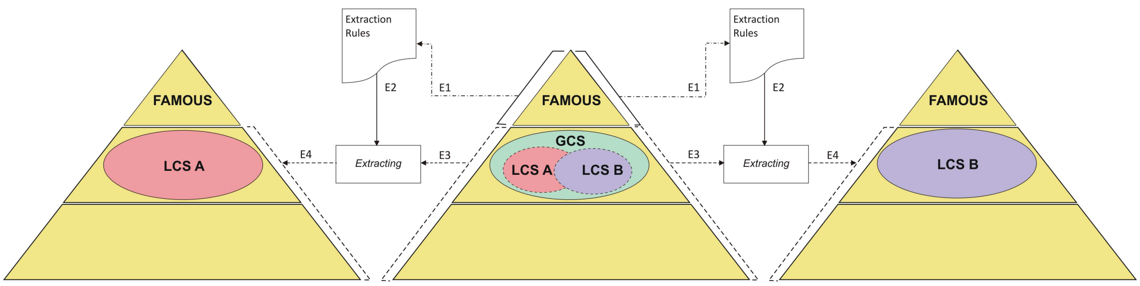

Figure 32: Extracting from the global conceptual schema the local conceptual schemas.

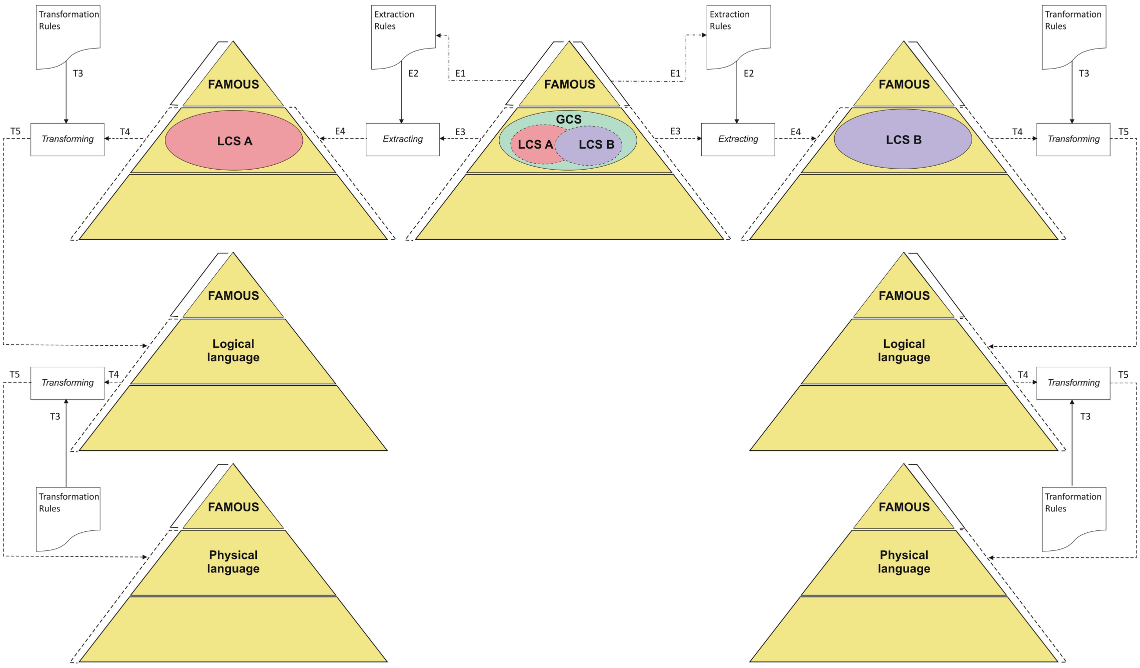

Figure 33: Overview of the FAMOUS-2 architecture.

Applying the FAMOUS-2 architecture as shown in Figure 33 to interoperability it is clear for the exchange of data between local conceptual schemas, which data can be exchanged and which cannot be exchanged. Because only data that corresponds with the overlapping part within the global conceptual schema for the extracted local conceptual schemas can be exchanged with semantic correctness.

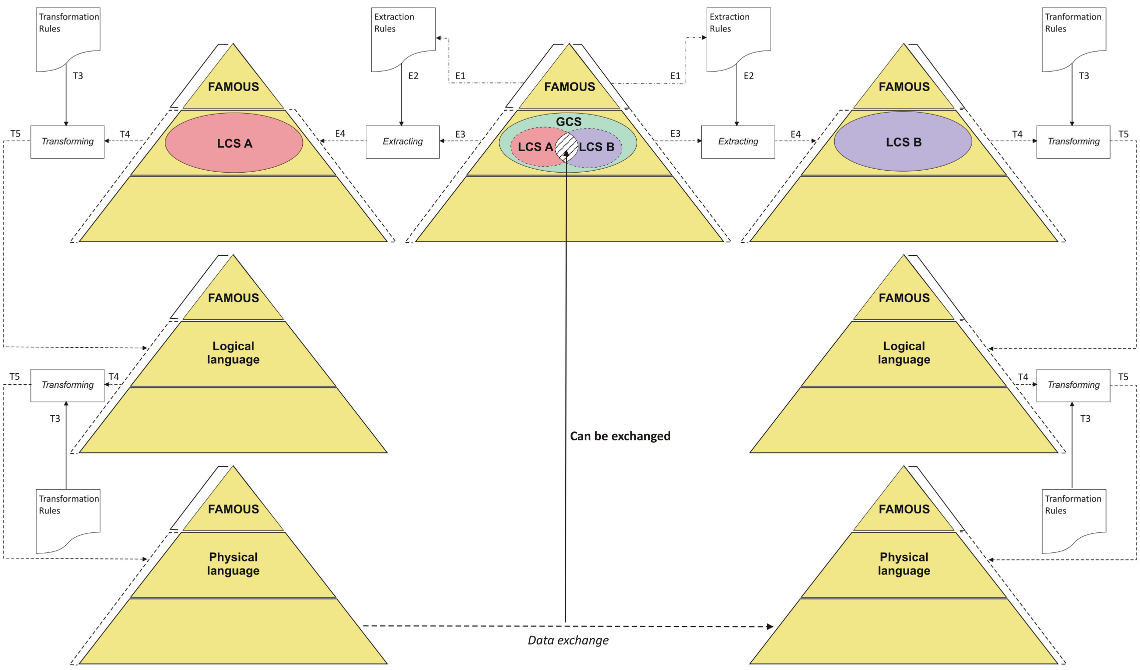

Figure 34: Semantic interoperability.