Christophe Debruyne and Pieter De Leenheer. Business Semantics as an Interface between Enterprise Information Management and the Web of Data: A Case Study in the Flemish Public Administration. In Marie-Aude Aufaure and Esteban Zimányi, editors, eBISS 2012, volume 138 of LNBIP, pages 208-233. Springer, 2013

Business Semantics as an Interface between Enterprise Information Management and the Web of Data: a Case Study in the Flemish Public Administration

Christophe Debruyne (1) and Pieter De Leenheer (2, 3)

(1) Semantics Technology and Applications Research Lab,
Vrije Universiteit Brussel, Brussels, Belgium
chrdebru@vub.ac.be
(2) VU University, Amsterdam, The Netherlands
(3) Collibra nv/sa, Brussels, Belgium
pieterdeleenheer@acm.org

Abstract. Conceptual modeling captures descriptions of business entities in terms of their attributes and relations with other business entities. When those descriptions are needed for interoperability tasks between two or more autonomously developed information systems ranging from Web of Data with no a priori known purposes for the data to Enterprise Information Management in which organizations agree on (strict) rules to ensure proper business, those descriptions are often captured in a shared formal specification called an ontology. We present the method Business Semantics Management (BSM), a fact-oriented approach to knowledge modeling grounded in natural language. We first show how fact-oriented approaches differ from approaches in terms of, amongst others, expressiveness, complexity, and decidability and how this formalism is easier for users to render their knowledge. We then explain the different processes in BSM and how the tool suite supports those processes. Finally, we show how the ontologies can be transformed into other formalisms suitable for particular interoperability tasks. All the processes and examples will be taken from industry cases throughout the lecture.

Keywords. Conceptual Modeling, Knowledge Management, Ontology Engineering, Business Semantics Management

1 Introduction

The increasing need for reusing and sharing information across peers in global value networks demands information systems to become Web-enabled and semantically interoperable. *Semantic interoperability* is defined as "the ability of two or more autonomously developed and maintained information systems or their (computerized) components to communicate data (using Web-based standards) and to interpret the information in the data that has been communicated in a meaningful manner" [6]. Most legacy information systems were developed in a time when these requirements were non-existing. The lack of interoperability is basically due to the different underlying formal semantics. The formal *semantics* of a (computer-based) system is the correspondence between this system and some real world as perceived by humans and usually given by a formal mapping of the system's symbols. As the real world is not accessible inside a computer, the world needs to be represented by an agreed conceptualization if we want to store and reason about semantics. Semantics are often stored in the shape of a formal (mathematical) construct. E.g., consider a particular car that is a real-world object and its license plate being a digitized reference in a database system. The formal semantics is defined by the correspondence between the car and it's unique license plate.

In order for systems to semantically interoperate, one has to have a shared understanding about this formal semantics. This is usually known as an *ontology* [14]. Ontologies constitute the key resources for realizing a Semantic Web [1]. While theoretically ontologies should be perfect renderings of a real world, in practice they evolve as successive approximations of it [15]. The problem is not so much what ontologies in computer science are, but how they come to be. The construction of ontologies is guided by appropriate ontology engineering methods. Ontology engineering is an advanced form of conceptual modeling. It requires the involvement of many parties, and they should be defined such that they are useful but also reusable. Rooted in knowledge management.

In this paper, we describe to develop and maintain ontologies for Web-based semantic interoperability. We have to address approaches from two worlds here: Web of Data and Enterprise Information Management. This article is organized as follows: in Section 2 we introduce the case for mind setting. In Section 3, we provide a background in semantic interoperability, including some of the challenges. Section 4 introduces the Business Semantic Management Method, describing the formalism, framework and brief description of the two processes: semantic reconciliation and semantic application. These two processes are then described in more detail using examples from the case in Sections 5 and 6 respectively. We then conclude this paper in Section 7.

2 The Flanders Research Information Space (FRIS) Case

For a country or region in the current knowledge economy, it is crucial to have a good overview of its science and technology base to develop an appropriate policy mix of measures to support and stimulate research and innovation. Also companies, research institutions and individual researchers can profit from the information maintained in such a portal. EWI[1] thus decided to launch the Flanders Research Information Space program (FRIS) to create a virtual research information space covering all Flemish players in the field of economy, science and innovation. The current version of this portal[2] contains, for instance, mash-ups of data on key entities (such as person, organization, and project; and their relationships) on a geographical map. **Fig. 1** contains a screenshot of the current FRIS portal.

[1]The Department of Economy, Science and Innovation (Economie, Wetenschap en Science in dutch) of the Flemish Government `http://www.ewi-vlaanderen.be/`
[2]`http://www.researchportal.be/`

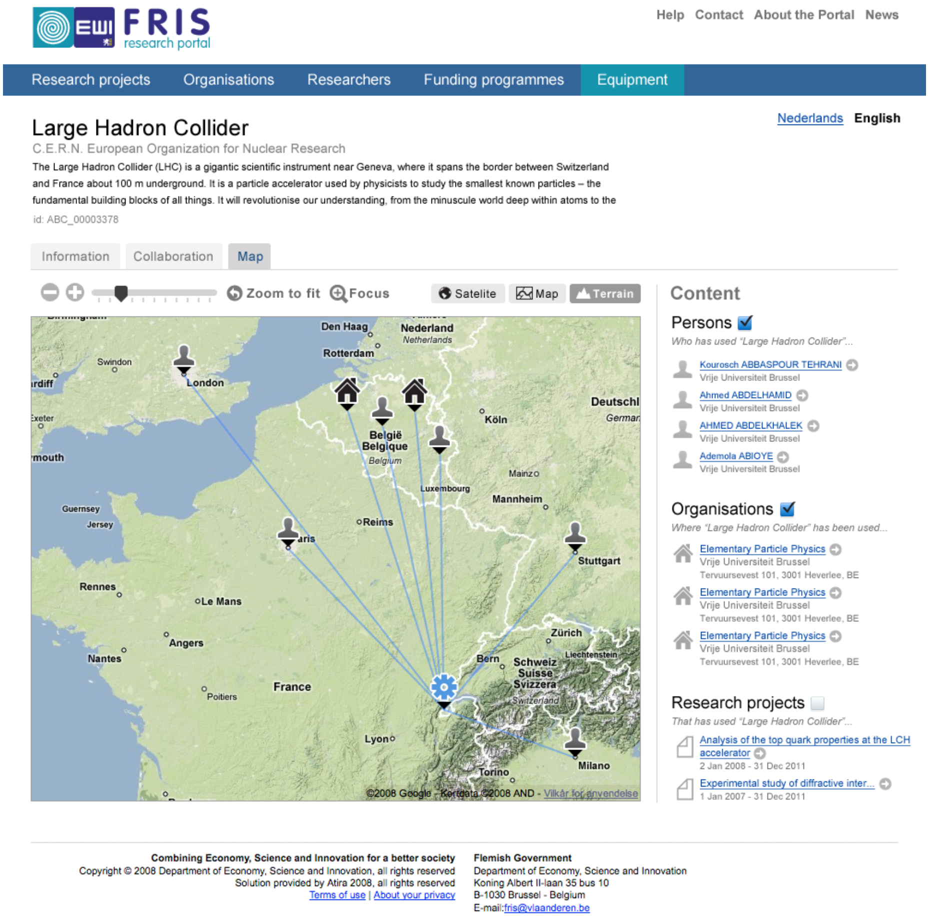

Fig. 1. FRIS already provides a European map visualizing data about international cooperation between individuals, organizations and projects, e.g. in the context of the Large Hadron Collider.

Another aim of FRIS is to reduce the current administrative burden for universities as they are confronted with repeatedly reporting the same information in different formats to various institutions. Universities receiving funding from the Flemish government are asked to regularly report the same information to different organizations (local and international). As there is little alignment between those reports, universities are confronted with repeatedly sending the same information in other formats, other structures or according to different classifications, not always compatible with each other. This creates a heavy administrative burden on these knowledge institutions. Universities furthermore store their information in autonomously developed information systems, adding to the complexity of the problem. As the EU also wants to track all research information in Europe, they ask all universities to report using the

Common European Research Information Format (CERIF)[3], a recommendation to EU-members for the storage and exchange of current research information. If all information would be centralized and accessible in a uniform way, creating services for such reports, would greatly facilitate the reporting process.

While the CERIF model, created with Entity-Relationship (ER) [3] diagrams, allows for an almost unlimited flexibility on roles and classifications used with entities, the actual approach has shown its limitations when it comes to communicating the modeled domain knowledge to domain experts and end users. The learning curve for the domain experts to understand the ER model and translate it back to the conceptual level is quite steep [38]. For instance, the example in **Fig. 2** (taken from [38]) shows the complexity of adding (multilingual) attributes to relations between core entities Person `cfPerson` and Project `cfProject`. This relation is represented by `cfPerson_Project` (linked by the two identifiers of the linked entities). In the same way, the example shows the CERIF entity `cfProject` and its relationship with the entity `cfClassification`: cfProject`_Classification`. A CERIF relationship is always semantically enriched by a time-stamped classification reference. The classification record as such is maintained in a separate entity (`cfClassification`) and allows for multilingual features (`cfClassificationTerm` and `cfClassificationDescription`). Additionally, each classification record or instance requires an assignment to a classification scheme (`cfClassificationSchemeIdentifier`). The management of the classification terms and classification schemes is organized in what is called the CERIF Semantic Layer [23].

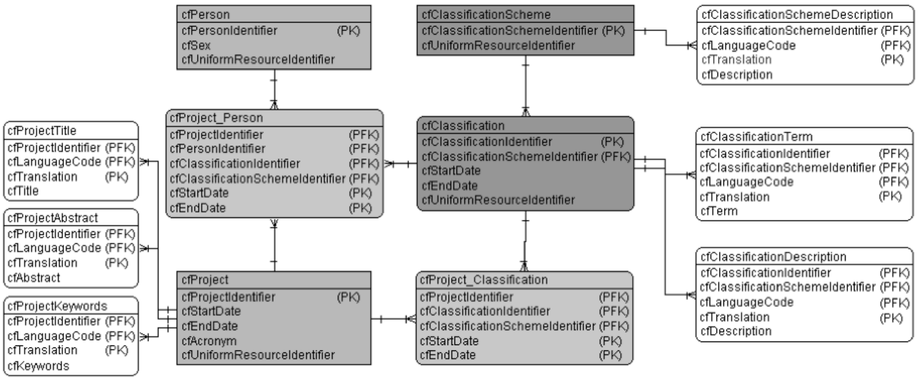

Fig. 2. The CERIF entity `cfProject` and its relationship with the entity `cfProject_Classification` (linked by the two identifiers of the linked entities). A CERIF relationship is always semantically enriched by a time-stamped classification reference. The classification record is maintained in a separate entity (`cfClassification`) and allows for multilingual features. Additionally, each classification record or instance requires an assignment to a classification scheme (`cfClassificationSchemeIdentifier`).

[3] `http://cordis.europa.eu/cerif/`

Semantic mismatches occur at different levels: 1) terminology, 2) relations and 3) business rules. Due to this semantic layer, mismatches between stakeholders that need to interoperate via the CERIF standard occur at the first two levels. An example of these two mismatches on relation level is shown in **Fig. 3**. In this figure, two organizations use a different relation to denote that a particular researcher is the leader of a research project.

Thus, next to the conceptual complexity of the CERIF model aimed at flexibility, this flexibility also give rise to interoperability problems as heterogeneous representations for concepts and relations can be modeled.

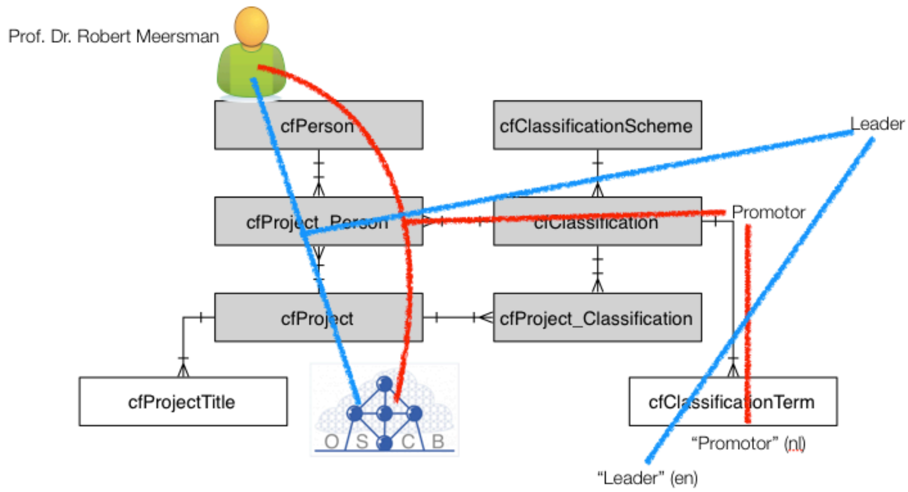

Fig. 3. Mismatch at "relation" level: two application refer to the relation between a researcher and a research project; one referring to the person as a leader of this project, the other as the promoter ("promotor" in Dutch).

To populate the FRIS portal with all information provided by the delivered CERIF files and other heterogeneous sources, needed are: 1) Consensus amongst the involved parties on a common conceptual model for CERIF and the different classifications (inside that semantic layer); 2) An easy, repeatable process for validating and integrating the data from those sources; 3) Make available the information in a generic way on the Web on which third parties can develop services as demonstrated by other Linked Data initiatives.

We furthermore have to take into account the non-technical expertise of most of the domain experts. From these requirements, it becomes clear that integrating all information and reducing the administrative burden faces some problems for which appropriate data governance methods and tools are needed. Before we present the Business Semantics Management method and its tool support, we provide the reader a background on system interoperability.

3 Background

Information systems that satisfy at least one formally specified semantic interoperability requirement are called *open information systems*. This is in contrast with *closed information systems*, where a data model represents the structure and integrity specification of the data of only the applications belonging to (often) a single enterprise. The vocabulary inside that data model in general is not a priori intended to be shared with other applications [31], i.e. the transitions caused by a closed information system are only meaningful within this system. For open information systems, however, a common vocabulary needs to be developed – and agreed upon – to which the different systems will commit to.

On the Semantic Web, a great deal of ontologies are developed in RDF(S) or OWL [39]. Both are W3C recommendations for knowledge representation languages on the Web. RDF(S) allows for the creation of simple vocabularies (concepts and relations). However, the elements provided by RDF(S) are very basic, offering little possibilities to model complex rules or constraints. The Web Ontology Language (OWL) is a family of knowledge representation languages that are more expressive than RDF(S) tailored to support some reasoning tasks such as consistency checking. Depending the "flavor" used, a particular OWL language is more expressive than another. An increase in expressiveness, however, is at the cost of efficiency or even decidability.

These ontologies are the result of knowledge management activities within a community (be it an organization, a group of organizations, etc.). Knowledge management aims at using knowledge as a production factor and comprises a range of strategies used in an organization to identify, create, represent, share, and adopt knowledge and information. Knowledge can be either elicited from individual persons or are embedded in organizations as processes or practices. Whenever two or more organizations need their autonomously developed information systems to interoperate (i.e. exchange and communicate information, do "business" together), knowledge management activities help support the group of organizations in establishing consensus on a common approximation of the real world to ensure a proper and smooth system-interoperation. Knowledge management is an important activity for both *Enterprise Information Management* (EIM) and the *Web of Data*. The first aims at satisfying the information technology needs emerging from an organization's requirements, e.g. ensure proper business. EIM is thus a "top down" application of knowledge management. The latter aims at structuring and providing existing data in such a way (third party) services can be easily created on top of that structured information ("bottom up").

3.1 Reusability vs. Usability of Ontologies

In many cases, ontologies contain references to the instances used in the application or application domain, and domain rules [35]. Those domain rules typically contain constraints of identity, cardinality, mandatoriness, etc. and thus restrict the semantics (i.e. interpretation) in a specific conceptualization of a particular application domain. In other words, these rules must be satisfied by any application that wishes to commit to such an interpretation for an ontology in order for interoperability to work [14]. However, providing rules that are important for effective and meaningful interoperation between applications may (and will) limit the generality of an ontology [35]; in other word the increase of business rules decreases the generality of ontologies. This renders ontology modeling turns out to be far from trivial. Lightweight ontologies that hold none or few domain rules however are not very effective for communication between autonomously developed and maintained software systems. A requirement for different organizations in a certain domain to communicate is to have a common understanding about a relevant part of that domain. In other words, the more an ontology becomes intended for a particular application domain (more requirements, more business rules), the less general the ontology becomes.

3.2 Context of the Ontology Application

The aforementioned challenge corresponds with the variation of requirements for the Web of Data and Enterprise Information Management. The Web of Data needs meaningful annotations of data sources to enable machines to access, process and apply that information. Describing existing (legacy) data can be done with lightweight ontologies. However, as more business rules are needed to ensure proper business within the community of stakeholder, EIM will be applied to capture the requirements on how and under what conditions data will be exchanged, even up to the point how certain things have to be encoded. The Web of Data and EIM are thus residing in two different business domains and have different business drivers. The first annotates the data bottom up for third parties to develop a priori unknown services. On the other side you need top down planning with EIM to facilitate business.

The process of reaching that common understanding will involve dialogue; dialogue based on the perspectives of (ideally all) involved stakeholders. A perspective intends to capture the meaning within a given or assumed context on what the stakeholder thinks is currently relevant to the community he is in.

This semantic gap is also noticed in the discrepancy between the need for interoperability within enterprises and the actual implementation of solutions [28]. Also in the cloud computing community, the role of platform-agnostic semantic modeling is coming back (see e.g., [32]).

3.3 Requirements for a Method

Community involvement is essential for semantic interoperability. Enabling communities to develop and maintain a representation of their (business) world needs a method since reaching a common agreement between many stakeholders proves to be difficult [12]. Community involvement is crucial for facilitating the uptake and governance of, for instance, Linked Data, a set of practices for annotating and exposing data sets on the Web for which the community ultimately needs to reach an agreement on the meaning of such annotations. The Linked Data initiative relies on RDF and URI mechanisms to represent these annotations, which cannot directly map on the language of the human community. It turns out that appropriate methods for this can learn from database modeling following the principles below.

- **Technology matures.** The non-involvement of non-tech savvy domain experts is not longer an excuse. For instance, wiki technology has been put forward as a mean to reach agreement and share knowledge about different subjects over the past decade [20]. The advantage of such technology is that anyone can add content without much technical knowledge and have already been adapted in the field of ontology engineering to enable non-technical users to create, visualize and maintain ontologies.
- **Analyzing natural language discourse.** Database design methods such as NIAM [40] and ORM [18] already showed that the closer the link between human natural language communication and the system and/or business communication that results from it, the more likely such systems will work as intended by their various stakeholders. This is particularly important for interfaces where humans, systems and businesses interact, as the human discourse needs to be mapped meaningfully onto application symbols. Since people naturally communicate with words, pictures, and examples, the best way to arrive at a clear description of the domain is to use natural language, intuitive diagrams, and examples. These techniques furthermore allow scalable solutions to ontology engineering through a classical *separation of concerns* - as done in databases - by separating the schema level from the instance level. As a consequence, applications become minimally sensitive to changes in data representation.
- **Employing legacy data,** output reports, and interviews with domain experts as fulcrum for leveraging validation. The source (or context) of a certain fact needs to be traceable for future reference. In the case of ontology engineering: lift data models into ontologies by removing application specific context (e.g., non-conceptual identifiers such as an automatically incrementing key).

One method for collaborative ontology engineering that complies with the three principles above is Business Semantics Management.

4 Business Semantics Management

For the last twenty years, many methods have been put forward for how to develop ontologies. It seems, however, that research on methods has diminished in recent years [2]. Bergman (2010) noted that very few discrete methods exist and those that do are often older in nature [2]. He furthermore noted that most methods shared a number of logic steps from assessment to deployment, from testing to refinement.

Quite a few surveys on the state of the art on ontology engineering methods exist. Recent surveys include [34], [33] and [13]. Corcho et al. (2003) observed that there is often no correspondence between ontology building methods and tools [5]. For both the DOGMA initiative [24, 22] and Business Semantics Management (BSM), suitable tools for adequate support of these methods were developed.

BSM prescribes steps and processes for bringing a community of stakeholders together to realize the reconciliation of their heterogeneous metadata, and consequently the application of the derived business semantics patterns in partial fulfillment of well-established semantic interoperability requirements. We identify six principles of Business Semantic Management [6]:

1. **ICT Democracy** An ontology should be defined by its owning community, and not by a single developer. In the FRIS case, the community of stakeholders contains - amongst others - the Flemish government, funding agencies, and knowledge institutions (universities).
2. **Emergence** Semantic interoperability requirements emerge autonomously from community evolution processes. By default, business semantics serve "open" information systems, and hence the requirements and limitations for semantic interoperability cannot be entirely known before completion.
3. **Co-evolution** Ontology evolution processes are driven by the changing semantic interoperability requirements. In contrast to waterfall-like approaches that focus on a broad design upfront, agile methods perform short milestone-driven revision iterations in order to cope with dynamic environments.
4. **Perspective Rendering** Ontology evolution processes must reflect the various stakeholders' perspectives. There is no generally applicable ontology, as each application will generate a contextualized model to match local needs and functionalities. Conflicts will arise from differences in how domains are perceived by the stakeholders. The different knowledge institutions in Flanders, for instance, use different classification schemes for scientific publications.
5. **Perspective Unification** In building the common ontology, relevant parts of the various stakeholder perspectives serve as input for the unified perspective [29].
6. **Validation** The explicit rendering of stakeholders' perspectives allows us to capture the ontology evolution process completely, and validate the ontology against these perspectives respectively.

Ultimately, co-evolving communities with their ontology will increase overall stakeholder satisfaction.

Based on the above principles, we devised a teachable and repeatable method and system for fact-oriented BSM. The representation of business semantics is based on the DOGMA [25] ontology framework. BSM draws from DOGMA-MESS (a collaborative ontology engineering method developed on top of the DOGMA framework, first introduced in [12], further formalized in [9, 30, 6] and implemented in [4, 10]), and best practices in ontology management [19, 36] and ontology evolution [11].

4.1 Fact-Orientation

The fact-oriented paradigm that was introduced in the conceptual modeling approach NIAM (pre Object-Role Modeling). NIAM simplifies the design process by using natural language, as well as intuitive diagrams[4], which can be populated with examples, and by examining the information in terms of simple or elementary fact types. In other words, to simplify the modeling task, stakeholders examine the information in the smallest units possible: one elementary fact at a time. By expressing the model in terms of natural concepts, like objects and roles, it provides a conceptual approach to modeling. NIAM was further refined into Object-Role Modeling, or ORM. ORM's rich graphic notation is capable of capturing many business rules that are typically

[4] In this paper, we will not go into details of ORM diagramming. More information on these diagrams can be found in [18].

unsupported as graphic primitives in other popular data modeling notations (e.g., role hierarchies).

Moreover, breaking down the domain into several elementary fact types reduces the problem complexity into smaller and thus more easily manageable subproblems. This leverages the potential of domain experts to effectively externalize conceptions that were not revealed otherwise [16, 17, 41].

NIAM/ORM's attribute-free approach, as opposed to frame-based techniques such as UML or (E)ER, promotes semantic stability. Semantic stability is a measure of how well models or queries expressed in the language retain their original intent in the face of changes to the application [16]. The more changes one is forced to make to a model (or query to cope with an application change), the less stable the model is. In BSM, semantic interoperability is promoted by elementary fact types that are the fundamental conceptual units of information, and are uniformly represented as relationships. How they are grouped into structures is not a conceptual issue. Given the co-evolution principle, it is critical that the underlying ontology be crafted in a way that minimizes the impact of these changes. Therefore regarding our objectives, fact-oriented models are more stable under business changes than e.g., UML or (E)ER models.

ORM models can be easily verbalized and populated for validation with domain experts, they are more stable under changes to the business domain, and they typically capture more business rules in diagram form. For instance, given the fact type:

Project, having, of, Acronym

The combination of following constraints state that a Project is totally and uniquely identified by its Acronym:

- Each Project having at most 1 Acronym
- Each Project having at least 1 Acronym
- Each Project is identified by Acronym of Project

For conceptual modeling (of information systems), the ORM method has thus several advantages over the (E)ER and UML approaches. (E)ER diagrams and UML class diagrams are closer to the final implementation, so they also have value [18] by providing "implementable" summaries of the conceptual model. In doing so, (E)ER and UML take into account constructs related to the implementation that are not relevant to the conceptualization (e.g., the difference between an attribute and a relation). The *late aggregation principle* - the act of postponing whether an object becomes an entity or an attribute until the implementation of a database is done - is well known, and fundamental, in database modeling [24] and improves the maintainability of the schema. As fact-oriented modeling techniques do not make this distinction - everything is a fact type – the modelers do not even have to consider these aspects, rendering the conceptual modeling easier.

The Semantics of Business Vocabulary and Business Rules (SBVR) [27] is an adopted standard of the Object Management Group (OMG) pushed by the business rule community and the fact-oriented modeling community. SBVR provides a fact-oriented framework for describing the semantics of terminology used in a business,

business facts and business rules. The advantage of SBVR is the fact that it is an integral part of OMG's model driven architecture. SBVR uses OMG's Meta-Object Facility (MOF) [26] to provide interchange capabilities; transforming (parts) of a model into other formalisms with a MOF model (e.g., UML). MOF is essentially a set of concepts that can be used to define other modeling languages. SBVR models can be structurally linked at the level of individual facts with other MDA models based on MOF. Driven by its success in conceptual data modeling, the fact-oriented approach of SBVR provides the basis for formal and detailed natural language declarative description of complex business entities.

The structure of SBVR (illustrated in **Fig. 4**) allows implementing a business semantics system that takes into account the existence of multiple perspectives on how to represent concepts (by means of vocabularies), and includes the modeling of a governance model to reconcile these perspectives pragmatically (read: insofar practically necessary) in order to come to an ontology that is agreed and shared (by means of communities and speech communities) [8].

- A *semantic community* is a group of stakeholders having a body of shared meanings. Stakeholders are people representing an organization or a business unit. They already informally share knowledge via social network functionality.
- A *body of shared meanings* is a unifying and shared understanding (perception) of the business concepts in a particular domain. Concepts are identified by a URI. The scope of this body emerges from breakdowns during informal knowledge sharing.
- A *speech community* is a sub-community of a semantic community having a shared set of vocabularies to refer to the body of shared meanings. A speech community groups stakeholders and vocabularies from a particular natural language in a multi-lingual community, or from a certain technical jargon.
- A *vocabulary* is a set of terms and fact types primarily drawn from a single language to express concepts within a body of shared meanings.

The notion of vocabularies allows multi-linguality or within one language synonymous terms may refer to the same set of concepts, or a polysemous term may refer to different concept URIs depending on the vocabulary it is residing in. The following function maps a term in a vocabulary to a concept URI: $concept{:}Vocabulary \times Term \rightarrow URI$. For the full formalization, we refer to [9]. E.g., consider a term "student" in a Dutch vocabulary and a term "étudiant" in a French vocabulary, both meaning the same thing. Both terms are equal if and only if *concept(Dutch, student)* and *concept(French, étudiant)* refer to the same URI.

Fact-oriented models are not only suitable for modeling conceptual models for information systems. NIAM and ORM were successfully adopted for ontology engineering in a method called DOGMA.

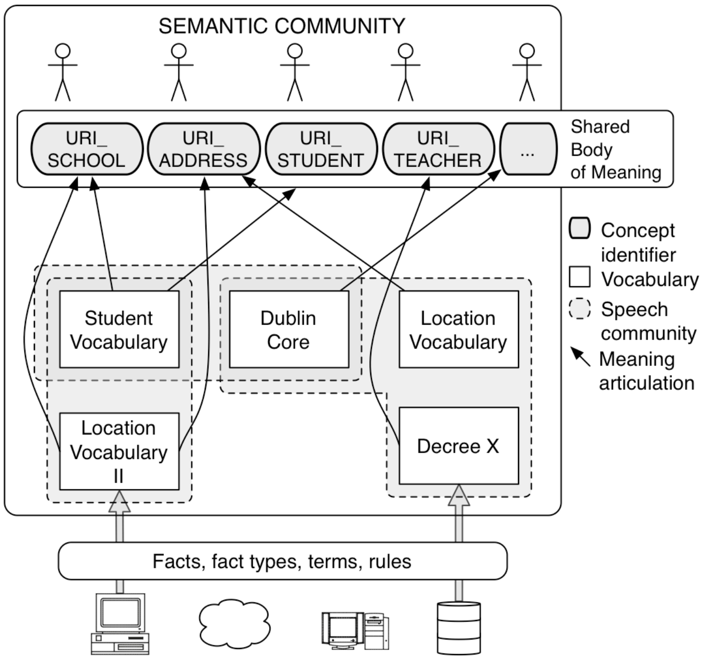

Fig. 4. The structure of business semantics: communities, stakeholders, concepts, vocabularies, facts, and rules. Speech communities are sub-communities of a semantic community having a shared set of vocabularies to refer to the body of shared meanings. Meaning articulations record those references. Applications and their symbols are then mapped onto the different fact types, terms and rules agreed upon within the community.

4.2 Development of Ontology-Grounded Methods and Applications

In the previous section we briefly described fact-oriented modeling. In this formalism, the basic knowledge building block is a fact-type; a generalization of facts encountered in the world. Initially used for developing closed information systems, it was successfully applied for modeling ontologies in the DOGMA *framework* for ontology engineering. The DOGMA framework that we will present in this section thus follows the fact-oriented paradigm.

Ontologies in DOGMA allow the application world to be associated with a lexical world relying on the fact that the knowledge building blocks expressed in natural language are easily obtained and agreed upon. These building blocks - called *lexons* [25] - only need in principle to express "plausible" fact types (as perceived by a community of stakeholders) in order to be entered into the Lexon Base, a repository containing large sets of such lexons. A lexon is formally described as a 5-tuple $\langle \gamma$, *headterm, role, co-role, tailterm*$\rangle$, where γ is an abstract *context identifier* pointing to a resource such as a document on the Web. The context identifier is assumed to identify unambiguously (to human users at least) the concepts denoted by the term and role labels. For example the lexon: $\langle \gamma$, *Person, with, of, First Name*$\rangle$, can be read as: in the context γ, *Person* plays the role of *with First Name* and *First Name* plays the role of being *of Person*. The Lexon Base may contain redundant lexons, even apparently "contradictory" ones, but lexons are meant to be highly reusable and so provide semantic leverage.

The *Commitment Layer* contains ontological commitments that use a selection of lexons to annotate applications and specify constraints defining the use of the concepts in the ontology. DOGMA distinguishes two types of ontological commitments: *community commitments* and *application commitments*. The first denotes a meaningful selection of lexons, and constraints that capture the intended semantics of the data that the stakeholders want to interchange for a particular application. The latter extends the community commitment mappings describing how application symbols of one individual application commit to the ontology. The application commitment can furthermore contain additional lexons and constraints that describe how the application - as a whole - commits to the ontology [37]. Individual applications committing to the same ontology can thus have different sets of constraints. The act of selecting and constraining a meaningful selection of lexons for a particular application is called the *double articulation principle* [35]. How the lexons are used in a specific application, and the complexity associated with that use, are delegated to the ontological commitment. The use or pragmatics of lexons are thus the responsibility of the application.

Because of the resulting separation of concerns, DOGMA's layered approach does not map one-on-one with ontologies implemented in OWL. In OWL, instances can reside next to their schema and properties are immediately constrained. DOGMA keeps the instances out of the ontology and leaves (all) interpretation and constraining of a fact type to the commitment layer. Ontologies in DOGMA are easily transformed into RDF(S) or a similar formalism and allows reasoning over domain terminology, by the late aggregation principle.

In this section, we presented the DOGMA framework to ontology engineering. What is lacking is a method for collaboratively building ontologies on top of this framework. One such method was DOGMA-MESS, in which MESS stood for Meaning Evolution Support System. We will not provide details on DOGMA-MESS, but note it was the basis for BSM. Thus in the next section, we will present the BSM method.

4.3 Business Semantics Management: Semantic Reconciliation & Application

BSM draws from best practices in ontology management [19] and ontology evolution [11]. The representation of business semantics was originally based on the DOGMA approach and provides a method and tool that enable parties to (i) obtain consensus on (the semantics of) key business terms, and (ii) evaluate this consensus uniformly in various applications throughout the organization. Respectively, BSM consists of two complementary cycles: semantic reconciliation and semantic application (see **Fig. 5**) where each cycle groups a number of activities.

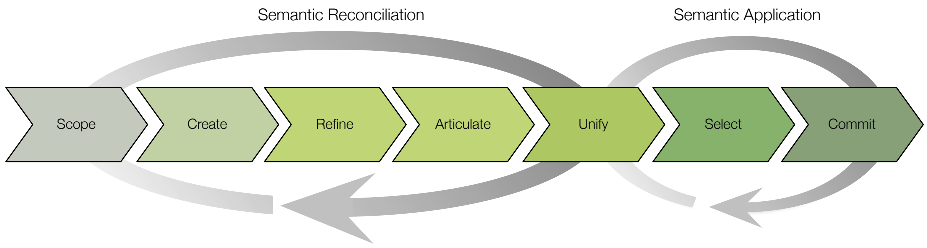

Fig. 5. Business Semantics Management consists of two complementary cycles: semantic reconciliation and semantic application. Both cycles communicate via the unify-activity.

- *Semantic Reconciliation* is the first cycle of the method. In this phase, business semantics are modeled by extracting, refining, articulating and consolidating fact types from existing sources such as natural language descriptions, existing metadata, etc. Ultimately, this results in a number of consolidated language-neutral semantic patterns that are articulated with informal meaning descriptions (e.g., WordNet[5] word senses). These patterns are reusable for constructing various semantic applications.
- *Semantic Application* is the second cycle. During this cycle, existing information sources and services are committed to a selection of semantic patterns. This is done by selecting the relevant patterns, constraining their interpretation and finally mapping (or committing) the selection on the existing data sources. In other words, a commitment creates a bidirectional link between the existing data sources and services and the business semantics that describe the information assets of an organization. The existing data itself is not moved nor touched.

As DOGMA's lexons and constraints are fully compatible with SBVR (supported by OMG), BSM recently adopted SBVR for representing the business domain and rules. SBVR does provide constructs that were not available in the DOGMA framework, such as support for unary fact types to represent characteristics of a business entity (e.g., Project is terminated).

The derived formal vocabularies and rules can be interpreted and used by computer systems to develop Web, software and business intelligence applications. This constitutes the semantic application of business semantics. As mentioned in a previous section, MOF provides bridges to link SBVR to OWL, RDF(S), UML, ER, etc. Via MOF, business semantics in SBVR forms the basis for forward engineering of software (i.e. UML diagrams), business intelligence (i.e. OMG common warehouse model), and Web applications (W3C RDF(S) and OWL) and vice versa: existing models can be reverse engineered to feed the BSM process.

Rather than presenting more detail on the different steps in this section, we will present the tool supporting the BSM method and work out the different steps whilst describing the tool.

---

[5] `http://wordnet.princeton.edu/`

5 Semantic Reconciliation with Business Semantics Glossary

The Business Semantics Glossary (BSG) supports the semantic reconciliation processes of BSM. BSG is a Web-based software application aimed at both business as well as technical users. It lets people collaboratively manage their business semantics according to the BSM method. BSG is based on the Wiki paradigm that is a proven technique for stakeholder collaboration and is essential for evolving business semantics.

Fig. 6 illustrates the concept page (identified by a URI) in BSG for term `Project` in the BSG. The page consists of a gloss providing a natural language description; a number of fact types (e.g., `CFProject executed by CFOrganization`); a number of rules; examples; notes; and synonyms. Governance models are built-in and user roles (e.g., steward, stakeholder, as shown in **Fig. 6**) can be applied to distribute responsibilities and increase participation. The software takes care of the audit trails who changed what, when and why. Fine-grained permission and rights management decide which users or user groups can view/edit/monitor/ etc. different parts of the business semantics.

In this case, the BSG aims to provide a single point of reference for Flemish Public Administration's business vocabulary and rules. The different processes of semantic reconciliation are explained and exemplified with the use case in the Flemish Public Administration.

The information shown in **Fig. 6** is the result of the application of the BSM method. In this section, we will describe each of the semantic reconciliation phases with examples from the FRIS case.

5.1 Scope

Scope sets out the scoped terms that are actually needed to establish semantic interoperability. Specific business drivers that want to resolve a weakness or threat in a certain application context fuel this activity. Regarding the considerations made above, a distinction between information technology or information system (IT/IS) and business contexts is made.

In an IT/IS Context, a communication breakdown may be caused by an inadequate transformation of incoming personnel data from the more than 1,500 educational institutions to the data semantics of the central salary system. The breakdown here is caused by a lack of specification of terms such as "personnel" and "salary". The derived need for manual translation (e.g., using XSLT) introduces a weakness, as defining the translation requires know-how about the respective formats. Moreover, such a translation introduces even more legacy that is difficult to interpret.

In a business context, the lack of a uniform and unambiguous meaning of the term "study area" following externally imposed rules may form a legal threat. This observation initiates another semantic reconciliation cycle where metadata related to "study area" are to be reconciled.

In any context, it is important to involve the relevant stakeholders in this process and assign them with appropriate roles and responsibilities within communities. Note that the scoping process in this paper was oversimplified; consult [7] for supporting scoping techniques.

In a previous section, we showed how SBVR foresaw structure for modeling communities (semantic and speech) and their respective communities. This was adopted in BSG and roles can be assigned to members within each community. **Fig. 7** shows how these structures can be navigated in BSG.

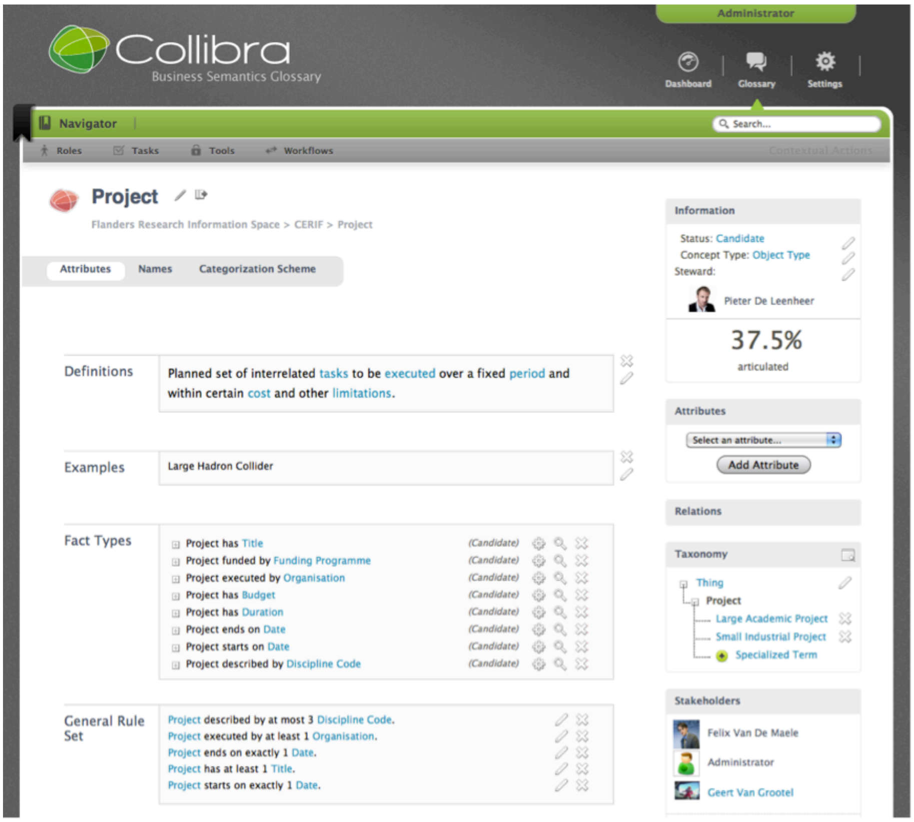

Fig. 6. Screenshot of the definition for term `Project` (a CERIF term) in the `Project` vocabulary taken from BSG that currently deployed at the Flemish public administration. Even though the concept definitions look like natural language, thanks to the underlying MOF-compliant SBVR meta-model, one can automatically generate an enterprise information model from it that provides a formal specification in UML, XSD or the like. Governance is built-in and roles can be applied to distribute responsibilities. Here, the user Pieter De Leenheer is a steward.

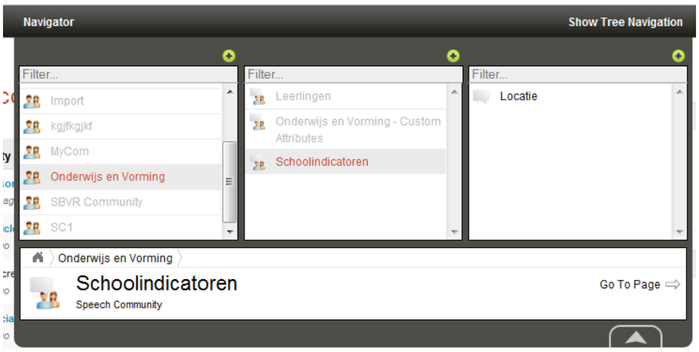

Fig. 7. Screenshot of the BSG navigator in which the user can browse through the different vocabularies. In this example the user first chooses the semantic community "Education and Training" in the first column and then the speech community "School Indicators" in the second column. This displays all the vocabularies used by that community, in this case the "Location" vocabulary can be seen.

5.2 Create

During this activity, every scoped term is syntactically defined and rules for these terms and the roles they play in their fact types are created as well. During this phase, inspiration can be drawn from existing sources (manuals, users, standards, etc.). For example, in the FRIS case, terminology may be reused from the CERIF standard:

- CFProject executed by / executes CFOrganization
- CFPerson having / of Person_Name
- CFPerson having / of CFPersonAddress
- CFPersonAddress of / used in CFAddress
- CFPerson affiliated with / with affiliation CFOrganization_UNIT
- <u>EACH</u> CFPerson having <u>EXACTLY</u> <u>ONE</u> Person_Name
- …

To each scoped term, there are also certain roles appointed such as a "concept steward" and a number of relevant stakeholders. The definition is fed by implicit know-how from the involved domain experts, or by automatic extraction of facts from existing metadata (see [7] for a review of ontology extraction techniques).

5.3 Refine

During this activity, fact types (and constraints) that were created during the creation activity are refined so they are understandable to both business and technology. The refined fact-types and constraints are i) correct, ii) useful, iii) reusable, and iv) elegant. During this activity, additional fact types can be created by means of objectification (regarding a fact type as a concept, playing a role with terms of the original fact type) or capturing missing links and relation (e.g., transforming an attribute of an entity into an attribute of a second entity related to the first entity).

In the FRIS case, the somewhat technical term `CFProj` becomes `Project` or `EmplAddr` is decomposed into a fact type `Employee is located at / locates Address`. Coding conventions can be applied here to guide the process. Below we find a set of refined fact types and constraints based on the list from the previous section:

- Project executed by / executes Organization
- Person having / of Person_Name
- Person located at / locates Address
- Person with / of Affiliation
- Organization_Unit with / of Affiliation
- <u>EACH</u> Person having <u>EXACTLY</u> <u>ONE</u> Person_Name
- <u>EACH</u> Affiliation of EXACTLY ONE Person
- <u>EACH</u> Affiliation of EXACTLY ONE Organization_Unit
- <u>EACH</u> Affiliation *a* <u>IS IDENTIFIED BY</u> Person with *a*
<u>AND</u> Organization_Unit with *a*
- …

5.4 Articulate

Create informal meaning descriptions as extra documentation. These descriptions include *definitions* and *examples* and can serve as anchoring points when stakeholders have used different terms for the same concepts (i.e., detecting synonyms). Where available, already existing descriptions can be used (e.g., the euroCRIS website on CERIF) to speed up the process and facilitate reuse.

Since multiple users may render their perspective concurrently on a term, it may be that after the refine activity some fact types and rules impose contradicting statements. During this activity, conflicts and inconsistencies are removed. Specifically designed algorithms may help here. E.g., in The Netherlands, an address is uniquely identified by a combination of postcode and house number, while in Belgium a combination of postcode, street name and street number is required. Articulating these differences is crucial in order to be able to deal with different data integrity rules during information exchanges. **Fig. 8** depicts an example of a definition and example of the term “Project” in the FRIS Case.

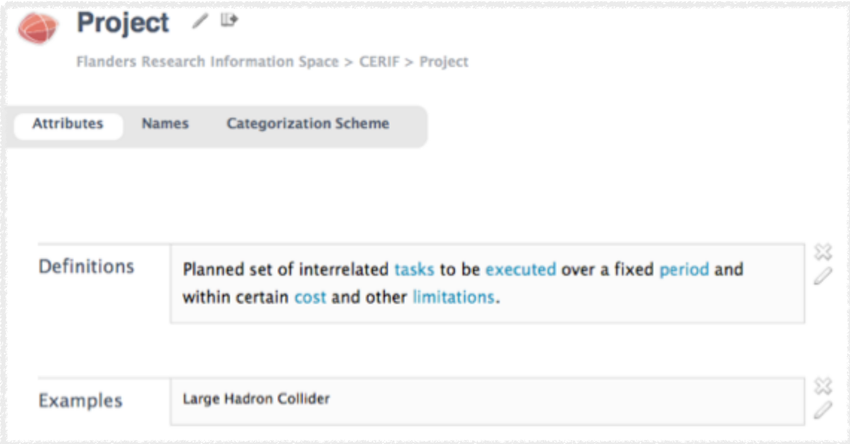

Fig. 8. Screenshot of the definition and an example for the PROJECT in the Project vocabulary of the CERIF speech community.

5.5 Unification

During unification a new version of the EIM is generated, which is a "flattened" version of the BSG that is generated in a timely manner. The EIM is the product of semantic reconciliation and serves as a uniform technical specification to implement semantic applications.

In order to optimally consolidate equivalent groups in vocabularies, one has to check for each of these groups where redundant conceptual patterns could be combined, and note any arithmetic derivations. For instance:

- Can the same concept be a member of two concept types? If so, combine the concept types into one (unless such identities are not of interest).
- Can two objects instantiating two different concept types be meaningfully compared? Do they have the same unit or dimension? If so, combine the concept types into one.
- Is the same kind of information recorded for different entity types, and will you sometimes need to list the entities together for this information? If so, combine the entity types into one, and if necessary add another fact type to preserve the original distinction.
- Is a fact type arithmetically derivable from others?

The consolidation is finished if you were able to remove all noteworthy redundancies.

6 Semantic Application

In the previous section, we elaborated on the BSM processes that lead to descriptions of a domain, agreed upon by a community of stakeholders. These descriptions better approximate reality over time (i.e. with each iteration). Once a new version of the EIM is created, these can be applied to support the semantic interoperability requirements of that community. Through the underlying MOF framework, this EIM can be represented in many formats, such as UML, OWL, or XSD, serving a wide variety of applications.

Conceptually, we distinguish two activities: select and commit.

- **Select** Given an application context (such as a workflow or business artifact), relevant concepts are selected from the EIM for a particular application. It may be required to add additional application-specific constraints that could not be agreed upon on the community level, or that are currently not supported by SBVR.
- **Commit** Information systems are improved using the selected concepts. Depending on the application context, this can be implemented in different ways. Concretely, this boils down to data transformation, validation, and governance services. For example, two or more XML structures can be virtually integrated by defining XSLT transformations to a shared XSD-formatted EIM. The EIM may also be used to convert relational databases into RDF triple stores (cf. RDB2RDF initiative). Here, the application of an EIM to generate data transformation services is illustrated.

Selection and commitment thus also involves choosing the appropriate formalism for a particular task. These two activities also correspond with the creation of application commitments in the DOGMA framework for ontology engineering. The Business Semantics Studio (BSS)[6] is a tool suite that supports these two processes. BSS provides mapping functionality to commit existing data sources and applications onto the EIM with Ω-RIDL [37]. Below are two examples of such mappings: one committing a field in a database to a concept in the EIM and another path in an XML-document. These mappings can be used to automatically generate data transformations from one format into another by generating the appropriate queries (SQL, XPath, etc.). The examples are intentionally kept simple for didactic reasons.

```
A) map "DatabaseName.TBLSchool.Street" on
   Street of (/ with) Address of (/ with) School.
B) map "/schools/school/street" on
   Street of (/ with) Address of (/ with) School.
```

The Flemish Public Administration wishes to set up a Linked Data portal for the key entities in their business-ecosystem: researchers, research projects, research organizations, etc. The Linked Data initiative aims at providing interlinked information in a representation suitable for the type of requesting agent: human readable format for users, structured data for software agents. For the latter, two simple technologies are used: URIs to identify things on the Web, and the Resource Description Framework (RDF) for describing things on the Web. To add semantics to these descriptions,

[6] http://www.collibra.com/products/business-semantics-studio

ontologies materialized in RDF(S) or OWL are often used. The selection and commitment phases for this particular goal will thus include the an implementation of relevant parts of the EIM into RDF(S) or OWL. This will be described in the next section.

6.1 Towards a Web of Data: Implementation in other Formalisms

In this section, we briefly describe how (relevant parts) of the EIM is translated into other formalisms. To this end, relevant parts of the EIM need to be translated into formalisms adopted for these particular initiatives. Via MOF, parts of the EIM are also translated into – for instance – UML for the development of applications that need to be developed between stakeholders.

Even though UML is richer than SBVR for capturing some aspects of application design such as operations and components packaging, SBVR has several advantages over UML. The fact types and constraints are easily verbalized and populated (with examples) for validation with domain experts. SBVR makes no use of attributes in its base models. All fact types are represented in terms of objects playing roles. An attribute-free approach has advantages for conceptual analysis, including simplicity, stability, and ease of validation [18]. The UML specification recommends the Object Constraint Language (OCL) for formal expression of business rules, but OCL is too mathematical in nature to be used for validation by nontechnical domain experts. By design, the translation of SBVR into UML via MOF can tackle some of these issues. UML class diagrams' are valuable as the structure of those diagrams is closer to the implementation of a system. With this in mind, SBVR can be used for domain modeling and a UML diagram can be derived for the system's implementation.

Translating SBVR into OWL DL is fairly straightforward [21]. Again, not all transformation from one schema in a language into another language is *lossless*. Lossless means that both schemas are population equivalent. Fact types with arity n where n > 2, for instance, cannot be modeled with OWL DL. As SBVR is grounded in first order logic, it is not decidable whether a statement is provable (i.e., true under all possible interpretation). Decidability is important when one was to do reasoning, e.g., find out whether a class can have any instances or subtype inference. Many description logics are decidable fragments of first order logic, more suitable for such tasks. However, as those description logics are subsets of first order logic, translation from one to the other are not guaranteed to be equivalent.

In a first instance, EWI aimed to publish the FRIS portal data as Linked Data on the Web. In a second instance, they want to validate this data based on the business rules modeled by the community of stakeholders. To achieve the first goal, the ontology resulting from the BSM activities were translated into OWL and this OWL schema was published on the Web. The OWL schema was then used to structure, annotate and publish the information as Linked Data on the Web. This process actually corresponds with the semantic application of BSM; facts are selected to annotate the existing data source to achieve interoperability.

Fig. 9 shows a part of the generated OWL from the concept depicted in the previous figure. In this figure, we see that `Organizational_Unit` is a `Class` and instances of that class can be characterized by keywords (a `Literal`). Furthermore, an Organizational_Unit is composed of instances of Person (again a Class) and through the Organizational_Unit_composed_of_Person property. The inverse role is also specified.

```
<owl:DatatypeProperty
rdf:about="#Organizational_Unit_characterised_by_Keyword">
 <rdfs:label>characterised by Keyword</rdfs:label>
 <rdfs:domain rdf:resource="#Organizational_Unit"/>
 <rdfs:range
  rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
<owl:ObjectProperty
 rdf:about="#Organizational_Unit_composed_of_Person">
 <rdfs:label>composed of Person</rdfs:label>
 <rdfs:domain rdf:resource="#Organizational_Unit"/>
 <rdfs:range rdf:resource="#Person"/>
 <owl:inverseOf
  rdf:resource="#Person_member_of_Organizational_Unit"/>
</owl:ObjectProperty>
```

Fig. 9. Screenshot of the OWL around Project generated by BSG. In this picture, we see that Person is a Class and Persons have roles in an organization.

The contents of the databases to be annotated can be published with off-the-shelf solutions such as D2R Server[7]. D2R Server generates an RDF description containing a mapping for transforming the content of a database into RDF triples. This mapping – also described in RDF – contains a "skeleton" RDF(S) of classes and properties that are based on the database schema. **Fig. 10** below depicts a part of the generated mapping file around the table containing information around projects.

```
@prefix map: <file:///.../OSCB/d2r-server-0.7/map.n3#>.
@prefix vocab: <http://192.168.0.136:5432/vocab/resource/>.
@prefix d2rq: <http://www.wiwiss.fu-
berlin.de/suhl/bizer/D2RQ/0.1#>.
...
map:CFPROJ a d2rq:ClassMap;
 d2rq:dataStorage map:database;
 d2rq:uriPattern "CFPROJ/@@CFPROJ.CFPROJID|urlencode@@";
 d2rq:class vocab:CFPROJ;
 d2rq:classDefinitionLabel "EWI.CFPROJ";
...
```

Fig. 10. Part of the generated mapping file by D2R server, it maps the table CFProj to the generated CFPROJ RDF(S) class. It uses the primary key to generate a unique ID and the class definition label is taken from the table's name.

Even though classes and properties are generated and populated with instances,

7 http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

these RDF triples are not semantic as they stem from one particular information system (its database schema). The RDF(S) skeleton is thus complemented with the generated RDF(S)/OWL classes and properties generated from the BSM ontology. The commitments described in the previous section are used as a guideline to create this alignment. **Fig. 11** below shows the changes (highlighted) made on the generated mapping file with the ontology. The ontology can then be used to access the data.

```
@prefix map: <file:///.../OSCB/d2r-server-0.7/map.n3#>.
@prefix vocab: <http://192.168.0.136:5432/vocab/resource/>.
@prefix d2rq:
http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#>.
@prefix ont: <file:///.../Project.rdf#> .
...
map:CFPROJ a d2rq:ClassMap;
 d2rq:dataStorage map:database;
 d2rq:uriPattern "CFPROJ/@@CFPROJ.CFPROJID|urlencode@@";
 d2rq:class ont:Project;
 d2rq:classDefinitionLabel "Project";
...
```

Fig. 11. Modified mapping file with the ontology exported from BSG. An extra namespace (for the exported ontology) is added and the generated classes and properties are appropriately annotated with that ontology.

To achieve the second goal, the resulting OWL file can be used for one of its popular decision problems: *classification*. Classification or instance checking corresponds with the question: "is a particular instance a member of a given concept?" Whenever we have an instance of one of EWI's key entities (e.g., Project), it can be compared against the business rules around that concept by asking a reasoner whether this particular instance fits this class.

6.2 Full-cycle BSM: Validation and Feedback

Once semantic applications are running, it must be possible to monitor and feed unexpected side effects or failures back, calling for a new iteration of BSM. We call this full-cycle BSM: the scope of the next version of the EIM is fed by the validation of the previous version in IT/IS contexts as well as business contexts. The BSG is the vehicle that serves the reconciliation of the newly scoped concepts.

The BSM cycle is repeated until an acceptable balance of differences and agreements is reached between the stakeholders that meets the requirements of the semantic community. Gradually, closed divergent metadata sources are replaced with metadata sources that follow an open standard, and are kept coherent via BSG.

7 Conclusions

In this paper, we presented the Business Semantics Management (BSM) method for knowledge modeling and ontology engineering. BSM was implemented in the Flemish Public Administration for the building in the context of the Flanders Research

Information Space (FRIS) program. The examples throughout this paper originate from this case.

Even though different formalisms exist for capturing certain parts of the domain, BSM's fact oriented nature, expressed in natural language enables stakeholders to quickly participate in the knowledge modeling processes. Depending on the actual goal of the community, translations or "implementations" of the fact-oriented ontology into other formalisms can be generated. We have shown how the BSM ontology was translated into OWL to publish the FRIS portal data as Linked Data on the Web.

From a high-level perspective, three different kinds of data exchange exist within large organizations: 1) Exchange of knowledge between people; 2) Exchange of understanding between people and information systems; and 3) And exchange of data between disparate information systems.

In this paper and given the requirements of the FRIS case, we focused on the third aspect. All large enterprises, however, face a semantic gap that makes all three of these exchanges extremely inefficient. The BSM method and supporting tools help in capturing the necessary semantics for rendering these exchange processes more efficient by providing a reference point for data governance questions such as: (1) what does my data mean? (2) where and how is my data utilized? (3) who is responsible for my data?

Acknowledgements

The authors would like to thank the Flemish government for the case study. This work was partially funded by the Institute for Promotion of Research and Innovation in the Brussels Capital Region and the European FP7 ACSI project.

8 References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5) (2001) pp. 35–43
2. Bergman, M.: A brief survey of ontology development methodologies. Via http://www.mkbergman.com/906/a-brief-survey-of-ontology-development-methodologies/ (2010)
3. Chen, P.: The enity-relationship model: Toward a unified view of data. In: VLDB: Kerr, D. (ed.), ACM (1975) pp. 173
4. Christiaens, S., De Leenheer, P., de Moor, A., Meersman, R.: Business use case: Ontologising competencies in an interorganisational setting. In: Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience: Hepp, M., De Leenheer, P., de Moor, A., Sure, Y. (eds.). Springer (2008)
5. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A.: Methdologies, tools and languages for building ontologies, where is their meeting point? In Data and Knowledge Engineering, 46(11), pp 41-64. (2003)
6. De Leenheer, P.: On Community-based Ontology Evolution: Foundations for Business Semantics Management. Phd thesis, Vrije Universiteit Brussel (2009)
7. De Leenheer, P.: Ontology elicitation. In: Encyclopedia of Database Systems: Liu, L., Özsu, M.T. (eds.). Springer US (2009) pp. 1966–1972

8. De Leenheer, P., Christiaens, S., Meersman, R.: Business semantics management: A case study for competency-centric HRM. Computers in Industry **61**(8) (2010) pp. 760–775
9. De Leenheer, P., de Moor, A., Meersman, R.: Context dependency management in ontology engineering: a formal approach. Journal of Data Semantics **8** (2007) pp. 26–56 LNCS 4380, Springer.
10. De Leenheer, P., Debruyne, C.: DOGMA-MESS: A tool for fact-oriented collaborative ontology evolution. In: On the Move to Meaningful Internet Systems 2008: ORM (ORM 2008). LNCS, Monterrey, Mexico, Springer (2008)
11. De Leenheer, P., Mens, T.: Ontology evolution: State of the art and future directions. [19] pp. 131–176
12. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Proc. of the 14th Int. Conf. on Conceptual Structures (ICCS 2006). Vol. 4068 of LNCS, Springer (2006) pp. 189–203
13. Gomez-Perez, A., Fernandez-Lopez, M. and Corcho, O.: Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer-Verlag (2003)
14. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies **43** (1993) pp. 907–928
15. Guarino, N.: Formal ontology, conceptual analysis and knowledge representation. International Journal of Human and Computer Studies **43**(5-6) (1995) pp. 625–640
16. Halpin, T.: Metaschemas for ER, ORM and UML data models: A comparison. J. Database Manag. **13**(2) (2002) pp. 20–30
17. Halpin, T.: Comparing metamodels for ER, ORM and UML data models. In: Advanced Topics in Database Research, Vol. 3: Siau, K. (ed.). Idea Group (2004) pp. 23–44
18. Halpin, T.: Information Modeling and Relational Databases. Morgan Kaufmann, San Francisco, CA, USA (2008)
19. Hepp, M., De Leenheer, P., de Moor, A., Sure, Y. (eds.): Ontology Management, Semantic Web, Semantic Web Services, and Business Applications. Vol. 7 of Semantic Web And Beyond Computing for Human Experience. Springer (2008)
20. Hepp, M., Siorpaes, K., Bachlechner, D.: Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management. IEEE Internet Computing **11**(5) (2007) pp. 54–65
21. Hodrob, R., Jarrar, M.: ORM To OWL 2 DL Mapping. In proceedings of the International Conference on Intelligent Semantic Web – Applications and Services. Pages 131-137. ACM (2010).
22. Jarrar, M., Meersman, R.: Ontology Engineering -The DOGMA Approach. Book Chapter in "Advances in Web Semantics I". Chapter 3. Pages 7-34. LNCS 4891, Springer (2008)
23. Jörg, B.; Krast, O.; Jeffery, K.; van Grootel, G. (2009b): CERIF2008XML - 1.0 Data Exchange Format Specification, euroCRIS
24. Meersman, R.: Towards models for practical reasoning about conceptual database design. In: Proc. of the 2nd IFIP 2.6 Working Conference on Database Semantics, “Data and Knowledge” (DS-2). (1986)
25. Meersman, R.: Semantic ontology tools in IS design. In: ISMIS: Ras, Z.W., Skowron, A. (eds.). Vol. 1609 of LNCS., Springer (1999) pp. 30–45
26. OMG: Meta object facility, v2.0. http://omg.org/spec/MOF/2.0/ (2009)
27. OMG: Semantics of business vocabulary and business rules, v1.0. http://omg.org/spec/SBVR/1.0/ (2009)
28. OVUM, INMARK: Value it (support action grant agreement no.: 216710): D3.2 final demand driven mapping report (2010)

29. Petrie, C.: Pragmatic semantic unification. IEEE Internet Computing **9**(5) (2005)
30. Schürr, A., Nagl, M., Zündorf, A. (eds.): Applications of Graph Transformations with Industrial Relevance, Third International Symposium, AGTIVE 2007, Kassel, Germany, October 10-12, 2007, Revised Selected and Invited Papers. In: AGTIVE: Schürr, A., Nagl, M., Zündorf, A. (eds.). Vol. 5088 of Lecture Notes in Computer Science, Springer (2008)
31. Sheth, A., Kashyap, V.: So far (schematically) yet so near (semantically). In: DS-5: Hsiao, D., Neuhold, E., Sacks-Davis, R. (eds.). Vol. A-25 of IFIP Transactions, North-Holland (1992) pp. 283–312
32. Sheth, A., Ranabahu, A.: Semantic modeling for cloud computing, part 2. IEEE Internet Computing (4) (2010) pp. 81–84
33. Simperl, E., Tempich, C., Ontology engineering: a reality check. In OTM Conferences (1): Meersman, R., Tari, Z. (eds.), Volume 4275 of LNCS, pp. 836-854, Springer (2006)
34. Siorpaes, K., Simperl, E.: Human intelligence in the process of semantic content creation. In World Wide Web, 13(1-2): 33-59 (2010)
35. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record Special Issue **31 (4)** (2002) pp. 12–17
36. Spyns, P., Tang, Y., Meersman, R.: An ontology engineering methodology for DOGMA. Applied Ontology **3**(1-2) (2008) pp. 13–39
37. Trog, D., Tang, Y., Meersman, R.: Towards ontological commitments with Ω-RIDL markup language. In: RuleML: Paschke, A., Biletskiy, Y. (eds.). Vol. 4824 of LNCS., Springer (2007) pp. 92–106
38. Van Grootel, G.; Spyns, P.; Christiaens, S.; Jörg, B.: Business Semantics Management Supports Government Innovation Information Portal. In: Meersman, R.; Herrero, P., Dillon, T.; et al. (Eds.); Proceedings of the OTM 2009 Workshops, LNCS 5872, Springer (2009) pp. 757 – 766
39. Wang, T. D., Bijan, P., Hendler, J.: A Survey of the Web Ontology Landscape. The Semantic Web – ISWC 2006, LNCS 4273, Springer (2006) pp. 628
40. Wintraecken, J.: The NIAM Information Analysis Method, Theory and Practice. Kluwer Academic Publishers (1990)
41. Zhao, G.: AKEM: an ontology engineering methodology in ff poirot. FF POIROT Project Deliverable 6.8 (2005)