Using Semantics to Aid Scenario-Based Analysis

Ana Karla Alves de Medeiros[2], Alessio Carenini[1], Irene Celino[1], Emanuele Della Valle[1], Federico M. Facca[1], Michael Oppitz[3], Gernot Zeissler[3], and Stefan Zöller[3]

[1] CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, I
name.surname@cefriel.it
[2] TUE – Technische Universiteit Eindhoven, Postbus 513, 5600 MB, Eindhoven, NL
A.K.Medeiros@tue.nl
[3] IBIS Prof. Thome AG – Mergentheimer Str. 76a, 97082 Wuerzburg, D
surname@ibis-thome.de

Abstract. Analysis of data and processes based on business scenarios is a well known and consolidated technique. Current research efforts are introducing formal knowledge representation techniques to support the selection of relevant metrics, which is one of the crucial tasks in the scenario-based analysis. We believe that semantics may aid not only the selection of the metric, but the whole analysis process. In this paper we introduce our semantically aided analysis approach, that supports all the phases of the analysis process: the selection of metrics relevant to the selected scenario, the configuration of selected metrics, the execution of configured metrics, the evaluation of returned results and, finally, the reporting of results. In particular, to support our approach, we show how it can be applied to the Reverse Business Engineering analysis technique. Nevertheless, the proposed approach can be generalized to be applied to any analysis technique.

1 Introduction

Data model design is a well consolidated discipline, that relies on well defined formalism that allows, starting from a conceptual model to obtain the corresponding logical and physical model with the proper set of translation. Not all the information contained in the conceptual model is translated into the logical model, and not all the knowledge required to analyse data is translated when it is lowered from a conceptual level to its implementation. Ontology engineering has been efficiently applied to solve the gap between conceptual and logical model in the data model design field.

A similar gap also exists in the data analysis domain. The analysis functions are rarely defined using a conceptual model. In most cases they are modeled using human understandable sentences. Then, starting from the data model, analysis functions are implemented by means of proper technologies (e.g. SQL queries, Java code, ...). Thus the gap between the physical model and the corresponding initial specification – which in most of the cases is not formal at all – is even wider than in the previous case. Therefore the conceptual model, which underlies the data to be analysed, is separated from the model of the analysis function.

The approach we present in this paper aims at overcoming also the knowledge gap between the high level model of the analysis functions and their physical model. In particular, our approach relies on Ontology Engineering also to define a common framework for the definition of the conceptual model of the data to be analysed and the related analysis functions.

Scenario-based analysis is a well established technique to extract knowledge from data using a set of predefined scenarios. Each scenario can be configured for the proper domain defining a set of metrics (possibly generic and parametric). Such metrics are defined by domain and analysis experts, representing the information that analysts desire to measure within a particular system, and are evaluated according to the data the system is able to expose. Usually, for a given scenario, a huge number of analytical content (predefined metrics or analysis functions) is available and the analysts need to be properly assisted to select those that are relevant to the analysis scenario and the data domain. Once the metrics are defined, the main steps of a generic analysis process can be identified with:

- *selection*, by the analyst, of the interesting metrics according to the specific analysis scenario
- *configuration* of the analysis metrics by providing proper parameters to restrict the results
- *execution* of the configured analysis metrics
- results *evaluation*, like calculation, data filtering and aggregation
- results *reporting*

Also the problem of selecting the proper post processing (or evaluation) function is not trivial, and analysts often face the problem to identify which evaluation function can be applied to the results returned by analysis metrics.

Usually these analysis metrics are expressed by means of stored procedures inside a database or functions in a programming language, and so users have to know what each of the metrics accomplishes before starting the analysis. The selection is completely dependent on the user's knowledge about kind of analysis, input data and expected output. In some case the selection is guided by a tool, nevertheless, in such case, the link between the model of the metrics and the metrics "implementation" is not formally defined and does not rely on a well defined underlying conceptual model.

The research we describe in this paper shows how metrics can be formally defined using knowledge representation techniques and how they can be linked with the semantic data model at a conceptual level, maintaining the link and all the information expressed at the conceptual level also at the implementation stage. The reason therefore is that the conceptual model and the implementation model are expressed by the same set of semantic descriptions. We will also show how the whole analysis process gains in automation and provides a better support for analysts with the adoption of a proper ontological modeling framework. In particular, we do not only describe by means of ontologies the data model underlying the analysis and the analysis functions, but we also formalize analyst's needs and the analysis process needs into a knowledge base.

The remainder of this paper is structured as follows: Section 2 provides motivation for the use of semantics in analysing data and introduces the basic idea and abstraction used in this work. In Section 3 we propose a concrete scenario to apply our approach, in particular we exemplify it on top of the Reverse Business Engineering analysis technique. Section 4 reports an overview of related researches, with a detailed insight into semantic process analysis. Finally Section 5 presents an assessment of the technique, lessons learned, conclusions and future directions of our research.

2 Semantically-aided analysis

A process is a set of transformations of input elements into products, respecting constraints, requiring resources, meeting a defined mission and corresponding to a specific purpose adapted to a given environment.

If we focus on the scenario-based analysis process, the starting step is the selection of the process that has to be performed in order to fulfill an analysis need. The transformation has then to be fed with input elements that represent its execution constraints, and its results can be used to feed other transformations before being visualized. This workflow traditionally features a massive use of implicit knowledge in all of its steps, and this knowledge has to be provided by the user and manually exploited to make the workflow work correctly. We will now explain how adding semantics can improve these steps, and so the overall analysis process, by using explicit, formal and shared knowledge.

Focusing on the mission and the purpose of a given analysis function, in a traditional scenario the user must have implicit knowledge in order to select the right function for analysing a certain portion of the domain scenario. This knowledge derives from an extensive usage of the system and from studying the documentation, about what task is performed by each of the functions provided by the system and about what is the context in which each function can be performed. The first expected benefit resulting from adding semantics to the analysis functions is the ability to explicitly and automatically link back a function to its execution context. With an explicit link it is possible to help the user in selecting functions to be executed, given a set of selected concepts and instance data that represents the analysis domain. In the first case, by enriching the description of a function with the concepts it refers to, it is possible to select functions by expressing the concepts that will be relevant in the analysis context. In the second case, given a set of instance data, it is possible to suggest user which are the functions that can be applied and what kind of context they apply to.

Once the process (or function) has been successfully selected, it still just represents a relation between a set of input elements and the expected resulting products, so in order to be executed, there is still the requirement of feeding it with the right input data. Semantics can help in automating this analysis step by suggesting existing instance data that can be used to fill the parameters, or help in the creation of the needed values by linking to concepts expressed in the data model.

Next in the process of performing an analysis function is the execution of the function itself. In a traditional scenario, this is accomplished using a programming language, thus embedding the logic into a particular implementation choice; this solution implies that, every time a new function has to be executed, a programmer must code the corresponding business logic. Semantics can turn the logic of a function explicit by expressing it in a formal language, completely decoupling the logical part from the code that performs the execution. Moreover, the results of the execution of the analysis function expressed by using semantics are still instances of the concepts defined in the ontology that formalizes the data model. As an analysis process can be composed of sub-processes that perform smaller tasks, the output of a function can be used to feed another select-configure-execute loop, thus enabling arbitrarily complex processes without having to use a programming language to fill the analysis needs.

The last step in the analysis process is the visualization of the results. In a traditional scenario, the user has to know which is the best way of visualizing the results according to the kind of data and the analysis context. The same modeling principles used on the analysis functions to ease selection can also be used to suggest the best visualization strategy for a given context. Moreover, the linkage with concepts expressed in external ontologies eases the task of transforming the results into data directly suitable by existing visualization systems.

We will now introduce and discuss our proposed solution that aims at offering a sound ontological framework to enable data extraction (§2.1 and §2.2) and to support evaluation and reporting of the analysis results (§2.3).

2.1 The analysis ontology

As introduced in Section 1, our scenario-based analysis approach relies on the usage of semantic technologies. The overall idea is not bounded to any peculiar semantic framework; the only requirements is the support for definition not only of data models but also of rules/axioms. Although ontologies provide the basis for some forms of reasoning, ontologies by themselves do not support the range of knowledge-based services that are required to fulfill the complexity of the common analysis functions. In this work we use the WSML modeling language, since it is the language used in the SUPER research project we are taking part in. However, we can think of adopting any other ontological language (like OWL [1]) with support for rules (e.g., SWRL [2]).

In the rest of the section we describe how adopting semantic technologies enables supporting and providing better automation for the whole analysis process. The key factor to enable the whole analysis process using semantic technologies is to provide a conceptual model that formalizes all its relevant aspects and covers all its fundamental steps. In particular, we devise an ontological model that comprises the two main aspects of the analysis process: the analysis metric to be

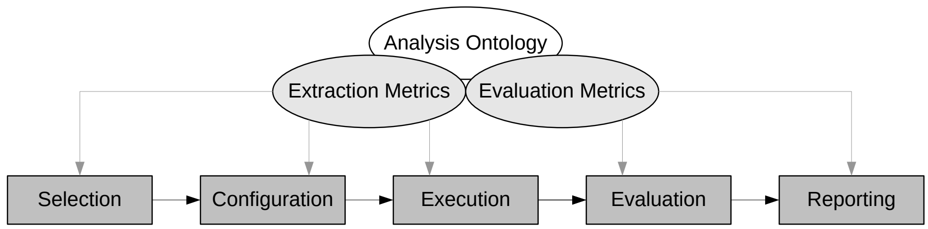

Fig. 1. The proposed modeling stack and its relations with the scenario-based analysis process

applied on the data to generate intermediate results and evaluation templates to be applied on the intermediate results to generated final reports.

The modeling abstractions used to model the analysis metrics should support analysts for their categorization and selection, their configuration and their execution. While models of the evaluation templates should support the filtering of templates according to the used analysis metrics, the aggregation/filtering of results and their presentation. Figure 1 gives an overview of the modeling stack at the base of our approach.

2.2 Metrics for data extraction

To model analysis metrics, we have to keep into account the three fundamental goals we want to achieve: (i) an easy way to allow categorization and thus selection of metrics according to analyst needs and the considered scenario; (ii) enable assisted configuration of the required parameters for selected metrics, and, if needed, the assisted creation of complex parameters values; and, finally, (iii) a specification of how to compute the metrics, based on (i) and (ii) and on the model of the data under analysis.

Given such objectives, the natural modeling abstraction that fits the categorization of metrics, is the use of an ontology. Thus, we formalize metrics as ontological concepts. In more details, we define a generic *Metric* concept that is described by a set of attributes. The attributes are defined over an ontology, thus enabling a fine grained categorization of metrics; e.g., an attribute may be linked to a concept *Scenario*, to allow metrics categorization according to an ontology of scenarios. Then, each peculiar metric is defined as sub concept of the generic *Metric* concept and its attributes are defined over sub concepts of the original attribute type; e.g., an attribute may be linked to a particular analysis scenario defined by a concept belonging to a branch of the same ontology used at the previous step. The choice of modeling metrics not as instances of the metric concept, is fundamental to allow the definition of sub-metrics using the inheritance mechanism provided by ontologies. This way, we can define a well formalized ontology of metrics, where also the attribute ranges of the metric concept belongs to ontologies for an efficient and powerful metrics categorization, thus enabling the analysts to easily select the most suitable metrics among a huge variety of them, by simply specifying a set of possible values for the range of their attributes (i.e. filtering them by the attributes values). At this stage of the modeling process there is still no need of any connection with the actual model of the data to be analysed.

While the modeling abstraction just introduced is perfect for the scope of classifying analysis metrics, it is not suitable for the specification of the metrics themselves and their parameters. This aspect is crucial to enable the assisted configuration of metrics. For this purpose, we select relations with arity n as abstraction. Relations are used to define the metric and its possible parameters, i.e., each relation is a metric and its terms represent the parameters for the metric. Terms are constrained to a type range, thus specifying the expected input types for a given metric. The analyst configures the metrics by bounding the terms in the relation definition with the actual instances in the range of the terms type. Configured parameters (i.e. bounded terms) act as input parameters, while free parameters (i.e. not bounded terms) act as output parameters. Parameters may be defined in terms of the model of the data to be analysed (e.g. a given concept in the analysed data), or may be generic and not grounded to the actual data model (e.g. temporal constraints). Such modeling abstraction solves the problem of parameter definition and their expected order. Nevertheless, it decouples the parameter definition problem from the conceptualization used to the define the metric ontology. To preserve such important link in the model, we use a relation to link the metric concept with the definition of its relation. This is not always required, metrics in the upper level of the ontology, may not have any associated relation, since they may be only abstract containers for a set of concrete metrics.

The relation itself specifies how to configure the metric in order to enable its evaluation. It does not contain any information about the way the relation, and hence the metric evaluation results, are actually computed. To formalize such fundamental requirements that enable the metric execution, we use axioms. Axioms (or rules), are logical expression that define how a relation is actually computed, enabling its evaluation with the support of a reasoning engine. In particular, logical expressions, given the parameters defined in the relation and the model of the analysed data, define how the terms of the relation instances, which represent the results of the metric execution, are correlated.

2.3 Post processing metrics

FMF: better title?! this section is just a quick idea. who's is taking care of reviewing it?

The post processing of the analysis results includes calculation, data filtering/aggregation and data visualization. Post processing functions can be viewed as the dual of the analysis metrics and are formalized in a similar way. In particular, we define a generic *Function* concept that is described by a set of attributes. The attributes are defined over an ontology, thus enabling a fine grained categorization of functions. Among the attributes, it is possible to include the type of post processing function, and the analysis metrics to which the function may be applied. Then, each peculiar function is defined as sub concept of the generic *Function* concept and its attributes are defined over sub concepts of the original attribute type. In this way, we can define a well formalized ontology of post processing functions, that can be automatically selected according to the analysis metric used in the previous steps of the analysis process.

The *Function* concept ontology enables only the selection and filtering of existing functions, as it does not describe the behaviour of such functions. The modeling support provided by semantic languages to describe such kind of function is still incomplete. The enactment post processing of the analysis results may require the use of aggregate functions (like count, sum, ...) that while are well defined and support in the relational world, are not still included in any semantic query or rule language. Current research efforts within the European integrated project SUPER[1] are leading to the design of such language, thus enabling the formal definition of post processing function over analysis results [4]. We adopt this extension of the WSML logical expression to define our post processing functions.

Finally, the evaluation templates, that specify how the results of the post processing step are visualized or reported to analysts, are defined as an ontology of presentation concepts. Such ontology defines the possible types of reporting templates and their configuration parameters. Then each peculiar template may be linked to its implementation (e.g. a proper XSLT stylesheet), that given a configured evaluation template, uses input data to generate the visual rendering for the analyst. The implementation of a evaluation template may be intended in a broader way and include actions like triggering alarms or sending emails.

FMF: memories from A.K. speech

3 Use case scenario: semantic reverse business engineering

Currently most companies use information systems to support the execution of their business processes. Examples of such information systems are ERP, CRM or Workflow Management Systems. These information systems usually generate events while executing business processes [**?**] and these events can be recorded in so-called event logs. The competitive world we live in requires companies to adapt their processes in a faster pace. Therefore, continuous and insightful feedback on how business processes are actually being executed becomes essential. Additionally, laws like the Sarbanes-Oxley Act force companies to show their compliance to standards. In short, there is a need for good analysis tools that can provide feedback information about how business process are actually being executed based on the observed (or registered) behavior in event logs. Scenario-based analysis is a common technique to do so.

3.1 Reverse Business Engineering

The basic methods behind RBE were developed at the University of Wuerzburg, Germany [5], applied to the SAP R/3 system and converted into the tool Reverse Business Engineer by IBIS Prof Thome AG in collaboration with SAP

[1] `http://ip-super.org/`

AG. The fundamental idea of RBE is the scenario-based analysis of business processes and configuration of application systems (e.g. ERP or CRM) in an automated process [6]. RBE supports various analysis scenarios, like as-is analysis or user and role analysis. Each of them describes customer needs and focuses on different aspects within the information system. A scenario is constituted by a multiplicity of predefined business questions, which shall provide specific information with respect to transaction data, master data, organisational structures, and configuration settings [7].

In the process of an RBE analysis the customer chooses one or several analysis scenarios according to his needs. Based on the scenarios the relevant business questions are selected and composed to an RBE extractor. The results from the extractor are then imported into the RBE tool for evaluation purposes. Various analytical methods are used to evaluate the extracted data, e.g. to determine average cycle times or calculate KPIs. Finally the customer is provided with the outputs in form of reports.

Reverse Business Engineering enables the analysis and improvement of existing business processes and system settings. As explained before, it gives great benefits by supporting various analysis scenarios according to user needs. However, RBE has some limitations regarding the degree of automation and the reuse of RBE contents. The key elements in the RBE analysis process are the business questions, which are used to collect details about the current implementation of a process and its usage. So far RBE is only applied to SAP systems, thus every business question contains proprietary patterns to query the SAP database. To use RBE for analysing other information systems the patterns of the business questions have to be adapted manually because of the peculiarity of each single system and the absence of a middle layer that hides these differences to the system. Hence modelling new business questions causes a lot of manual work and requires a substantiated knowledge about the data structure of the respective information system. Currently the selection of business questions is performed solely according to the chosen analysis scenario. Moreover, business questions are assigned to a standard process model for the purpose of evaluation, so if a company has individual processes that should be evaluated, it causes a lot of manual work to assign the business questions to the corresponding elements in those individual processes.

3.2 sRBE Analysis Process

Scenario-based analysis describes customer needs and focuses on different aspects within the information system. A scenario is constituted by a multiplicity of predefined business questions, which shall provide specific information with respect to transaction data, master data, organisational structures, and configuration settings. A business question consists of a generic question and system-specific patterns. These patterns extract information from event logs. From a conceptual point of view, business questions can be seen as *metrics* that are defined by analysts. These metrics represent the information to be measured within a particular system and are evaluated according to the data the system is able to

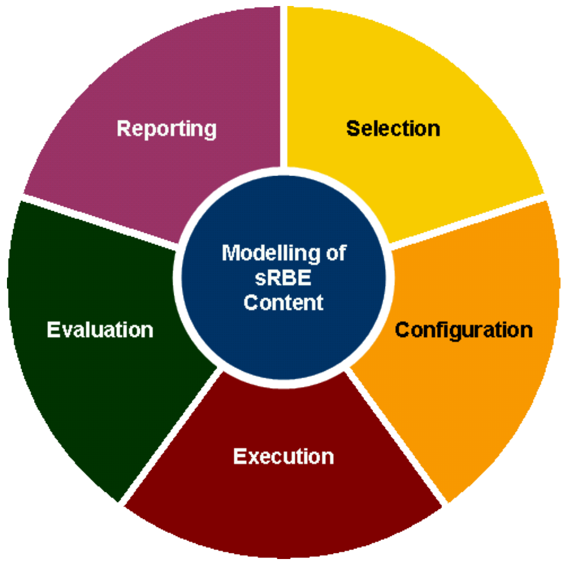

Fig. 2. sRBE analysis process

expose. A given scenario typically contains a huge number of metrics and the analysts need to be properly assisted while selecting relevant metrics to a given analysis scenario and data domain. In a nutshell, once the metrics are defined, the main steps of a generic analysis process are exactly those that have been described in Section 2 and depicted in Figure 2.

The research activities of Semantic Reverse Business Engineering (sRBE) aim at introducing semantic technologies in the RBE process to augment its degree of automation and its flexibility. The goal is to provide a model to describe the sRBE content (i.e. business questions and business metrics) at an abstract level so that they can be defined and categorised regardless of the underlying technology in the adopted system.

At the beginning of an sRBE analysis the customer defines his goals. Generally companies have concrete problems they want to resolve with sRBE, for example the sales manager is interested in all business exceptions that occurred in sales order processing in order to avoid those costly deviations from the standard sales processes. According to this goal the relevant business questions are selected. This selection is done automatically by choosing the corresponding concepts of the ontologies, e.g. analysis scenario and business area.

The second phase in the analysis process is the configuration of the selected business questions. The business analyst can specify various parameters. For example by entering an analysis period the analysis results can be restricted in a way that only those instances are considered whose timestamp is between the start date and the end date. Subsequently the selected and configured business questions are executed to retrieve the analysis results. Generally a reasoner is used to query the repository that contains the instances.

The evaluation phase deals with operating with the results of the executed semantic business questions. Operating means to compute business metrics, generate statistical information and benchmark the relevant processes. Finally the evaluated values have to be presented to the business analyst. This can be done by using spreadsheets and charts or by displaying the results in the context of the executed process model.

3.3 Applying semantics to Business Questions

The semantic annotation of the RBE content by linking to concepts of ontologies allows on the one hand to map generic questions to domain-specific ontologies and hence to consider specific issues and terminologies of the selected domain. On the other hand the semantic business questions and metrics can be assigned to any customer-specific process model which is also described semantically and thus the individual processes of a company can be easily evaluated. The main concepts to classify the business questions are:

- Question Type: Each business question belongs to one of the question types "List", "Count", or "Sum", which determine the extraction and presentation of their answer.
- Activity: Every business question refers to an executed activity. For example the business question "How many sales offers were approved?" refers to the activity "approve sales offer".
- Analysis Scenario: This concept classifies business questions according to the analysis scenarios they support, e.g. as-is analysis or exception analysis.
- Analysis Period: By assigning this concept to business questions the result values can be constrained by a time slice.

Analysing these concepts, we can see that the first three items allow the actual classification, while the last can be considered an execution parameter. According to this distinction, due to having identified our metric in the business question, we have applied the method described in Section 2.2 and enabled using these classification concepts inside the concept *Business Question*, as reported in Listing 1.1.

In Listing 1.2 we can see how an example business question, namely "Which sales orders were processed?", have been modeled in the sRBE ontology. Since it is classified as a part of an as-is analysis scenario, belonging to an order processing business function and to a transaction data category, the corresponding concepts in the ontology have been linked to the business question, thus allowing selection. The relation "WhichSalesOrdersWereProcessed" instead models the execution parameters, such as the analysis period and the function of the successfully executed event.

As explained in Section 2.2, according to our modeling abstraction, the execution of each business question is performed by an axiom which is executed by

```
concept BusinessQuestion
      hasDataCategory impliesType DataCategory
      hasScenario impliesType AnalysisScenario
      belongsToBusinessFunction impliesType UPO#BusinessFunction

concept DataCategory

concept ConfigurationData subConceptOf DataCategory

concept MasterData subConceptOf DataCategory

concept OrganisationData subConceptOf DataCategory

concept TransactionData subConceptOf DataCategory

concept AnalysisScenario

concept AnalysisOfExceptions subConceptOf AnalysisScenario

concept AsIsAnalysis subConceptOf AnalysisScenario

concept HarmonisationAndStandardisation subConceptOf AnalysisScenario
```

Listing 1.1. The BusinessQuestion concept

```
concept WhichSalesOrdersWereProcessed_C subConceptOf RBEO#BusinessQuestion
      nonFunctionalProperties
              dc#description hasValue "Which sales orders were processed?"
      endNonFunctionalProperties
      hasScenario ofType RBEO#AsIsAnalysis
      hasQuestionType ofType RBEO#List
      hasDataCategory ofType RBEO#TransactionData
      belongsToBusinessFunction ofType BFO#OrderProcessing

relation WhichSalesOrdersWereProcessed( ofType RBEO#SuccessfulExecutionEvent, ofType
        BFO#OrderProcessing, ofType _"http://ip−super.org/ontologies/BRO/20070801#
        MarketingAndSalesRole", ofType RBEO#AnalysisPeriod)
      nonFunctionalProperties
              dc#description hasValue "relation related to business question Which sales orders were
                      processed?"
      endNonFunctionalProperties
```

Listing 1.2. Example of a semantic business question: selection-oriented concept and result-oriented relation

a reasoner, as reported in Listing 1.3, while querying the results is accomplished by using the configured relation.

The thus obtained results are the basis for further calculations, therefore metrics have to be defined and described semantically. A classification criteria is the concept Dimension with its specifications Time, Cost and Quality. An example of a metrics is "cancellation rate of sales orders", which is calculated by dividing the results of the business questions "How many sales orders were cancelled?" by "How many sales orders were created?" [8].

The SUPER project is still ongoing, and only first efforts have been invested in the implementation of a sRBE framework. In particular we have not yet formalized the post processing metrics, since their requirements have not been

```
axiom WhichSalesOrdersWereProcessed_X
    definedBy
        ?pe[hasCreationTimeStamp hasValue ?date, occurredDuringProcessExecution
            hasValue ?proc, GeneratedBy hasValue ?actor] memberOf EVO#
            SuccessfulExecutionEvent
  and ?proc memberOf BFO#OrderProcessing
  and ?role[playedBy hasValue ?actor] memberOf BRO#MarketingAndSalesRole
  and ?period[hasStartValue hasValue ?start, hasEndValue hasValue ?end] memberOf RBEO#
      AnalysisPeriod
  and ?date > ?start
  and ?date < ?end
implies
WhichSalesOrdersWereProcessed(?pe,?proc,?role,?period).
```

Listing 1.3. Example of an execution axiom

yet collected and the WSML-Flight query extension was just released. Hence here we do not present a concrete example of the concepts introduced in Section 2.3.

3.4 Implementation

To test our approach we developed within the SUPER IP project a sRBE engine prototype and integrated it with the SUPER architecture. Figure 3 illustrates the overall architecture of the sRBE engine including its connection with the SUPER architecture. The sRBE engine itself is composed by three layers, the reasoning engine that provides support for querying and inferring over semantic data; the business logic that includes a set of predefined functions to support the analysis workflow as introduced in Section 2 and provides access to the reasoner; a Graphical User Interface that, using the functionalities provided by the business logic, allows the users to performs the sRBE process.

The sRBE engine includes a Business Question Repository, where the modeled questions and a set of other ontologies that are used to define the Business Question taxonomy are stored.

Analysis data are imported from the Semantic Service Bus provided by SUPER architecture, which includes the access to: the Business Process Library, that contains all the model of the deployed and logged processes; the Execution History, that contains all the log of the processes executions; and finally, any other repository (also non semantic) that contains data needed to analyse business processes (e.g. ERP data).

Data exposed by the Semantic Service Bus are seamless integrated through the reasoner engine, that allows in this way to perform query over distributed data.

4 Related Work

The idea of using semantics to perform process analysis is not new. In 2002, Casati et al. [9] introduced the *HPPM intelligent Process Data Warehouse (PDD)*,

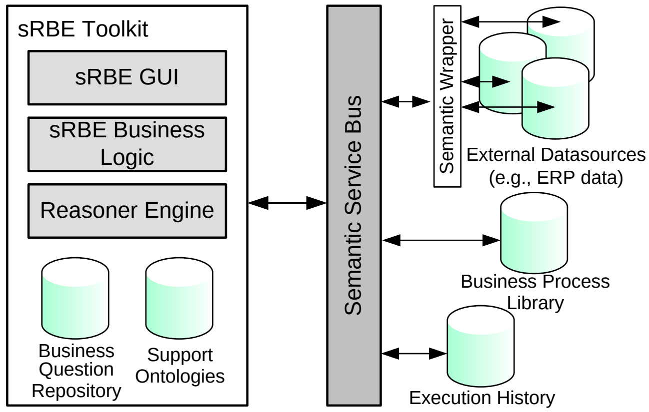

Fig. 3. SUPER architecture with regards to the Analysis scenario

in which taxonomies are used to add semantics to process execution data and, therefore, support more business-like analysis for the provided reports. The work in [10] is a follow-up of the work in [9]. It presents a complete architecture for the analysis, prediction, monitoring, control and optimization of process executions in Business Process Management Systems (BPMSs). This set of tools suite is called *Business Process Intelligence (BPI)*. Some differences of these two approaches to ours are that (i) taxonomies are used to capture the semantic aspects (in our case, ontologies are used), and (ii) these taxonomies are flat (i.e., no subsumption relations between concepts are supported). Hepp et al. [11] proposes merging Semantic Web, Semantic Web Services (SWS), and Business Process Management (BPM) techniques to build Semantic BPMSs. This visionary paper pinpoints the role of ontologies (and reasoners) while executing semantic analysis. However, the authors do not present any concrete implementations for their ideas. The works by Sell et al. [12] and O'Riain et al. [13] are related to ours because the authors (i) also use ontologies to provide for the semantic analysis of systems and (ii) have developed concrete tools to support such analysis. The main differences are the kind of supported analysis. The work in [12] can be seen as the extension of OLAP tools with semantics. The work in [13] shows how to use semantics to enhance the business analysis function of detecting the core business of companies. This analysis is based on the so-called Q10 forms. Alves de Medeiros et al. [14] contains an outlook on the use of semantics to improve the analysis provided by existing process mining and monitoring techniques. The core idea is to annotate event logs and models with ontologies in order to support analysis at the concept level. In fact, more from an event log point of view, Pedrinaci et al. [15] have defined the Event Ontology and the Process Mining Ontology. These two ontologies can be used to give semantics to the event types and the process instances in logs. For instance, it is possible to say that a process instance was successfully executed. Although the semantic extensions in [14,15] are necessary to realize our approach, the authors do not discuss how to use ontologies to facilitate an analysis based on scenarios. In other words, the focus is more on the actual answering of the questions rather than on also using semantics to classify and retrieve these questions. Our paper is the first one to explore the use of semantics for a *scenario-based analysis* and to show a concrete implementation in this direction, following the initial ideas presented in [rif al paper che abbiamo pubblicato al workshop dell'ESWC scorsa!].

5 Conclusions

In this paper we presented our approach towards the adoption of semantics to support scenario-based analysis. According to our knowledge this is the first attempt made in such direction and opens a new whole research about the use of semantic technologies to model not only data but also functionalities and their execution logic.

In particular through the paper, we evidenced the specific contribution that semantics may give at each step of the process. Thus we proposed a generic semantic metamodel suitable for any scenario-based analysis technique leveraging on the expressive power offered by semantic languages, that allows not only to describe data models but also functionalities and their execution logic, lifting it from the traditional implementation level.

To validate the proposed approach we showed how we applied it to the Reverse Business Engineering technique, one of the most important scenario-based analysis techniques in the context of business process analysis.

We also discussed in detail the benefits that the introduction of semantics brings to RBE, such as a greater level of automation, generation of system-independent analysis, better reusability of sRBE content and better administration of sRBE content.

Moreover, as the explained analysis architecture is database independent, wider spread business questions can also be used as extractors to get and semantically annotate service based transaction data, e.g. for BI purposes (extractors for Data Warehouses)

In the next future we plan to better refine our approach, particularly for the part of post processing and reports generation that is not yet well formalized as the previous steps. We also plan to test our approach on different use case scenarios from the one proposed such as Process Mining or ...

FMF: ideas?

References

1. Michael K. Smith, Chris Welty, and Deborah L. McGuinness. Owl web ontology language guide. W3c recommendation 10 february 2004.
2. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. W3c member submission 21 may 2004.

3. Jos de Bruijn, Holger Lausen, Axel Polleres, and Dieter Fensel. The web service modeling language wsml: An overview. In York Sure and John Domingue, editors, *ESWC*, volume 4011 of *Lecture Notes in Computer Science*, pages 590–604. Springer, 2006.
4. Stijn Heymans, Cristina Feier, Jos de Bruijn, Stefan Zller, and Emilia Cimpian. Process ontology query language. Technical report, SUPER Integrated Project, 2007.
5. H. Wenzel. Reverse business engineering: Ableitung von betriebswirtschaftlichen modellen aus produktiven softwarebibliotheken. *Published dissertation at the chair of business administration, University Wuerzburg*, 1999.
6. A. Hufgard and W. Walz. The road behind, Continuous business improvement with Reverse Business Engineering. 2003.
7. R. Thome and A. Hufgard. Avoiding bad choices. 2005.
8. C. Ebert. *Best Practices In Software Measurement: How To Use Metrics To Improve Project And Process Performance.* Springer, 2004.
9. F. Casati and M.-C. Shan. Semantic Analysis of Business Process Executions. In *EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology*, pages 287–296, London, UK, 2002. Springer-Verlag.
10. D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.-C. Shan. Business Process Intelligence. *Computers in Industry*, 53(3):321–343, 2004.
11. M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel. Semantic Business Process Management: a Vision Towards Using Semantic Web services for Business Process Management. In *IEEE International Conference on e-Business Engineering (ICEBE 2005)*, pages 535 – 540, 2005.
12. D. Sell, L. Cabral, E. Motta, J. Domingue, and R. Pacheco. Adding Semantics to Business Intelligence. In *DEXA Workshops*, pages 543–547, 2005.
13. S. O'Riain and P. Spyns. Enhancing the Business Analysis Function with Semantics. In *OTM Conferences (1)*, pages 818–835, 2006.
14. A.K. Alves de Medeiros, C. Pedrinaci, W.M.P. van der Aalst, J. Domingue, M. Song, A. Rozinat, B. Norton, and L. Cabral. An Outlook on Semantic Business Process Mining and Monitoring. In *OTM Workshops (2)*, pages 1244–1255, 2007.
15. C. Pedrinaci and J. Domingue. Towards an Ontology for Process Monitoring and Mining. In *Proceedings of SBPM 2007 Semantic Business Process and Product Lifecycle Management in conjunction with the 3rd European Semantic Web Conference (ESWC 2007)*, Innsbruck, Austria, June 2007.