Using Semantics to Aid Scenario-Based Analysis

Ana Karla Alves de Medeiros[1], Alessio Carenini[2], Irene Celino[2], Emanuele Della Valle[2], Federico M. Facca[2], Michael Oppitz[3], Carlos Pedrinaci[4], Gernot Zeissler[3], and Stefan Zöller[3]

[1] TUE – Technische Universiteit Eindhoven, Postbus 513, 5600 MB, Eindhoven, NL
A.K.Medeiros@tue.nl
[2] CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, I
name.surname@cefriel.it
[3] IBIS Prof. Thome AG – Mergentheimer Str. 76a, 97082 Wuerzburg, D
surname@ibis-thome.de
[4] Knowledge Media Institute, The Open University – Milton Keynes, MK7 6AA, UK
c.pedrinaci@open.ac.uk

Abstract. Scenario-based analysis describes customer needs and focuses on different aspects of information systems. A scenario typically has several metrics which compute specific information about transaction data, organizational structures and configuration settings. The selection and configuration of metrics is not a trivial task and normally cannot be reused over different information systems. Therefore, this paper shows how semantics can aid in this process. In fact, the proposed semantically aided analysis approach supports the five phases of the scenario-based analysis process: (i) *selection* of metrics relevant to a given scenario, (ii) their *configuration* and (iii) *execution*, (iv) *evaluation* of returned results and (v) *reporting* of results. Our approach is illustrated by applying it to Reverse Business Engineering, a method for scenario-based analysis commonly used by commercial ERP systems. However, the proposed approach is general enough to also be applied to other analysis techniques.

1 Introduction

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 store and manage transactional data. In workflow-oriented systems events which are generated during the execution of business processes can be recorded in so-called event logs [1]. 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.

Scenario-based analysis describes customer needs and focuses on different aspects within an information system. An analysis scenario is constituted by a multiplicity of predefined metrics, which will provide specific information with respect to transaction data, master data, organizational structures, and configuration settings. These metrics extract information from data sources, for example event logs. An example of a tool using scenario based analysis is the Reverse Business Engineering tool [2]. Metrics are defined by business analysts and represent the information to be measured within a particular system and are evaluated according to the data the system is able to expose. A given scenario typically contains a huge number of metrics and analysts usually need to be properly assisted while selecting relevant ones for a given analysis scenario and data domain. In a nutshell, once metrics have been defined, the main steps of a generic scenario-based analysis process are as follows:

1. Selection, by an analyst, of the interesting metrics according to specific analysis scenario;
2. Configuration of selected analysis metrics by providing proper parameters to restrict the results;
3. Execution of configured analysis metrics;
4. Evaluation of results, like data filtering and aggregation;
5. Reporting of results.

This paper shows *a framework where ontologies are used to facilitate scenario-based analysis*. The idea of using semantics to improve the analysis of business processes is not new [3,4,5,6,7] but none of the existing works have focused on using semantics to support the analysis from the selection of the metrics to its execution and reporting. In fact, most of the existing work uses ontologies to enhance the execution and re-use of metrics [5]. Therefore, this paper is the first one to show how to use ontologies to provide for semantic scenario-based analysis. This approach is illustrated by showing our first prototype to perform semantic RBE. This prototype is being implemented within the European project SUPER. As stated in [8], SUPER "aims at providing a semantic-based and context-aware framework, based on Semantic Web Services technology that acquires, organizes, shares and uses the knowledge embedded in business processes within existing IT systems and software, and within employees' heads, in order to make companies more adaptive". This semantic framework will also support the semantic analysis of business processes.

The remainder of this paper is organized as follows. Section 2 motivates the use of semantics in analyzing data and introduces the approach defined in this work. Section 3 proposes a concrete scenario to apply this approach to Reverse Business Engineering analysis technique. Section 4 contains an overview of related work in the field of semantic process analysis, and Section 5 has the conclusions and future directions of our research.

[5] See Section 4 for more details.

2 Semantically-Aided Analysis

Scenario-based analysis is characterized by a complex methodology where each step has its own peculiarities. Figure 1 gives an overview of the modelling stack and the analysis phases at the base of our approach.

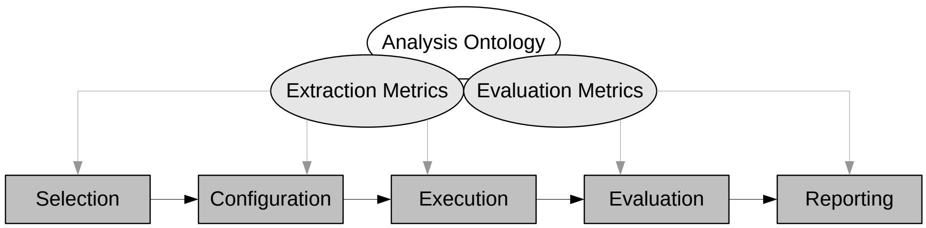

Fig. 1. The proposed modeling stack and its relations with the scenario-based analysis process.

The first step in the methodology is the selection of the metrics that have to be evaluated in order to fulfill a certain analysis need. Then, selected metrics have to be fed with input elements that represent its execution constraints; data returned from the execution of analysis metrics are again processed by evaluation metrics that perform further transformations to clean and filter results. Finally results returned by evaluation metrics are transformed into the correct reporting template. This procedure traditionally features a massive use of implicit knowledge in all of its steps, and this knowledge has to be provided by the user. We will now explain how adding semantics can improve these steps, and so the overall analysis process, by using explicit, formal and shared knowledge.

In order to select the proper analysis metrics for a given domain scenario, analysts have to exploit any possible knowledge (usually implicit) linking the metrics with the analysis scenario. This knowledge, in current systems, derives from an extensive usage of the system and from studying the documentation that specifies which task is performed by each of the metrics provided by the system and in which context each metrics can be performed. Formally defining analysis metrics using semantic technologies enables to link explicitly and automatically metrics to their execution context. Explicit links help analysts in the **selection of metrics** to be executed according to a set of constraints posed by the data domain and analysis scenario. Semantics can support this selection in two ways: firstly, the analyst may restrict the set of analysis metrics he may apply on the data by *selecting a set of concepts* he considers relevant for the analysis scenario; secondly, given a set of instance data, it is possible to trace back which metrics may be applied to such data, and the analysis scenario to which the obtained metrics belong.

Once a metric has been successfully selected, it still only represents a relation between a set of input elements and the expected resulting products. To be executed, such a metric still needs to be properly configured with the right input data. Semantics may provide a great support to automate this **configuration step**: it can filter data present in the knowledge base to identify possible parameter instances; or it may support the analyst in the creation of the correct instance leveraging on the formalized data model of the metric inputs.

Then, after configuration, an analysis metric is ready to be executed. Currently, the logic of a metric is described using a programming language, thus embedding the logic into a particular implementation choice; therefore, every time a new metrics has to be executed, a programmer must code the corresponding business logic. Semantics can be adopted to explicitly express the logic of an analysis metric in a formal language, thus completely decoupling the logical part from the code that performs the **execution**. Moreover, in this way, the results of a metrics's execution are directly expressed using the same ontological model adopted to describe the input data, thus removing any need for grounding mechanism, to lift results from the code level to the data model level. In this way also the definition of **post processing metrics** is easier: the output of a metrics can be used to feed another select-configure-execute loop. In particular the selection of a possible following metrics can directly selected according to the output of the preceding one.

The final step in the analysis process is **reporting** the results: the user, according to the kind of data and the analysis context, selects the best way of visualizing the results. The same modeling principles used on the analysis metrics to ease selection can also be used to suggest the best reporting strategy for a given context. Moreover, the linkage to concepts expressed in external ontologies eases the task of transforming the results into data directly suitable for existing reporting systems.

The remainder of this section introduces and discusses our approach in more details.

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 requirement 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 metrics. For example, the model can be defined using OWL [9] and SWRL [10], a rule specification language for OWL, or WSML [11]. In this work we use the WSMO framework and its modelling language WSML, the choice is motivated by the fact that such framework natively supports the definition of axioms or rules and, besides, it is adopted in the SUPER project. The Web Service Modelling Language (WSML), offers a set of language variants for describing the elements of the Web Service Modelling Ontology (WSMO) that enable modelers to balance between expressiveness and tractability according to different knowledge representation paradigms. The conceptual syntax of WSML adopts a frame-like style. The information about classes, attributes, relations and their parameters, instances and their attribute values are specified in one large syntactic construct, instead of being divided in a number of atomic chunks.

In the rest of the section we describe how adopting semantic technologies enables the support and provision of 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 of its relevant aspects and covers all of its fundamental steps. In particular, we devise an ontological model that comprises the two main aspects of the analysis process: *the analysis metrics to be applied on the data to generate intermediate results and evaluation templates to be applied on the intermediate results to generated final reports.*

The abstractions used to model the analysis metrics should support analysts in their selection, configuration and execution, while models of the evaluation templates should support the filtering of templates according to the chosen analysis scenario, the aggregation/filtering of results and their presentation.

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 a particular scenario; (ii) support for assisted configuration of the required parameters for selected metrics, enabling, 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 a taxonomy. 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 particular 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 for not modeling metrics as instances of the metric concept is fundamental to allow the definition of sub-metrics using the inheritance mechanism provided by ontologies. In this way it is possible to define a well formalized ontology of metrics, where also the attribute ranges of metric concept belong to ontologies enabling efficient and powerful metrics categorization. This enables 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.

The modelling abstraction just introduced, while being suitable for the scope of classifying analysis metrics, 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 scope we select as abstraction n-ary relations. **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 coupling the terms in the relation definition with the actual instances in the range of the terms type. Configured parameters (i.e. bound terms) act as input parameters, while free parameters (i.e. unbound 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.

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 thanks to 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

The post processing of the analysis results includes calculation, data filtering/aggregation and data reporting. Post processing metrics 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 metrics. Among the attributes, it is possible to include the type of post processing metrics, and the analysis metrics to which the metrics may be applied. Then, each peculiar metrics 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 metrics, 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 metrics, as it does not describe the behaviour of such metrics. The modeling support provided by semantic languages to describe such kind of metrics is still incomplete. The enactment post processing of the analysis results may require the use of aggregated metrics (like count, sum, etc.) which, while are well defined and supported in the relational world, are not still included in any semantic query or rule language. Current research efforts within the European integrated project SUPER are leading to the design of such a language, thus enabling the formal definition of post processing metrics over analysis results [12]. We adopt this extension of the WSML logical expression to define our post processing metrics.

Finally, the reporting 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 an evaluation template may be intended in a broader way and include actions like triggering alarms or sending e-mails. This last reporting step is still being studied and not included in our current implementation.

3 Use Case Scenario: Semantic Reverse Business Engineering

The basic methods behind Reverse Business Engineering (RBE) were developed at the University of Wuerzburg, Germany [13], applied to the SAP R/3 system and converted into the tool Reverse Business Engineer by IBIS Prof Thome AG in collaboration with SAP 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 [14]. 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 [15].

In the process of an RBE analysis the customer chooses one or several analysis scenarios according to his/her 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 Key Performance Indicators (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 involves a lot of manual work and requires a substantiated knowledge about the data structure of the respective information systems. 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.

The remainder of this section describes how our approach has been applied to RBE, yielding a prototype to perform *semantic RBE* (sRBE).

3.1 sRBE Analysis Process

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. In line with the approach described in Section 2, the sRBE analysis process is also composed of the five phases (selection, configuration, execution, evaluation, and reporting).

At the beginning of an sRBE analysis the customer defines his/her aims. 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 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.

3.2 Applying Semantics to Business Questions

Semantic business questions have to be derived from existing RBE business questions and described using ontological concepts. The semantic annotation of RBE content is performed by linking it to ontological concepts. On the one hand, this allows for mapping 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 the following:

- *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.

Note that the first three items support the actual classification of analysis metrics while the last one can be considered as an execution parameter. According to this distinction, we have modelled classification concepts inside the concept *Business Question* as reported in Listing 1.1 using the method explained in Section 2.2.

```
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

Listing 1.2 shows an example of a business question, namely the question "Which sales orders were processed?", that has been modelled in the sRBE ontology. Since this question is classified as a part of an as-is analysis scenario, belonging to an order processing business function and to a transaction data category (lines 5–8), 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.

```
1  concept WhichSalesOrdersWereProcessed_C subConceptOf RBEO#BusinessQuestion
2        nonFunctionalProperties
3              dc#description hasValue "Which sales orders were processed?"
4        endNonFunctionalProperties
5        hasScenario ofType RBEO#AsIsAnalysis
6        hasQuestionType ofType RBEO#List
7        hasDataCategory ofType RBEO#TransactionData
8        belongsToBusinessFunction ofType BFO#OrderProcessing
9
10 relation WhichSalesOrdersWereProcessed (
11           ofType RBEO#SuccessfulExecutionEvent,
12           ofType BFO#OrderProcessing,
13           ofType BRO#MarketingAndSalesRole,
14           ofType RBEO#AnalysisPeriod )
15       nonFunctionalProperties
16             dc#description hasValue "relation related to business question Which sales orders were
                     processed?"
17       endNonFunctionalProperties
```

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

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

```
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

The 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?" [16].

3.3 Implementation Experience

To test our approach, we have developed within the SUPER project an sRBE engine prototype and integrated it with the SUPER architecture. Figure 2 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 (based on IRIS reasoner for WSML[6]) 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 (presented in Figure 3) that, using the functionalities provided by the business logic, allows the users to performs the sRBE process.

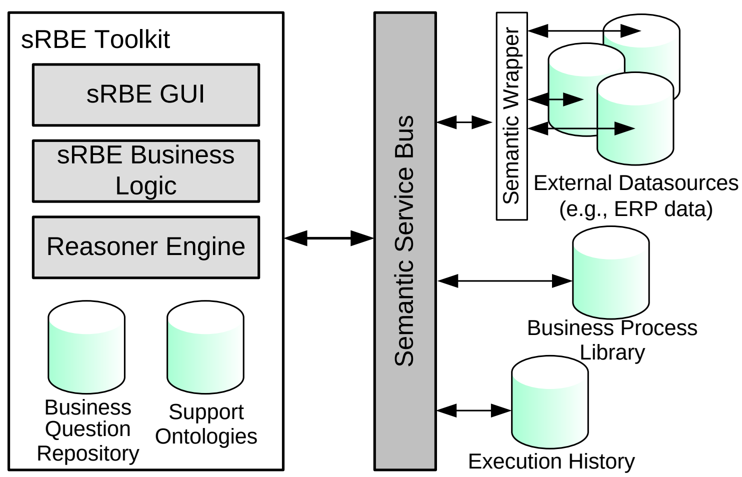

Fig. 2. Architecture of the sRBE prototype and its connection with the SUPER Semantic Service Bus.

The sRBE engine includes a Business Question Repository, where the modelled 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, which contains all the model of the deployed and logged processes; the Execution

[6] `http://iris-reasoner.org`

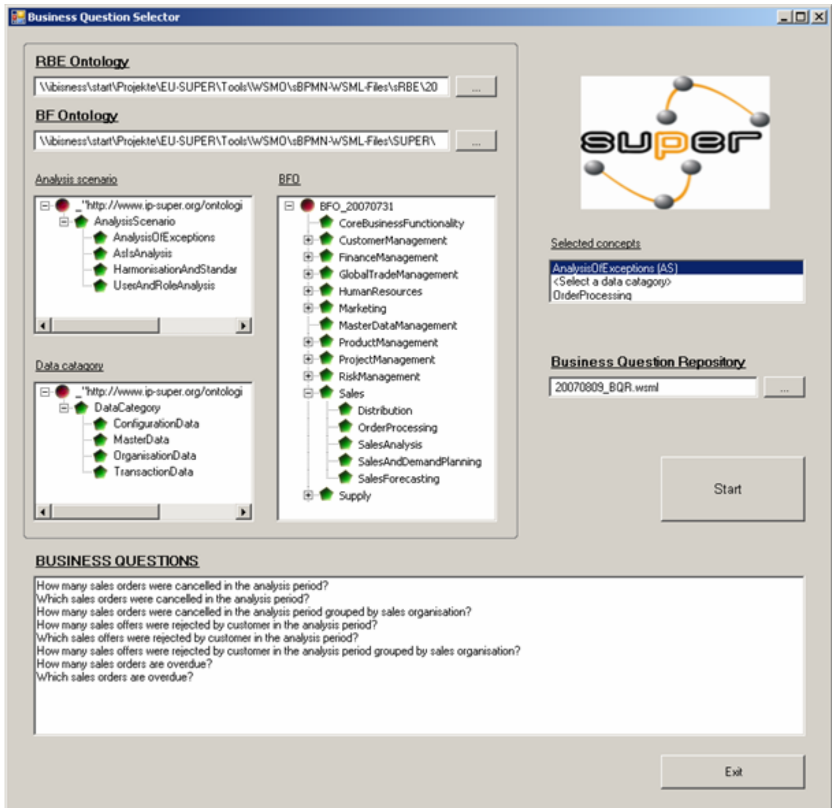

Fig. 3. A screenshot of the sRBE GUI showing the Business Question selection facility.

History, which has all the log of the processes executions; and finally, any other repository (also non semantic) with data needed to analyze business processes (e.g. ERP data).

Data exposed by the Semantic Service Bus are seamless integrated through the IRIS reasoner engine, making it possible to perform queries over distributed data.

4 Related Work

The idea of using semantics to perform process analysis is not new. In 2002, Casati et al. [3] introduced the *HPPM intelligent Process Data Warehouse (PDD)*, 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 [4] is a follow-up of the work in [3]. 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. [5] 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. [7] and O'Riain et al. [6] 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 [7] can be seen as the extension of OLAP tools with semantics. The work in [6] 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. [17] 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. [18] 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 [17,18] 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 [19].

5 Conclusions

This paper presented an approach towards the adoption of semantics to support scenario-based analysis. The proposed semantic meta-model is suitable for any scenario-based analysis because it encompasses all the phases necessary for the selection, configuration and execution of scenario-based metrics, as well as the evaluation and reporting of results. The immediate gain is the leverage of the scenario-based analysis techniques to the expressive power offered by semantic languages, which allows not only to describe data models but also functionalities and their execution logic.

The major strengths of our approach are related to the "lifting" of the analysis to the conceptual level: the results of the analysis are more precise (because they don't rely on syntactic or *ad hoc* data extractors); there are extended possibilities to reuse models at various levels (e.g. ontologies, business questions); we can get a larger automation of data processing and analysis. On the other hand, the weaknesses of our approach are related to the need for reliable models and ontologies in the whole process, thus requiring the effort to develop and maintain them. Moreover, our approach, even if very promising, is still a bit immature to be applied to large-scale and complex real-world scenarios.

However, we can see several opportunities to achieving the objective of an improved system analysis: semantic technologies are gaining momentum in the community, numerous supporting tools are becoming available and stable; finally, there is an increasing awareness of the need for semantics in business scenarios.

Additionally, the presented approach has been applied to the Reverse Business Engineering technique, one of the most important scenario-based analysis techniques in the context of business process analysis. This application emphasized the benefits that the introduction of semantics brings to RBE, such as a greater level of automation, generation of system-independent analysis, better reuse of sRBE content and better administration of sRBE content.

Finally, as our approach 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 Business Intelligence (BI) purposes (like extractors for data warehouses). Future work will focus on refining our prototype and developing more domain ontologies for the support of semantic-based analysis scenarios, so that it can be tested on a real-life use case scenario.

Acknowledgments

This research has been supported by the EU co-funded IST project SUPER (FP6-026850). We would like to thank the whole SUPER Consortium for their valuable contribution.

References

1. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology.* 2005.
2. IBIS Prof. Thome AG. Reverse Business Engineering Plus. http://www.rbe-online.de.
3. 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.
4. 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.
5. 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.
6. S. O'Riain and P. Spyns. Enhancing the Business Analysis Function with Semantics. In *OTM Conferences (1)*, pages 818–835, 2006.
7. D. Sell, L. Cabral, E. Motta, J. Domingue, and R. Pacheco. Adding Semantics to Business Intelligence. In *DEXA Workshops*, pages 543–547, 2005.

8. European Project SUPER - Semantics Utilised for Process Management withing and between Enterprises. *http://www.ip-super.org/*.
9. M.K. Smith, C. Welty, and D.L. McGuinness. OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004.
10. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3c member submission 21 may 2004.
11. J. de Bruijn, H. Lausen, A. Polleres, and D. 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.
12. S. Heymans, C. Feier, J. de Bruijn, S. Zoeller, and E. Cimpian. Deliverable D1.4: Process Ontology Query Language. Technical report, SUPER Integrated Project, 2007.
13. H. Wenzel. Reverse Business Engineering: Ableitung von betriebswirtschaftlichen Modellen aus produktiven Softwarebibliotheken. *Published dissertation at the chair of business administration, University Wuerzburg*, 1999.
14. A. Hufgard and W. Walz. The Road Behind: Continuous business improvement with Reverse Business Engineering. 2003.
15. R. Thome and A. Hufgard. Avoiding Bad Choices. 2005.
16. C. Ebert. *Best Practices In Software Measurement: How To Use Metrics To Improve Project And Process Performance*. Springer, 2004.
17. 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.
18. 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.
19. I. Celino, A.K. Alves de Medeiros, G. Zeissler, M. Oppitz, F. Facca, and S. Zoeller. Semantic Business Process Analysis. In M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, and N. Stojanovic, editors, *Proceedings of the Workshop SBPM 2007.*, volume 251 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.