Model Federation and Probabilistic Analysis for Advanced OSS and BSS

Jörg Niemöller, Leonid Mokrushin,
Konstantinos Vandikas, Stefan Avesand
Ericsson Research, Services and Software Research
Ericsson AB
Stockholm, Sweden
{joerg.niemoeller,leonid.mokrushin,konstantinos.vandikas,
stefan.avesand}@ericsson.com

Lars Angelin
BUSS DU Technology & System Management
Ericsson AB
Karlskrona, Sweden
lars.angelin@ericsson.com

***Abstract*—Advanced OSS and BSS will be expected to operate cooperatively and across multiple domains and business layers. This can be reached with shared information models providing a comprehensive insight into the entire operated heterogeneous environment. This paper contributes to this vision in two respects. It first introduces a technique for creating a federated information model by inter-relating existing and potentially very different domain specific models. Furthermore, the resulting federated model is used as structural base for defining probabilistic analysis with a Bayesian network. This demonstrates how valuable insights can be obtained through model federation rather than solely relying on separated models reaching only a limited set of information.**

Keywords—BSS, OSS, Model Federation, Probabilistic Analysis, Bayesian network, Business Process

I. Introduction

A networked society based on the internet of things is one major trend of the early 21st century. It refers to a massively interconnected environment. Things that in the past were used separately are now linked to each other and backend applications. The resulting landscape will allow new business models that can span multiple industry domains while at the same time it will need to be operated taking care of the massively increased complexity and the vast amounts of data to deal with. This puts high demands on future solutions for operation support systems (OSS) and business support systems (BSS). A key success factor will be the ability to comprehend and operate increasingly heterogeneous assets. Furthermore, also a closer integration between operation and business support promises more accurate understanding and planning of the business.

A. Current OSS/BSS Solutions

With these demands in mind, a look into today's operation and business support solutions shows that they are often characterized by strict specialization in two main dimensions:

Solutions are tightly coupled with the assets and domains they operate. For example the operation of network nodes is based on vendor specific tools for this particular node. On the business level the situation is similar with solutions for business planning and operation that are specifically designed to the ways of working of a particular enterprise within its industry segment. The second dimension is a strong separation of functionality in the operation support and business support layers.

An inherent symptom of this situation is the usage of a broad range of different models based on different modeling techniques. Every major software solution uses and maintains its own information and function models. Even data that is needed across several applications and tools is often not commonly organized and jointly accessed, but rather replicated. As a consequence there is a lot of incompatibility and redundancy in modeled information. It is easy to see that this is not a good foundation for new features trying to bridge and interconnect formerly separated domains.

B. Paper Overview

The existing situation is in clear contrast to demands of a future business environment. This paper contributes a potential component for an evolution of OSS and BSS solutions by proposing a technique for federating of previously separated and domain specific models. Furthermore, this paper demonstrates how the federated model provides an excellent base to build cross domain analytics functions. The presented work is still in progress.

In Chapter II, this paper provides a general overview of selected modeling techniques and a technique for probabilistic analytics. After Chapter III provides a brief look into the Analytics processes within a BSS/OSS, Chapter IV introduces a technique for federating different models. Finally chapter V will make use of the federated model by using it in order to construct a probabilistic analysis function.

II. Constituent Techniques

This chapter provides an overview of various techniques and languages for modeling information, activities, objects, concepts and in combination the properties of entire environments. As this is a very broad area only prominent examples and the most important categories of technologies are presented. However, all described techniques are highly relevant for the description of an environment operated by OSS and BSS.

978-0-7695-5090-9/13 $26.00 © 2013 IEEE
DOI 10.1109/NGMAST.2013.30

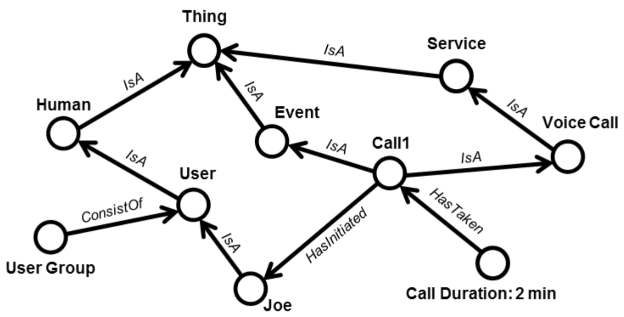


Fig. 1. An example of ontology based knowledge representation

A. Knowledge Modeling with Ontologies

Ontologies are structural frameworks for organizing knowledge and information by formally describing relations between concepts and entities and their properties. This is based on shared vocabulary and taxonomy. The formal nature of the knowledge representation allows the deployment of a broad range of analysis and reasoning techniques that can provide deeper understanding of the underlying information. All this makes ontology based knowledge representation a key technique for applications that try to understand their environment and also share their insights in a structured way. OSS and BSS are characterized by this kind of applications.

An ontology based model is based on a few generic building blocks that exist in one way or another in most techniques for describing ontologies. However, there is a lot of redundant terminology in use. Here is an overview a few central concepts:

- Classes are sets or collections that express concepts or types of objects and things. This is frequently used for categorization. In the example in Fig. 1 "User" is for example a class that represents all users. This class is a specialization of the class "Human", thus all users are also humans.
- Individuals instantiate classes. In Fig. 1 "Joe" is an individual that instantiates the class "User".
- Relations are the attributes and properties of classes and individuals. For example the individual "Joe" in Fig. 1 is related to the class "User" with the property "IsA" in order to express that Joe is a user. Thus relations express aspects, properties, features, characteristics or parameters that objects and classes can have.
- Assertions are explicitly defined relations. They are in contrast to inferred relations that are found by employing logic reasoning. In the example in Fig. 1 it is asserted, that "Joe" is a "User" and that "Users" are "Humans". Thus it can be inferred, that "Joe" is a "Human".
- Rules are statements in the form of if-then sentences. They define conditions, prerequisites and constraints of relations thus additional logical inferences that can be drawn from an assertion.

There are a couple of examples of languages and related tools that can be used in order to define ontologies. The most widely used example is the Web Ontology Language (OWL) [1]. It is specified by the World Wide Web Consortium (W3C). OWL addresses the tradeoff between expressiveness and decidability by means of language variants, ranging from OWL Lite through OWL DL, to OWL Full.

B. Business Process Models

Business processes describe a structured collection of tasks that together serve a particular purpose. This can for example be a sequence of activities with interleaving decision and branching points that express conditions and rules for the execution of activities. The goal of a business process model is to visualize activities and how they relate to each other and potentially also to enable execution of the process. For execution, activities within a business process need to be instantiated with service instances. Furthermore the business process itself can implement a service. Then ordering the service will then start the business process and delivering the service means executing the process.

Widely used languages for modeling business processes are the "Business Process Model and Notation" BPMN [8] and the "Business Process Execution Language" BPEL [10]. Both can provide business processes with execution semantics.

Events trigger and control the execution and express interaction with the environment. Connections model the flow of activities. Gateways allow conditional forking and merging of execution flows. These are the most basic building blocks of business processes that are here referred to using BPMN terminology.

OWL-S [9] is a language based on OWL that allows establishing a semantics process model. The resulting Ontology enables automatic discovery, invocation and composition of services. This is a good example where two modeling techniques for ontologies and business processes are combined in order create valuable new possibilities.

C. Databases

Databases provide a more or less structured storage of information. There are many different types of databases that are mainly distinguished by the way the information is organized, presented and accessed.

An example with many practical applications is relational databases. They are based on tables that store instances of data with a formal structure that is strictly determined by a schema. Furthermore references between tables constitute relations between distinct sets of data. A structured query language (SQL) allows to access data by formulating a query based on the schema.

In document-oriented databases a set of data is referred to as document. They are less strict than relational databases regarding the internal structure of a dataset as no fixed schema is required. However if the document is based on known description formalisms like for example the extensible markup language (XML), this can be utilized in formulating data access queries that reach into the documents.

Graph databases organize datasets as nodes of a graph. The edges of the graph represent direct links to adjacent datasets. Access is performed by following these links. The resulting structure can be very similar to that of ontologies.

These are just three prominent examples. It is important to note in the context of this paper that databases do not just store raw data, but they provide means to formally organize it. A schema defines the meaning of rows and columns within tables. The links of a graph database define how adjacent datasets are related to each other.

D. Probabilistic models and Bayesian networks

Bayesian probability interprets probability as an abstract quantity that can be used to represent and quantify a state of knowledge. Furthermore, the probability, thus the quantified state of knowledge can be calculated from pre-assigned probabilities.

A Bayesian network is a structured approach to utilize bayesian probability for practical analysis. It is a probabilistic graphical model that captures the conditional dependencies between variables. The Bayesian network is formally a directed acyclic graph (DAG). The variables are represented by the nodes the DAG, while an edge between two nodes expresses the direct conditional dependency between the respective variables.

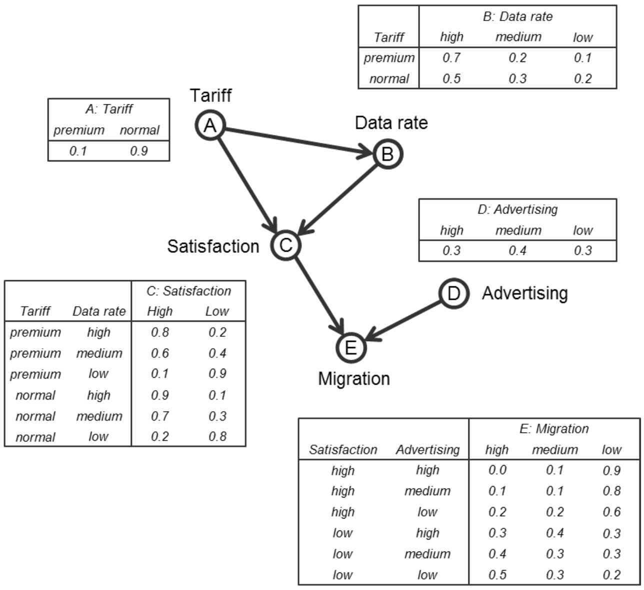
Fig. 2 shows an example Bayesian network with five variables. Each of these variables is interpreted to represent a different property. For example the variable C is interpreted as user satisfaction and the variable value respectively indicates the probability of high or low satisfaction. Another variables (A) is interpreted as tariff that can be premium or normal. Furthermore there are variables interpreted as provided data rate (B), advertising effort (D) and tendency to migrate (E). The variable value captures in these cases the respective probability for being “high”, “medium” or “low”.

The directed acyclic graph shows that variable E directly depends on C and D, but it does not directly depend on A and B. However C depends on A and B, thus E conditionally depends on A and B through C. Furthermore in this example D is mutually independent from A,B and C, while only E depends on D.

The tables shown in Fig. 2 quantify the probabilistic dependencies between variables. From these tables the value of a variable is determined. For example the probability of the satisfaction being high is 0.8 in case the tariff is premium and the experienced data rate is high. Please note, that each additional dependency adds a dimension to these tables.

The quantification of the dependencies is crucial. There are several ways to achieve it. An expert can of course manually edit the values. But with increased complexity this will become less practical. Another method would be to calculate the values from historical datasets, thus training the Bayesian network. It might also be feasible to use either method in combination on subsets of the network.

This example demonstrates the usage of discrete values and tables for quantifying the dependency of variables. If needed,



A: Tariff	
premium	normal
0.1	0.9

	B: Data rate		
Tariff	high	medium	low
premium	0.7	0.2	0.1
normal	0.5	0.3	0.2

		C: Satisfaction	
Tariff	Data rate	High	Low
premium	high	0.8	0.2
premium	medium	0.6	0.4
premium	low	0.1	0.9
normal	high	0.9	0.1
normal	medium	0.7	0.3
normal	low	0.2	0.8

D: Advertising		
high	medium	low
0.3	0.4	0.3

		E: Migration		
Satisfaction	Advertising	high	medium	low
high	high	0.0	0.1	0.9
high	medium	0.1	0.1	0.8
high	low	0.2	0.2	0.6
low	high	0.3	0.4	0.3
low	medium	0.4	0.3	0.3
low	low	0.5	0.3	0.2

Fig. 2. An example Bayesian network showing the directed acyclic graph of probabilistic variables and the probabilistic dependencies.

this can be replaced by using continuous functions by which the variable value can be calculated.

The reason why Bayesian networks are useful is that it provides means of performing tractable calculations locally on the network whilst using all information of the joint probability distributions. It has been proven that every discrete probability distribution and many continuous ones can be represented by a Bayesian network.

III. ANALYTICS IN BSS/OSS

Operation and business support systems deal to a great extend with detecting and curing problems and with optimizing the operated environment with respect to common concerns. This can for example be the reduction of operational expenses while at the same time fulfilling service level agreements and committed key performance indicators. OSS/BSS tooling supports these objectives with functionality such as visualization, symptom detection, analysis of the operated assets, etc. The following steps show the entire lifecycle of treating operational concerns:

- Symptom detection: How to detect a situation to be taken care of? This step actually creates the events triggering further actions and it is the first result derived from the analysis of raw data.
- Impact analysis: What are the consequences of the situation? Which parts of the environment of which business objectives are effected? There could be several independent consequences depending on the context. For example the detected user behavior might result in network overload and at the same time in revenue losses. In this step also the severity and priority of the situation might be decided.

- Root cause analysis: What did contribute to the occurrence of the situation? In this step those elements in the operated environment are identified that actually caused the situation.

- Proposal generation: Propose what to do for solving the detected situation. This can be very concrete by proposing a new configuration parameter value to be used or it can be relatively vague by pointing at a domain where additional attention and further analysis is needed. How concrete the result can be depends mainly on the detail level of the underlying models.

- Verification: Check if the change changes the situation in a good way for example by means of simulation.

- Execution: Deploy and execute the proposed changes. This can be automatic or manual with human approval and interaction.

- Feedback deployment: Adapt the parameters of the involved analytics and decision taking processes for better results.

In each of the steps very different techniques can be applied in order to generate a result. Complex event processing methods with pattern matching and pattern detection is a typical candidate for the symptom detection step. Bayesian networks can be a great help for example in the impact and or root cause analysis.

Common to all these steps is that they depend on the availability of a comprehensive model of all relevant aspects and contributing assets. It would not be acceptable to miss important conclusions, just because some data is handled through an incompatible model in an isolated application.

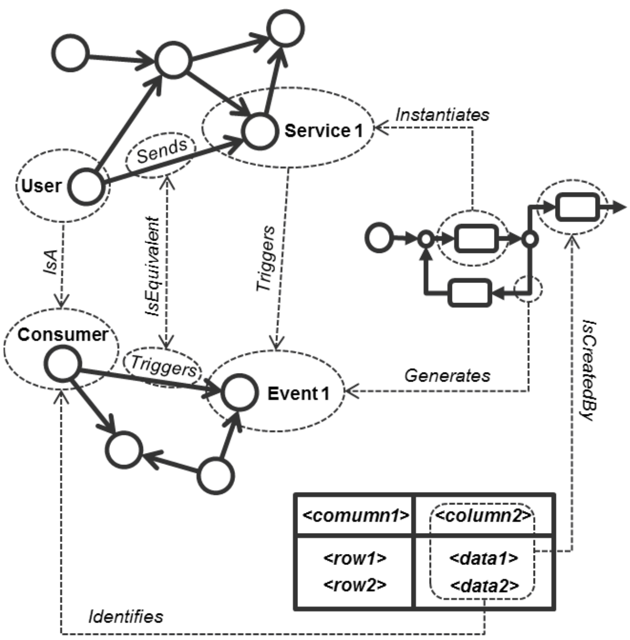


Fig. 4. Example usage of glue model relations

IV. A MODEL FOR MODEL FEDERATION

In the previous chapter we have argued that broad availability of information is the key to good results in OSS and BSS related analysis. A common approach to this challenge is model transformation. This means information elements that are modeled in one model and handled by a particular application are translated into the model and format that is understood by another application. This often implies replication of information.

In this paper, we alternatively propose using a model that allows flexibly relating elements of existing models to each other. The goal is to enable relations even for elements from very different modeling techniques. We show this by example for business process models, database schemas and ontologies. The result is a federated model that integrates building blocks from these constituent models. The model that establishes the relations and thus constitutes the federation is called "glue model".

A. Elements of the glue model

In order to inter-relate elements from various existing models, the glue model defines an abstract relation as atomic element. This can be understood as a relation in the sense of an ontology. For the proposed glue model the endpoints of the relation are not classes or individuals themselves. They are rather references to elements within other models. Thus, the glue model itself is an ontology of references.

Fig. 3. Glue Model Relation with references as end-points

The glue model reference can for example point to data elements within a database and addressed through its schema. Or it can refer to an activity or a connection within a BPMN modeled business process. Also elements from other ontology based models can be the end-point of the glue-model reference.

The reference is expressed using a uniform resource identifier (URI). However, building the URI needs to utilize the modeling language of the targeted model where the referred element resides. Thus, each reference, while being uniform, becomes to some extent specific to the particular modeling technique or language used for the model it is pointing to.

Fig. 4 shows some examples of relations that were established using the glue model. Note that the dotted lines refer to the glue model. This figure shows in a very simplified way two ontologies, a business process and a table from a database. In this example the glue model establishes the following relations:

- The "Consumer" in the first ontology is the same as the "User" in the second ontology.

- The relation "Sends" in the first ontology is equivalent to "Trigger" in the second ontology.

- The individual "Service 1" in the first ontology triggers "Event 1" in the second ontology.

- "Service 1" is instantiated by an activity in the business process
- A connection in the business process, when followed, generates "Event 1" as known in the second ontology based model.
- A column of data in the database is created by an activity in the business process.
- That same column in the database identifies the "Consumer" known in the second ontology.

Many more relations are possible in-between these models. In principle any addressable element in any of the participating models and also within the respective meta-models can be used as endpoints of relations. Furthermore we have only shown binary relations with two endpoints, but also relations with higher arity can be used, thus many-to-many relations are possible.

The glue model can be described based on a mathematical formalism with precisely defined semantics. This allows for automated reasoning. The reasoner exposes a query interface and queries based on the glue model follow the references into constituent models. Implementation-wise this can be solved by a number of dedicated agents that are specialized to de-refer the model specific references. Fig. 5 shows the runtime architecture for executing queries that require information residing within constituent models. This is essential functionality for applications that want to utilize the glue model for example to look for information or perform analysis across the entire federated model.

As briefly mentioned above, the references do not necessarily only relate elements of models. They can also point to elements of the respective meta-models. This can in some cases reduce the number of relations needed. For example the concept of a service might be defined as part of an ontology and the glue model might be used to relate all activities in all business processes to this concept of a service. The many relations needed for expressing this can be replaced by a single relation between the meta-model entity "Activity" and the service concept. This relationship of all used activities can then be inferred instead of explicitly expressed.

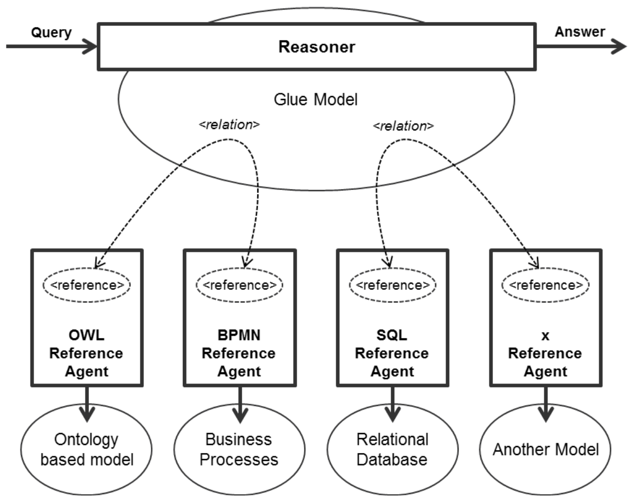


Fig. 5. Runtime architecture for glue-model based queries

The federated model can be used for a number of purposes. A comparably simple use of the federation would be consolidated information retrieval. Queries based on the glue model can collect information pieces from all over the federated information space. This alone is already a very valuable property for designing applications with a broad scope. Instead of mutually integrating them with each contributing model in their scope, the integration is to a great extend assisted by and already solved in the federated model. This makes a federated model a very valuable base of future OSS and BSS if they need to operate within the expected increasingly heterogeneous environment while at the same time broaden their capabilities.

However, a broad scope is not a self-contained requirement. It needs to enable features and insights that were not available without it or at least much harder to achieve. The following chapter demonstrates the value of the federated model for setting up one example of a probabilistic analysis technique.

V. Propabilistic Analysis with Federated models

By using a glue model in order to create a federated model many new relations between originally unrelated domains can be established. Some of the relations express binary facts. For example a user might have initiated the start of a business process or not. This is not gradual. But there are other relations that can be interpreted to express a more gradual quantification. For example the satisfaction of a user can be expressed in many increments between not satisfied or fully satisfied. Also something like the effect of an investment on key performance indicators is in general expressed by a parameter or even a set of parameters that express the extent of the effect.

The federated model as such can only express that concepts or individuals are related but it does not yet quantify the relation. Using relations like "Influences" or "DependsOn" are good examples. It would be highly interesting to quantify them in order to even better understand what actions or changes would have what influence and to what extent.

A Bayesian network can express and, through learning, also find the values that quantify relations. The basic idea is to use the federated model and in particular the glue model relations as blueprint for the directed acyclic graph in order to constitute the Bayesian network. This process of Bayesian network generation from the federated model will be illustrated based on the example shown in Fig. 6.

A. Introduction of the Example Model Federation

The example shows a number of constituent models that are federated using glue model relations. There are two distinct ontology based models. One is focused on user management. It for example expresses what user groups exist, what behaviors users might show and what actions they can perform. The other ontology based model captures business planning and business related activities, for example the available investment opportunities and also business objectives.

The glue model between these two ontologies expresses for example that a "user" in one model is actually equivalent to the concept of "customer" in the other model. This is a typical example of an asserted fact that cannot be further quantified. Furthermore the glue model relates user behavior to the business objective of keeping customers. This relation expresses "influence" that is so far not further quantified.

Next to the two ontology based models there are also two business processes in this example. One of them models the issue resolution process within a network operation center (NOC). The process is therefore related to the concept "issue" known form the user ontology and a user reporting an issue causes the issue resolution to start. The other business process models the decision about investments and also implements the decision. This is reflected in the federated model by relating the business process as a whole and three activities to the respective concepts of the business ontology.

The third model is a database. The datasets contain values that express user satisfaction, the performance of issue solving in the NOC and the number and type of reported issues. The federated model first of all inter-relates the datasets to each other. Here all suspected contributions to user satisfaction are related to the satisfaction dataset. This is a typical example of how a glue model can be applied within one of the constituent models for expressing semantics that is hard to capture with the constituent model's native techniques.

The data set of user satisfaction is also related to the user behavior concept as known in the user ontology, because as user satisfaction is considered to significantly contribute to their observed user behavior. The business ontology is interrelated to the datasets through suspected influence of investments on various performance indicating values.

B. Deriving the Bayesian network

The goal of the Bayesian network based analysis for this example is to quantify the relation between particular investments and the ability to keep customers. Both are concepts from the business ontology model. This model alone provides no means to directly verify and quantify this suspected relation. The federated model however provides additional paths that indirectly express a relation between the investments and the business objective. These paths of relations over multiple nodes on the federated model can be utilized to create a Bayesian network.

A Bayesian network is mainly defined by a set of variables defining the nodes of the directed acyclic graph, directed links between the variables forming the edges of the graph and the definition of what concrete property a variable represents. This

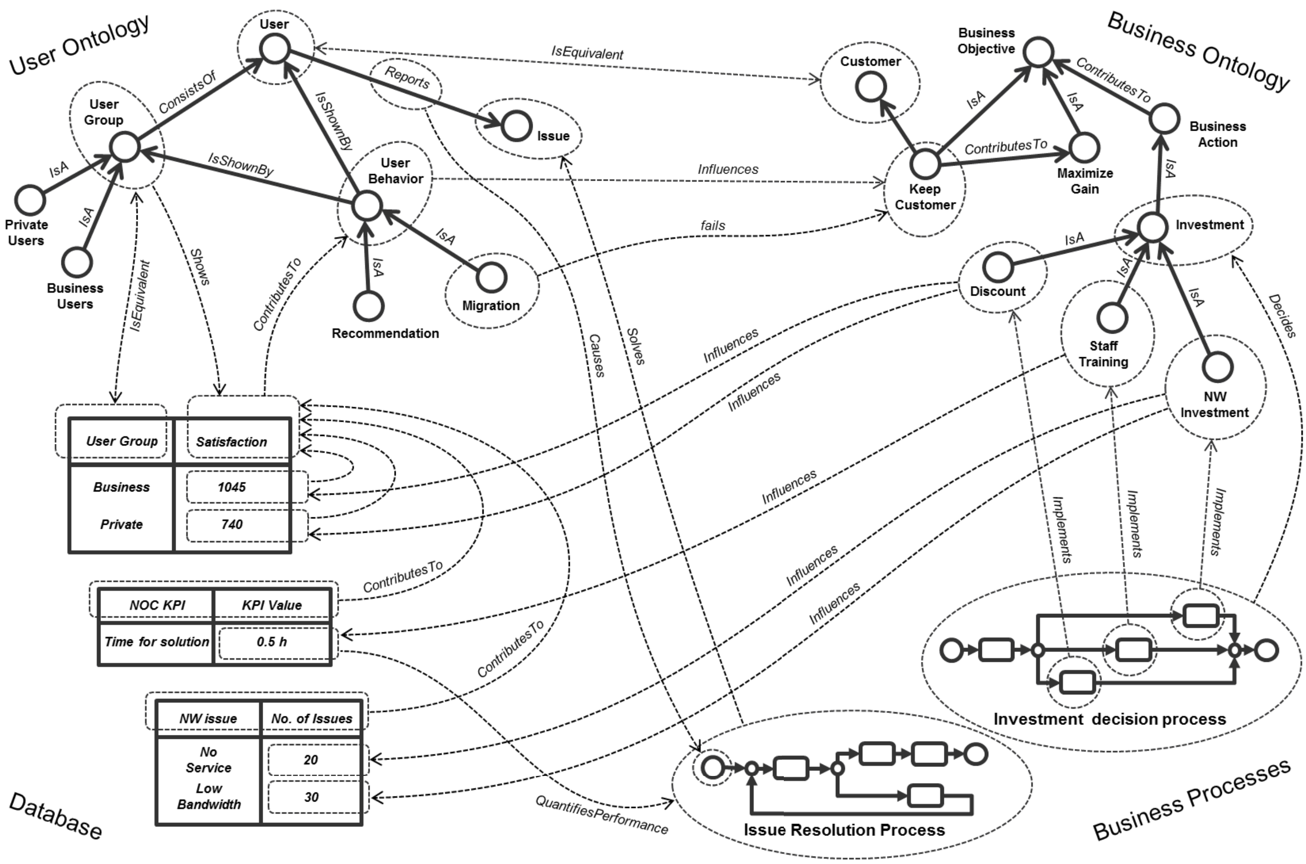


Fig. 6. Example of a federated model consisting of ontologies, databases and business processes

includes the mapping between the value of a variable and a concrete meaning.

Nodes on the federated model are candidates for the variables of the Bayesian network. The three investment options and the business objective to keep customers can immediately be selected, because they are directly concerned with the problem at hand. These variables can also be quantified, for example by means of the invested amount of money or a change in the number of customers per month.

Following all paths on the federated model between the investment options on one side and the business objective on the other one can provide a first outline of the directed acyclic graph. The direction of the graph is determined by the direction of the suspected influence. In this case the chosen direction would be from the investment to the business objective. Furthermore all nodes of the federated model on all identified paths are by default considered to be variables and nodes of the DAG.

The result of the so far taken steps is a graph that contains many redundant nodes and even loops that are not allowed in a Bayesian network. Thus, further steps to reduce and optimize the graph are necessary. They can be executed as a semi-automated and assisted procedure:

- Relations like “IsA” and “IsEquivilant” that basically assert equivalence or instantiation of concepts and individuals, might be treated by removing the edge and merging the nodes.
- Paths through nodes that cannot be quantified in a way that is considered relevant for the problem are removed.
- Paths that establish loops are removed or interrupted by removing relations to irrelevant nodes.

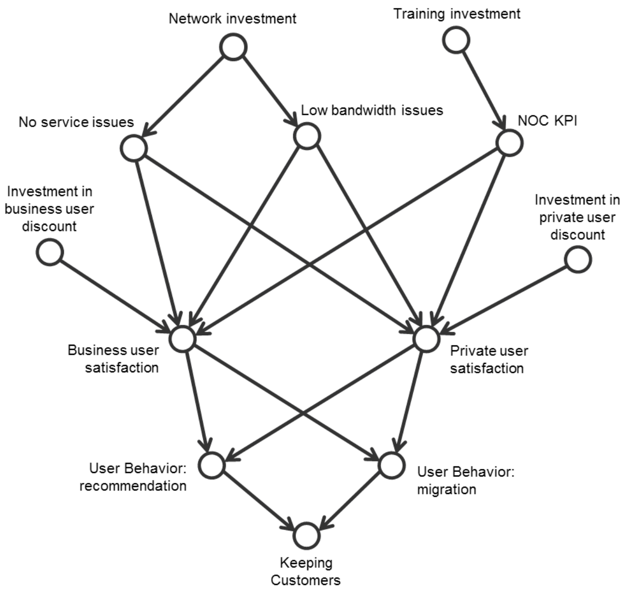


Fig. 7. Bayesian network based on the example federated model

- The direction of relations in the federated model is not necessarily relevant as it can often be changed to an inverse relation.

These steps remove irrelevance and redundancy from the model, but it is also possible to add additional relations when detecting relevant influencing factors that were missed.

Fig. 7 shows the resulting DAG of the Bayesian network of our example. The user satisfaction distinguishes business and private users, thus there are two distinct user satisfaction nodes in the DAG. Note that the investment in discounts was split into discounts for business and private users. This is not directly visible in the federated model. However refinements like this are explicitly allowed.

The “User Behavior” node in the user ontology is split into two nodes reflecting the instantiations of user behavior. The term “Recommendation” refers to the tendency to recommend the service provider to others while “Migration” refers to the tendency to leave the service provider thus being lost as customers.

Connecting the values of the variables to a concrete meaning is the last remaining step. For those variables that originated in the database, concrete values are available. Hence, only a mapping of the probability value range between 0 and 1 to of the concrete values is necessary. It does not matter if a discrete mapping table or a continuous function is used.

For the user behavior, a quantification measure is not yet available. It needs to be created first by interpreting the meaning of the node. Here the user behavior node was interpreted as tendency of a user to show the respective behavior. This could for example be roughly quantified by “low”, “medium” and “high”.

C. Train and Use the Bayesian network

In order to use this Bayesian network for analysis, the peer dependencies between the variables need to be quantified. This can for example be done explicitly by an expert who is manually writing values into the dependency tables. Other analysis functions focused particularly on a single dependency can provide the needed values. Ultimately it is also possible to generate the dependency parameterization with a learning process. This however requires a set of training datasets that fulfill two criteria. They cover all variables that are not yet quantified and a sufficiently representative base of data is available.

With the fully parameterized Bayesian network it is now possible to study the impact of changes in values of the input variables on the rest of the network and in particular on the output variables of interest. In the concrete example variations in the investment will show a particular effect on user satisfaction and ultimately on the business objective to keep customers. This way it is possible to determine the optimal distribution and weighting of investments.

Please note that the presented example is still relatively simplified. It shows the principle while real environment surely contains many more influencing factors or intermediate levels that would need to be considered in the Bayesian network. Nevertheless, the methodology to create, train and use the Bayesian network would be the same. It is easy to see how other Bayesian networks can be designed for different purposes like for example maximizing the company income.

The current example shows an analysis that answers a business planning question by utilizing contributions from network operation and user management. The federated model unlocks structures in the environment that were not visible in the separated models. Thus, federation of models is the moderator that provides sufficient insights and structure for identifying all relevant contributions and deriving a model for probabilistic reasoning. An active support for this kind of comprehensive analysis is a value that many OSS and BSS solutions do not sufficiently provide but that will become increasingly important.

Weighting many different outcomes based on input from many contributing sources is a particular strength of Bayesian networks. This is especially useful for impact analysis and root cause analysis. With a comprehensive approach as presented in this paper, impact can be identified all over the modeled environment and across all layers with a semi-formal method.

After the probabilistic analysis with the Bayesian network an optimal strategy might be found. The following question would be how and where changes need to be deployed. Also this question can be answered based on the federated model. In our example the investment decision process and activities within the process are linked to the investment concept through "Decides" and "Implements" relations. These relations involve the nodes in the federated model that directly or indirectly correspond to endpoints in the Bayesian network. Interpreting these relations as pointers to impacted parts of the environment directly guides the following actions. In this case this leads to the business process might need to be modified in order to deploy a new investment policy.

VI. Conclusion and Outlook

In this paper we have introduced a lean and flexible technique for creating glue models, which are defined as models that inter-relate entities of other models. We have shown how a glue model is used for creating a federated model that consolidates separately handled models from all over a heterogeneous environment. This is driven by expected needs of future operation and business support systems. Therefore, the federated model potentially spans across traditional domain and technological borders as well as across operational and business layers. It tries to capture all contributions that are expected to be in the scope of OSS and BSS and it presents a consolidated access to available information.

The federated model being formally described opens a broad range of use cases. It allows next to qualitative insights also the deployment of analytics and evaluation techniques that create a quantitative understanding of dependencies. As an example we have demonstrated how to derive the structure of a Bayesian network from the federated model. This Bayesian network relates investment decisions to their impact on a particular business objective. While their direct correlation is suspected, it could not be directly quantified. The federated model has unlocked further information sources and relations. Thus, by applying a multi-stage detour across other models it became finally possible to finally the influence of the investment decision on this business objective.

The provided consolidated access and understanding of various independent information sources across domain-borders makes the presented modeling techniques very relevant for OSS and BSS solutions. Therefore they would be a valuable part of a technical foundation and solution for evolved BSS and OSS, when preparing them for future demands.

References

[1] "OWL 2 Web Ontology Language Document Overview". W3C. 2009-10-27.

[2] Common Logic (CL): a framework for a family of logic-based languages, ISO/IEC IS 24707:2007

[3] M. Ashcroft, "Bayesian networks in Business Analytics", Proceedings of the Federated Conference on Computer Science and Information Systems pp. 955–961, 2012, ISBN 978-83-60810-48-4

[4] R.E. Neapolitan, "Learning Bayesian networks", Prentice Hall Series in Artificial Intelligence, 2004

[5] A. Darwiche, "Modeling and Reasoning with Bayesian networks", Cambridge University Press, 2009

[6] C.R. Gutiérrez (Editor), "Advances in Knowledge Representation", 2012, InTech, ISBN 978-953-51-0597-8

[7] J.G. Blodgett, R.D. Anderson, "A Bayesian network Model of the Consumer Complaint Process", Journal of Service Research, Volume 2, No 4, May 2000, pp 321-338, 2000 Sage Publications Inc.

[8] Object Management Group (OMG), "Business Process Model and Notation", www.bpmn.org

[9] D. Martin, M. Burstein et. al.:"OWL-S: Semantic Markup for Web Services", W3C Member Submission 22 November 2004, www.w3.org

[10] Organization for the Advancement of Structured Information Standards: "OASIS Web Services Business Process Execution Language v2", April 2007