

Association for Information Systems AIS Electronic Library (AISeL)

ECIS 2011 Proceedings

European Conference on Information Systems
(ECIS)

Summer 10-6-2011

DECISION SUPPORT BY AUTOMATIC ANALYSIS OF BUSINESS PROCESS MODELS

Bart-Jan van Putten

Clarissa da Silva Correa Romeiro

Leonardo Guerreiro Azevedo

Follow this and additional works at: <http://aisel.aisnet.org/ecis2011>

Recommended Citation

van Putten, Bart-Jan; da Silva Correa Romeiro, Clarissa; and Guerreiro Azevedo, Leonardo, "DECISION SUPPORT BY AUTOMATIC ANALYSIS OF BUSINESS PROCESS MODELS" (2011). *ECIS 2011 Proceedings*. 25.
<http://aisel.aisnet.org/ecis2011/25>

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2011 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

DECISION SUPPORT BY AUTOMATIC ANALYSIS OF BUSINESS PROCESS MODELS¹

Bart-Jan van Putten, SAP Research Dresden, Chemnitzer Straße 48, 01187 Dresden, Germany; Humboldt-Universität zu Berlin, School of Business and Economics, Spandauer Straße 1, 10178 Berlin, Germany, bart-jan.van.putten@sap.com

Clarissa da Silva Correa Romeiro, NP2Tec – Research and Practice Group in Information Technology, Department of Applied Informatics, Federal University of the State of Rio de Janeiro, Av. Pasteur, 458, Urca, 22290-240, Rio de Janeiro, Brazil, clarissa.romeiro@uniriotec.br

Leonardo Guerreiro Azevedo, NP2Tec – Research and Practice Group in Information Technology, Department of Applied Informatics, Federal University of the State of Rio de Janeiro, Av. Pasteur, 458, Urca, 22290-240, Rio de Janeiro, Brazil, azevedo@uniriotec.br

Abstract

It is advantageous for companies to have an in-depth understanding of their business processes. To support companies in decision making, based on the properties of their business processes, a method was developed for the automatic analysis of business process models. A machine-readable representation of the model is parsed to extract several features. Based on a set of domain-specific business rules, a recommendation is generated. The method was validated by implementing it in a software program and applying it to the domain of product data storage. Several experts in that domain participated in a survey. From the three features tested in this study, the 'data access frequency' seems to be most useful. This feature could thus be reused in future applications of the method. The method could be helpful for companies that have many large, complex, dynamic business processes, and which would like to (dynamically) optimize product data storage. In addition, by replacing a set of domain-specific rules, the method may be applied to other domains where business process models need to be analyzed to support decision making.

Keywords: business process modelling, decision support, feature extraction, smart products.

¹ This research is partly funded by the German Ministry for Education and Research (BMBF) under contract 01IA08002, project SemProM, from February 2008 through January 2011.

1 Introduction

Many companies want to store more information directly on the ‘smart’ products that they produce and sell, e.g., with RFID technologies (Chui et al. 2010). For example, in the domain of manufacturing, companies want to store production process related information, such as the product’s customized finishing, desired delivery time, and results from quality checks, directly on the product (Günther et al. 2008). In the domain of retail, companies want to provide customers with more information so that the customer can make more informed buying decisions (Schmitz et al. 2008; Klein & Permenter 2010). E.g., ‘local’ information on a bottle of wine could provide details on the taste of the wine, while information on perishable products such as milk, meat, and vegetables, could show if the product was not exposed to an extreme temperature.

It is however questionable if local storage is always the best solution. For example, when information often needs to be updated by the product manufacturer, it may be better to keep the information centralized and have the products connect to a central server where the information is stored (‘referenced’). In some cases, the best solution may be to store information partly decentralized and partly centralized (‘distributed’). Such different ‘data storage types’ may incur different costs and benefits for different business processes (Jaenen et al. 2010). Unfortunately, deciding which data storage type is most beneficial is difficult, as many properties of the business process need to be taken into account.

This study investigates how business process models could be used to support making the data storage type decision. More specifically, a method was developed for the *automatic analysis* of business process models. This method could be helpful for companies that have many large, complex, dynamic business processes, and which would like to (dynamically) optimize product data storage. In addition, by replacing a set of domain-specific rules, the method may be applied to other domains where business process models need to be analyzed to support decision making.

The remainder of this paper is structured as follows: section 2 describes the proposed method. Section 3 describes how the method was applied to the product data storage type case. In section 4 the validation of the method is described. The results are discussed in section 5 and section 6 concludes this paper and provides directions for future work.

2 Method Design

The method is intended to be used as part of a decision support system. Although the authors believe that after some adaptations the method may be used for a wide range of decision support situations, the decision was made to ‘start small’ and focus on one specific case, namely the data storage decision. Therefore, the required output of the method is a recommendation for a certain data storage type (local, referenced, or distributed). The input for the method is a business process model describing how the product is expected to be used and which data intensive tasks will occur. Business process models describe, in a structured way, the logical order and dependence of activities within an enterprise whose objective is to produce a desired result (Aguilar-Savén 2004). They help the organization to understand the information flow and serve as a strong base for many tasks in different research areas.

To get from the input to the output, five steps need to be executed (Figure 1). The first step is to make sure that the business process model is available in a machine-readable format, e.g., BPMN 2.0 XML (OMG 2010), so that it can be parsed by a software program. The second step is to extract features from the model, such as ‘data access frequency’, which may in turn be based on lower level indicators, such as ‘the number of tasks accessing a data store object’ and the ‘total number of tasks’. The third step is to execute business rules which specify the most suitable data storage type, depending on the

value of the feature. For example, the rule may be that when the data access frequency is low, the recommended data storage type is 'referenced'. Thresholds for the levels, i.e., what is high and what is low, need to be set by the user of the method. The fourth step is to resolve any conflicts that may occur between rules. E.g., based on the data access frequency rule, referenced storage should be recommended, but based on the network availability rule, local storage may be recommended. To resolve this conflict, rules need to be weighted by the user of the method. Based on the number of times a certain recommendation occurs and the weight of the specific recommendation, one recommendation can then be generated and presented to the user.

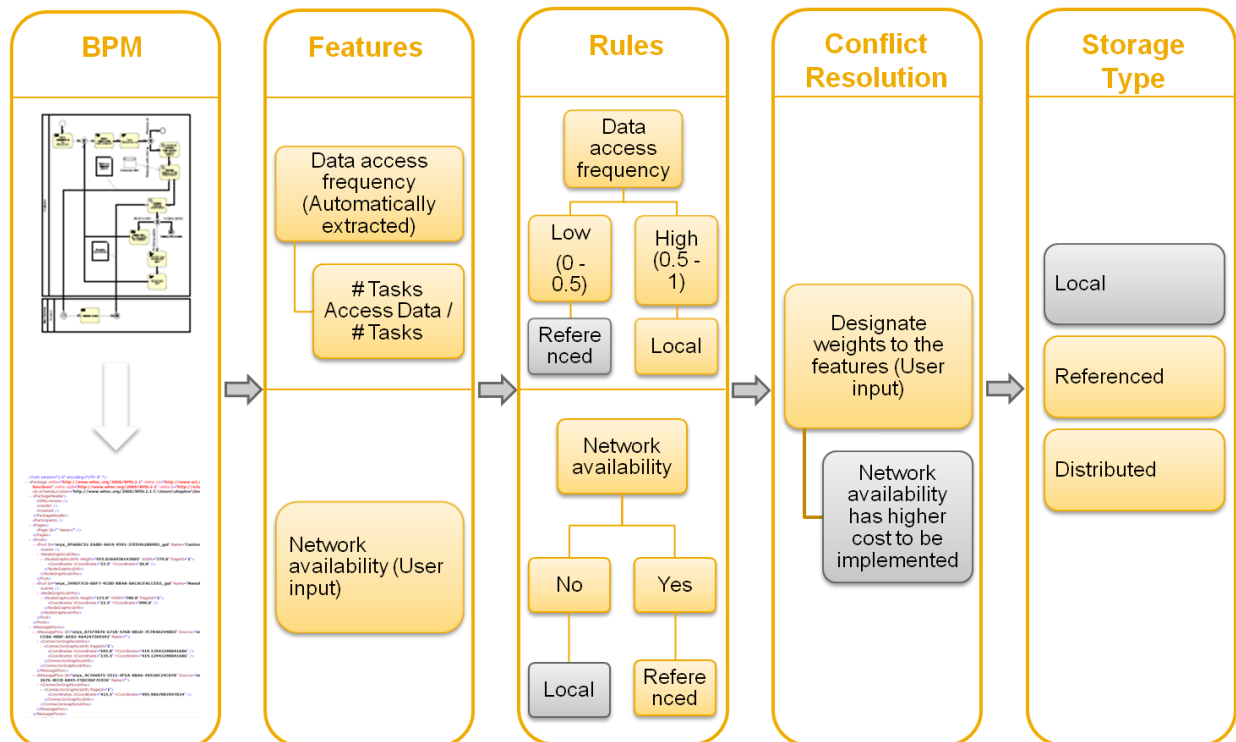


Figure 1. Overview of the analysis method.

3 Method Implementation

To enable the validation of the concept, the method was implemented in a software program, for the specific case of the data storage type decision. The development and working of this algorithm will now be explained in more detail.

3.1 Selecting the modelling language

The first step was to select a suitable business process modelling language. Therefore, some requirements were defined:

- It should have a graphical representation;
- It should have a machine-readable representation;
- It should support the modelling of communication and data in- and output (this is especially useful for modelling business processes in the context of the data storage type decision);
- It should be well-adopted or have the potential to become well-adopted by the BPM community.

Ko *et al.* (2009) provide an overview of several BPM languages, classified as graphical standards, interchange standards, execution standards, and diagnosis standards. As it does not lie within the scope of this research to simulate and execute business process models, the languages of interest were selected from the graphical and interchange classes. From those, five of the most stable and popular ones have been selected for comparison against the requirements. Finally, BPMN 2.0 (OMG 2010), was selected after a rather extensive analysis process which is beyond the scope of this paper.

3.2 Selecting the modelling environment

The second step was to select the modelling environment. The requirements were that it should support visual modelling of BPMN 2.0 and the automatic conversion of such a visual model to a BPMN 2.0 XML document. Two tools were selected: Gravity, originally a BPM plug-in for Google Wave for collaborative business process modelling (Elliott 2009), and Oryx (Hasso Plattner Institute 2010), an academic open source framework. Both tools are web-based and support BPMN 2.0. The support of Oryx for BPMN 2.0 was more comprehensive and therefore Oryx was selected for use in the study at hand.

3.3 Defining the business rules

The third step was to define the business rules in collaboration with domain experts. Table 1 shows the business rules in tabular form. E.g., when the data storage duration is short, the recommended storage type is local. Such rules are domain specific. This means that depending on the domain where the proposed method is applied, different business rules will need to be defined. For this study's case 30 rules were defined. The three rules that will be evaluated as an example in this paper concern the storage duration of product data, the collaboration and interaction among 'smart products' (Wahlster *et al.* 2008), and the frequency in which product data needs to be accessed.

Feature	Value	Local	Distrb.	Ref.
Data storage duration	short term	x		
	long term			x
Collaboration / interaction	necessary		x	x
	not necessary	x		
Data access frequency	high	x		
	low			x

Table 1. Examples of business rules.

3.4 Defining the features and indicators

The fourth step was to define the features and indicators that need to be extracted from business process models in order to be able to evaluate the business rules. The left column in Table 1 shows the features. The three features for which a feature extraction algorithm was implemented are: data storage duration, collaboration/interaction, and data access frequency. These features were selected from a larger set, based on the researchers' expectations regarding the feasibility of automatically extracting them. Each feature is calculated based on one or more indicators. Each of the features and the associated indicators will now be explained.

Data storage duration concerns the time the data is needed. Short-lived information may be stored locally since storage space will be cleared before it is needed again for other data. However, long-lived information is often better stored referenced. There are several indicators for this feature:

- Number of tasks: The number of tasks gives a direct notion of the size of the model and the bigger its size, the more likely it is that the process has a high duration.
- Relative incoming message flows: The number of incoming messages may influence the duration of a process as tasks may only be executed after the message has arrived. This indicator is calculated relatively to the size of the model.
- Relative delay events in normal flow: Intermediate catch events, more specifically the timer, message, signal, and condition events that appear in the normal flow indicate that the process has to wait for a message, a certain time, a condition to become true, or a signal to happen. This indicator is calculated relatively to the size of the model.
- Sequentiality: A sequence in a model indicates that an activity has to wait for the predecessor node to finish, making the process less efficient compared to parallelized processes. This effect is calculated based on Mendling's metric of sequentiality (Mendling 2008). If sequentiality is 1, then the model is a complete sequence of tasks and events.
- Relative loops: A loop means repetition, increasing the time as long as the loop's condition is true. This indicator is calculated relatively to the size of the model.

Collaboration/interaction concerns products that need to communicate or interact with other products. For example, a product could check if another product is available and if so, recommend itself for a lower price. Such interactions require more complex logic, which is often stored in a distributed or referenced manner. Indicators that can be used for this feature are:

- Relative message flows: The number of messages (incoming and outgoing), relative to the size of the model.
- Relative handover sequence flows: The number of message flows and sequence flows between different pools and lanes, relative to the size of the model.

Data access frequency concerns the number of tasks accessing a data store object (retrieve and/or store operations). There is only one indicator for this feature, which has the same name and which is calculated relative to the size of the model.

3.5 Defining the thresholds and weights

The fifth step was to define thresholds for the 'fuzzy values' as described in the business rules. I.e., it needs to be clear when the data access frequency is 'high' and when it is 'low'. Moreover, the weights of the features and indicators need to be defined to enable conflict resolution as described in section 2. For validation of the method, some default thresholds and weights were used. However, ideally, the thresholds and weights are determined by experts and tuned by the users of the method.

3.6 Selecting the business process model

The sixth step was to select a business process model for analysis. For this study, three business process models were selected. They all represented use cases for smart products, they were relatively small in size, and they covered different domains (maintenance, retail, and logistics). As an example, the maintenance model has been included graphically in the appendix (Figure 3). It concerns the failure and repair of a dishwasher:

“A customer places the tableware into his dishwasher and selects the appropriate washing program. Unfortunately, the rotary program selector knob breaks off. Normally, this would mean the customer would have to wash his dishes by hand for a few days or even weeks. In the future, the dishwasher could be equipped with a 'digital product memory' (the 'DPG' in Figure 3). With an enabled smartphone, the customer can identify his broken equipment. With support of the data stored in the DPG, the customer can contact the manufacturer through his smartphone. The manufacturer's service department may ship a spare part directly to the customer, or provide him with a CAD drawing to be

printed on a 3D printer in a nearby copy shop. This reduces storage and transportation costs for the manufacturer and improves the speed of the solution process considerably.”

3.7 Executing the software program

The seventh step was to execute the feature extraction algorithm, which was implemented as a C# software program. Oryx was used to convert the graphical business process model to a BPMN 2.0 XML document. The program parses the document, calculates the indicators and presents the results to the user, as shown in Figure 2. The user may then decide to change some thresholds and weights and regenerate the recommendation.

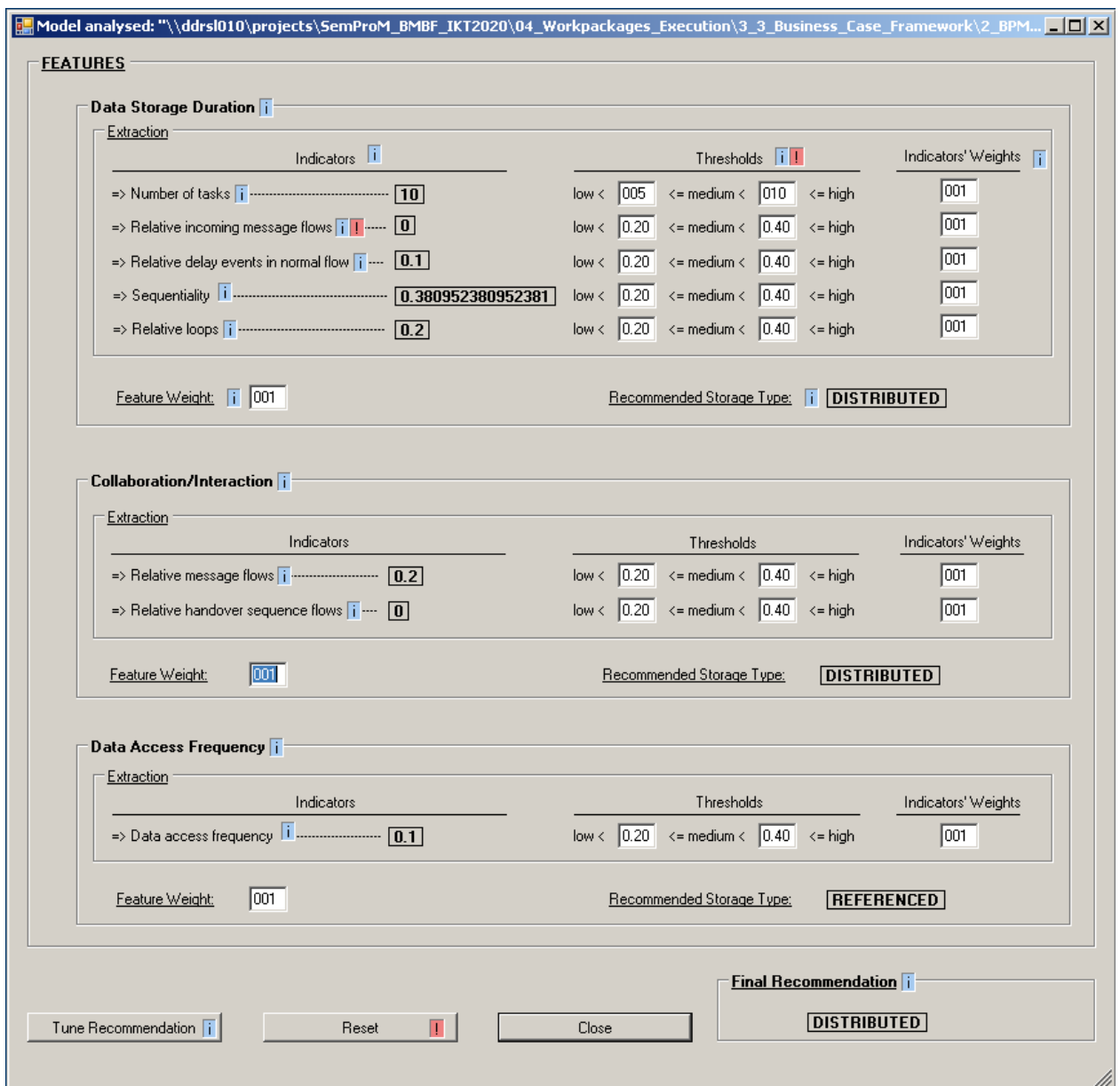


Figure 2. Screenshot of the software program's user interface after parsing the business process model in Figure 3.

4 Validation

4.1 Method

The goal of the validation was to determine whether the method's recommendation would fit with recommendations provided by domain experts. 17 experts were selected among researchers in a research project investigating (among other things) the local storage of data during production processes. Six experts took part in the online survey. The experts' recommendations were investigated in several stages: first, a graphical business process model was presented and they were asked which data storage type they would recommend. Second, the experts were asked to judge the levels (high, mid, low) of the features and indicators in the business process model, in order to be able to tune the algorithm's pre-defined thresholds for the indicator levels. Third, the experts were again asked to indicate which data storage type they would recommend, to see if they had changed their minds based on the features and indicators. Fourth, the experts rated the usefulness of the features and indicators for making the data storage type decision. This sequence of questions was repeated for the three selected business process models. Finally, there were some open questions:

- To what extent do you believe it is possible to determine the right storage type based on a business process model?
- Do you believe a software tool extracting features from business process models could assist you in determining the right storage type for a certain business process?
- Should we consider another feature?
- Should we consider another indicator?

For the test to work, the experts needed to be able to understand the business process models. Therefore, three use cases of product data, which were known to the experts, were selected. Whereas the algorithm could only 'see' the syntax of the models, the experts could also see the semantics, i.e., the names of the model's elements, and read an additional description. This way the experts could imagine the business processes described by the models.

4.2 Results

Directly comparing the experts' recommendations with the algorithm's recommendations is possible, but would be based on the pre-defined thresholds for the indicator levels, which may not be in correspondence with the thresholds that were determined through the survey questions. For example, the algorithm initially uses a threshold of five, to determine whether the 'number of tasks' indicator is low or mid. However, the experts may believe that a 'mid' level of tasks is only reached above 20 tasks. Therefore, the experts' recommendations were evaluated by comparing their perceived feature levels with the business rules. For example, for the first business process model, expert 2 recommended the referenced data storage type. In the next question, expert 2 stated that the data storage duration was high. Thus expert 2 implicitly associated (moderated by any other impressions the expert may have had of the model) that a high storage duration requires referenced storage. This is in conformance with the business rule in Table 1. This way, all 20 answers (which only include the high and low levels and not the mid levels) were checked with the business rules. 70% of the answers (14 out of 20) were in conformance (Table 2).

Feature	Answers	Conform	%
Data storage duration	7	4	57
Collaboration / interaction	6	4	67
Data access frequency	7	6	86
Total	20	14	70

Table 2. Conformance of expert recommendations with business rules.

The difference between the experts' recommendations directly after having seen the graphical business process models and later after answering the questions regarding the features and the indicators were compared as well. Out of the 13 recommendations in total (by all experts over all models), six experts changed their minds. Although it is hard to draw any reliable conclusions from this, it is most likely that this effect is attributable to either a better understanding of the models by the experts after studying the features and the indicators, or to chance.

The usefulness of the features was questioned explicitly in the survey and is shown in Table 3. N represents the number of answers received, Mean is the mean of the scores, where a score of 1 was assigned to the 'useless' category and a score of 5 to the 'useful' category. SD is the standard deviation, to give an indication of the range of answers received. Among these three features, only the 'data access frequency' feature seems to be useful enough.

Feature	N	Mean	SD
Data storage duration	13	3,3	1,3
Collaboration / interaction	11	3,3	1,6
Data access frequency	13	4,0	1,1

Table 3. Usefulness of the features.

In a similar way, the usefulness of the indicators was calculated (Table 4). If the minimal usefulness of 4,0 is applied again, the features 'relative message flows', 'relative handover sequence flows', and 'data access frequency indicator' seem to be useful enough to be used in future versions of the recommender system. However, the 'relative message flows' and 'relative handover sequence flows' indicators are used to calculate the 'collaboration/interaction' feature, which was not perceived to be useful enough by itself. Only 'data access frequency' is an indicator for a feature which is also perceived to be useful by itself. Thus, out of the features and indicators tested in this study, only data access frequency seems to be useful for a recommender system as proposed in this paper. The open questions were answered by only three of the experts and did not lead to any noticeable insights.

Feature	Indicator	N	Mean	SD
Data storage duration	Number of tasks	10	3,6	1,5
	Relative incoming message flows	10	3	0,9
	Relative delay catch events in normal flow	10	3,9	1,5
	Sequentiality	10	3,4	1,0
	Relative loops	10	3,9	1,2
Collaboration / interaction	Relative message flows	7	4,4	0,5
	Relative handover sequence flows	7	4,6	0,5
Data access frequency	Data Access Frequency indicator	8	4,4	1,4

Table 4. Usefulness of the indicators for indicating the features.

5 Related Work

In this section, some work will be presented which aims to derive information from business process models, in order to handle aspects of business or IT.

Fiorini *et al.* (1996) and Weston *et al.* (2004) propose the use of business processes for software requirements elicitation. Fiorini *et al.* propose a method to represent business processes in a conceptual model, aimed at making the relationships between pieces of process information explicit. The first step of the method is the construction of the model. In the second step, elements of the model are linked. In the third step, the analyst navigates on the linked data to locate requirements. Fiorini *et al.* also argue that other techniques, such as interviews and surveys can be used to elicit requirements. However, their experience shows that by using process models it becomes simpler to define concrete business actions.

Weston *et al.* (2004) define 'process thinking', i.e., thinking about current and possible future ways in which organised sets of value added activities can realise business goals by transforming inputs (such as material, sub-products, information and knowledge) into outputs (such as products and services) required by customers. Becker *et al.* (2000) state that modelling languages and tools can be used to enable process thinking for a broad range of purposes, such as business process reengineering, workflow specification, team systems design, and knowledge management. The study at hand proposes decision making as another purpose.

Klose *et al.* (2007), Papazoglou and Heuvel (2006) take a business process model as a starting point and aim to identify services (in a Service Oriented Architecture approach) from certain parts of the process. Azevedo *et al.* (2009) present a method that considers syntactical (structural) and semantic analysis of a process model towards service identification. Services are identified directly from business process elements without considering their value for business decision making. Moreover, the business process models used as input are often exclusively comprised of automatable tasks, while the study at hand proposes a method that also works with business process models consisting of human tasks.

Riehle and Züllighoven (1996) define a pattern as 'the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts'. Van der Aalst *et al.* (2003) define several types of 'workflow patterns' of which at least two are interesting for this work: control flow and data flow. Control flows represent the flow of execution control, e.g., sequence, choice, parallelism and join synchronization. Data flow patterns aim to capture the various ways in which data is represented and used in workflows. These patterns consider e.g., data visibility, data interaction, data transfer and data-based routing. Russel *et al.* (2005) extended the work of Van der Aalst *et al.* to the data perspective and propose thirty nine data flow patterns. Future work may investigate if and how workflow patterns could be used as indicators in the method proposed in this study.

Dijkman *et al.* (2008) define metrics for business process models, which are somewhat similar to the features and indicators, but mainly focus on semantic analysis, while the proposed method strives for a purely syntactic analysis. The work by Mendling (2008) also identifies metrics for business process models, but aims to identify syntactic errors in those models, rather than to create new knowledge that can support decision making.

In the area of 'process mining', information is extracted from information systems (such as Enterprise Resource Planning or Business Process Management Systems) event logs to discover the process model (W.M.P. Van der Aalst & Weijters 2005; Alvarez 2002). Decisions may be made on the basis of such models, but for their construction only the output of information systems is used. Human activities are not considered, since they normally do not appear in the event logs. Thus, using only this approach to identify relevant information for decision making may discard important information regarding human activities.

6 Conclusion

In this work a method for automatic analysis of business process models to facilitate human decision making is proposed. The method describes how features can be extracted from business process models stored in a machine-readable representation, and how those features can be used to evaluate business rules in the respective decision making domain. The method may make decision making more informed and efficient, which could be useful for companies that have many large, complex or dynamic business processes.

The method was evaluated by applying it to the domain of product data storage. Deciding how and where to store product-related data during the different phases of a product's lifecycle, (e.g., production, logistics, retail, usage, maintenance, recycling) can be a difficult issue as many properties of the business process need to be taken into account. Examples of such properties are how long the data is needed and how often it needs to be accessed. Those properties are the 'features' which are extracted and used to decide on the right data storage type: local, distributed, or referenced.

For the purpose of validation, three business process models related to the usage of product data were selected from the areas of maintenance, retail and logistics. The method was implemented in a software program and executed for each of the models. Next, the recommendations generated by the software program were compared with those of human decision makers. Therefore, several experts in the product data storage domain were asked to express which data storage type they would choose and how useful they would find the features and indicators for supporting their decision. The conformance between the experts and the program ranged from 57%-86% depending on the features of interest. Overall, with the current set of features implemented in the program, a conformance was reached of 70%. From the three features tested in this study, the 'data access frequency' seems to be most useful. This feature could thus be reused in future applications of the method.

Although the results provide some early signals regarding the usefulness of the features and indicators, their internal validity is limited due to the rather small number of participants in the survey (6). This can be explained by the relatively small number of people who were invited to take part (17), but also because several participants quit the survey before finishing it, most likely because it required more time than expected. In future work, the experiment could be repeated with a larger number of participants, but the different business process models under review should be distributed over the participants, to shorten the time needed for one person to complete the survey.

In future work, business process models could be analyzed by simulating or executing them. Especially for features that consider time and durations and for models that include repetitive tasks, this could provide more reliable information. Process mining techniques (W.M.P. Van der Aalst & Weijters 2005; Alvarez 2002) could be used to analyse logs resulting from process executions. Besides, new features and indicators could be identified by considering the workflow patterns from Van der Aalst *et al.* (2003) and Russel *et al.* (2005). Another way to obtain more reliable features could be to involve the semantics of the business process model elements, as used by Fiorini *et al.* (1996) in the third step of their method to identify requirements. A simpler approach would be to investigate the use of other features. Human experts probably use many more and other 'features' than just those three evaluated in this study. Examples of such features in the domain of product data could be 'data size' and 'data security'.

To extend the external validity of this research, the method could be applied to the analysis of business process models in other domains. The adaptation to other domains would require the definition of business rules for that respective domain and the definition of features and indicators to be extracted. In domains with a similar focus on the data-centric parts of the business process, features and indicators from this study could be reused. As features are based on indicators, indicators can be more domain agnostic than features. By developing a more extensive library of indicators, features may be composed more easily, to efficiently apply the method to other domains.

References

- Aguilar-Savén, R., 2004. Business Process Modelling: Review and Framework. *International Journal of Production Economics*, 90(1), pp.129-149.
- Alvarez, R., 2002. Discourse Analysis of Requirements and Knowledge Elicitation Interviews. In 35th Hawaii International Conference on System Sciences (HICSS).
- Azevedo, L.G. et al., 2009. A Method for Service Identification from Business Process Models in a SOA Approach. In T. Halpin et al., eds. *Enterprise, Business-Process and Information Systems Modeling*. pp. 99-112.
- Becker, J., Rosemann, M. & Von Uthmann, C., 2000. Guidelines of Business Process Modeling. In W. Van der Aalst, J. Desel, & A. Oberweis, eds. *Business Process Management: Models, Techniques, and Empirical Studies*. LNCS. Berlin Heidelberg: Springer-Verlag, pp. 30-49.
- Chui, M., Löffler, M. & Roberts, R., 2010. The Internet of Things. *McKinsey Quarterly*, (2).
- Dijkman, R.M., Dumas, M. & Ouyang, C., 2008. Semantics and Analysis of Business Process Models in BPMN. *Information and Software Technology*, 50(12), pp.1281-1294.
- Elliott, T., 2009. SAP's Gravity Prototype: Business Collaboration Using Google Wave. Available at: <http://www.sapweb20.com/blog/2009/10/sap%E2%80%99s-gravity-prototype-business-collaboration-using-google-wave/> [Accessed November 24, 2010].
- Fiorini, S., Leite, J.C.S.P. & Soares, T., 1996. Integrating Business Processes with Requirements Elicitation. In 5th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises.
- Günther, O., Kletti, W. & Kubach, U., 2008. *RFID in Manufacturing*, Berlin Heidelberg: Springer-Verlag.
- Hasso Plattner Institute, 2010. Oryx. Available at: <http://bpt.hpi.uni-potsdam.de/Oryx>.
- Jaenen, V., Tummel, C. & Henning, K., 2010. Benefits of RFID for the Production of Hybrid Micro Systems in Flexible Production Networks of SMEs. In V. Mahadevan & Z. Jianhong, eds. 2nd International Conference on Computer and Automation Engineering (ICCAE). Singapore, pp. 185-189.
- Klein, R. & Permenter, K., 2010. *Item-Level RFID Tagging in Retail*, Aberdeen Group. Available at: <http://www.aberdeen.com/aberdeen-library/6504/RA-rfid-retail-inventory-visibility.aspx>.
- Klose, K., Knackstedt, R. & Beverungen, D., 2007. Identification of Services - A Stakeholder Based Approach to SOA Development and its Application in the Area of Production Planning. In European Conference on Information Systems (ECIS). St. Gallen, Switzerland, pp. 1802-1814.
- Ko, R.K., Lee, S.S. & Lee, E.W., 2009. Business Process Management (BPM) Standards: A Survey. *Business Process Management Journal*, 15(5), pp.744 - 791.
- Mendling, J., 2008. *Metrics for Process Models - Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, Berlin Heidelberg: Springer.
- OMG, 2010. BPMN 2.0 Beta 2. Available at: <http://www.omg.org/cgi-bin/doc?dte/10-06-04>.
- Papazoglou, M. & v.d. Heuvel, W., 2006. Service-Oriented Design and Development Methodology. *International Journal of Web Engineering and Technology*, 2(4), pp.412-442.
- Riehle, D. & Züllighoven, H., 1996. Understanding and Using Patterns in Software Development. *Theory and Practice of Object Systems*, 2(1), pp.3-13.
- Russell, N. et al., 2005. Workflow Data Patterns: Identification, Representation and Tool Support. In L. Delcambre et al., eds. *Conceptual Modeling - ER 2005*. LNCS. Berlin Heidelberg: Springer-Verlag, pp. 353-368.
- Schmitz, M., Baus, J. & Dörr, R., 2008. The Digital Sommelier: Interacting with Intelligent Products. In The Internet of Things: 1st International Conference. Zurich, Switzerland: Springer-Verlag, pp. 247-262.
- Van der Aalst, W. et al., 2003. Workflow Patterns. *Distributed and Parallel Databases*, 14(1), pp.5-51.
- Van der Aalst, W. & Weijters, A., 2005. Process Mining. In M. Dumas, W. Van der Aalst, & A. Ter Hofstede, eds. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, pp. 235-255.

Wahlster, W. et al., 2008. Sharing Memories of Smart Products and their Consumers in Instrumented Environments. *Information Technology*, 50(1), pp.45-49.
 Weston, R., Chatka, K. & Ajaefobi, J., 2004. Process Thinking in Support of System Specification and Selection. *Advanced Engineering Informatics*, 18(4), pp.217-229.

A Business Process Model

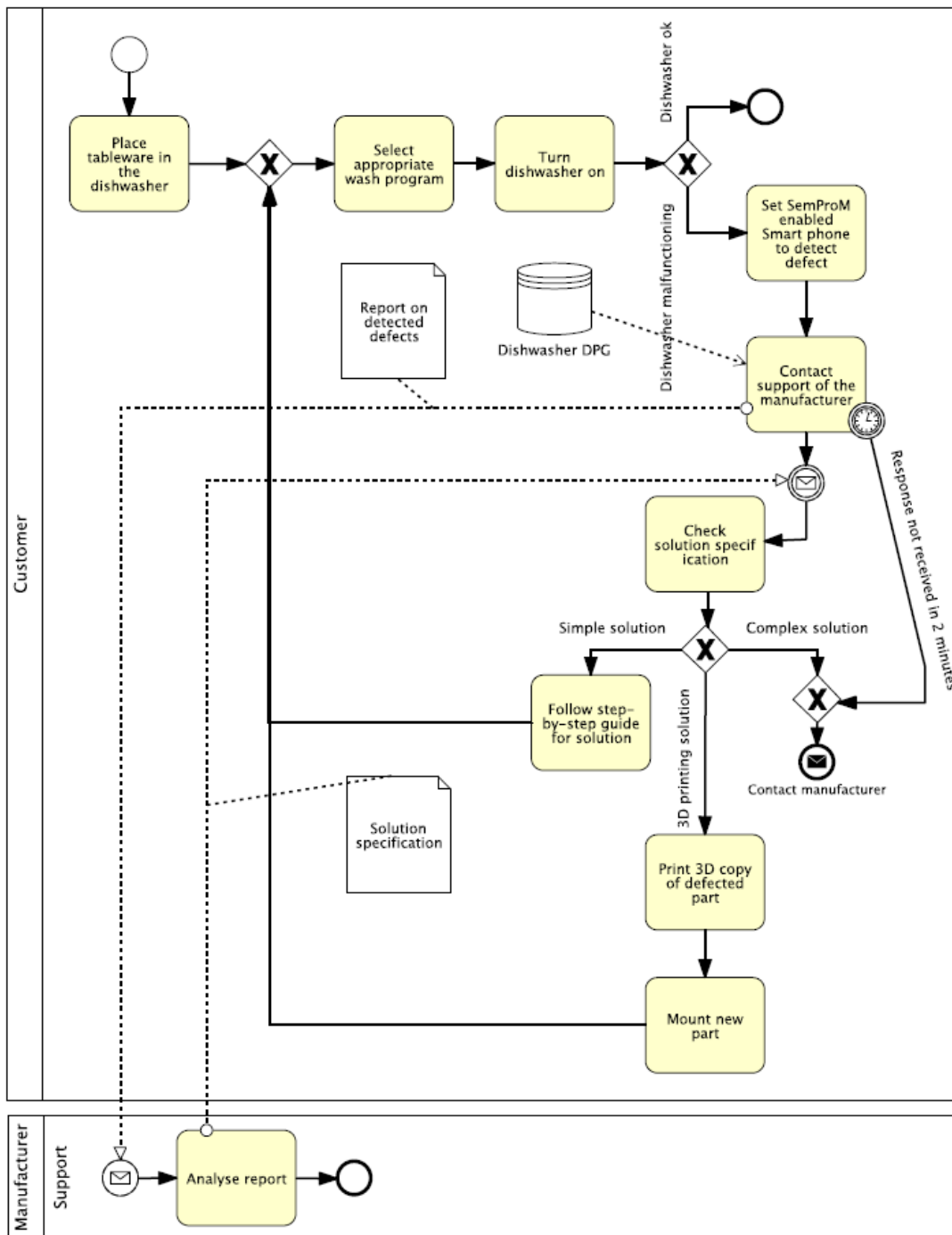


Figure 3. Business process model of a smart dishwasher (DPG is the dishwasher's digital product memory). The automatic analysis result is shown in Figure 2.