

## Association for Information Systems AIS Electronic Library (AISeL)

---

ECIS 2011 Proceedings

European Conference on Information Systems  
(ECIS)

---

Summer 10-6-2011

# CONFIGURABLE PROCESS MODELS: EXPERIENCES FROM A MEDICAL AND AN ADMINISTRATIVE CASE STUDY

Stephanie Meerkamm

Stefan Jablonski

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

---

### Recommended Citation

Meerkamm, Stephanie and Jablonski, Stefan, "CONFIGURABLE PROCESS MODELS: EXPERIENCES FROM A MEDICAL AND AN ADMINISTRATIVE CASE STUDY" (2011). *ECIS 2011 Proceedings*. 33.  
<http://aisel.aisnet.org/ecis2011/33>

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](mailto:elibrary@aisnet.org).

# CONFIGURABLE PROCESS MODELS: EXPERIENCES FROM A MEDICAL AND AN ADMINISTRATIVE CASE STUDY

Meerkamm, Stephanie, University of Bayreuth, Universitaetsstr. 30, 95440 Bayreuth, Germany, stephanie.meerkamm@uni-bayreuth.de

Jablonski, Stefan, University of Bayreuth, Universitaetsstr. 30, 95440 Bayreuth, Germany, stefan.jablonski@uni-bayreuth.de

## Abstract

*Reference (process) models are conceptual models that illustrate generic solutions for a certain domain. However, the generic concept of the reference process models requires the adaption of these process models to the demands of the individual customers. The concept of configurable process models is a step forwards towards this adaptation. Configurable process models integrate different variants of a business process into a single model. The deviation of an individual process model from a reference model can be seen as a variant of that reference model. While a technique for such models was already suggested in previous research, this paper presents two case studies in which this technique was tested in real-world scenarios. We sourced our material from a project at a middle-sized hospital introducing “clinical pathways”, and from our own university documenting the “student-life-cycle”. For each of these projects we identified the variants including all of the variation points. The variation points mark the differences between the variants. After modeling, each identified set of variants was integrated into a configurable process model and these models were configured accordingly. This paper reports on both the observation during the model creation and the subsequent configuration.*

*Keywords: Business Process Models, Configuration, Medical Processes, Student-Life-Cycle, Case Study.*

# 1 Introduction

*Reference (process) models* are conceptual models that illustrate generic solutions for a certain domain (Fettke & Loose, 2003; Frank, 1999). Also called “best practices”, they illustrate best ways of working to achieve a certain objective (Silverston, 2001a, 2001b; Winter & Schelp, 2006). In the context of Information Systems (= IS) and organizational design, reference models are widely accepted in both theory and practice (Fettke & Loose, 2003) for the purpose of documentation (LaRosa & Dumas, 2008) and, mainly for the reason of time and cost restriction, accelerating the modelling of individual process models. Domains for which such reference models have been developed, mainly those driven by corresponding industry consortia, are for example SCOR (Stephens, 2001) for the context of supply chain management, or ITIL (OGC, 2010) for service management. In addition, there is the SAP reference model which captures process supported by SAP (Curran & Keller, 1997).

However, the generic concept of the reference process models requires the adaption of these process models to the demands of the individual customers (Rosemann & Schütte, 1997). In doing so, the application of reference models is motivated by the “Design by Reuse” paradigm (Rosemann & van der Aalst, 2007; van der Aalst, Dreiling, Rosemann, & Jansen-Vullers, 2005; Winter & Schelp, 2006).

The *concept of configurable process models* is a step forward towards the systematic reuse of (reference) process models, or rather their adaption according to individual demands (Fettke & Loos, 2004; Fettke, Loos, & Zwicker, 2005; LaRosa, 2008; LaRosa & Dumas, 2008; Rosemann & van der Aalst, 2007). Configuration is a special design activity. It aims at designing an artifact (here the individual process model) assembled from a set of predefined components, which can only be connected together in a particular way (Mittal & Fraymann, 1989). As basic paradigm a configurable and generic model, for example the reference model, is useful- or rather necessary. It is an integrated representation of multiple variants in one model (LaRosa, 2008; Sabin & Weigel, 1997; Tiihonen & Soininen, 1997). The deviation of individual process models from the initial reference model can be seen as a variant and is achieved by disabling all process elements which are not relevant for the demanded variant (Thomas, 2006).

In order to evaluate the concept of configurable process models in practice, we performed two case studies based on projects of our chair. One project was located in the medical domain. In a hospital (779 beds) in our region we offered support to introduce so called “clinical pathways” as a quality management project. After an it-is analysis we modelled the clinical processes. The other project was located at the administration of our university where, for the purpose of accreditation and self-documentation, we modelled the “student-life-cycle” and the implementation/ cancellation of a degree program. Due to the primary application context the models were originally modelled in German. For this paper, we translated the examples into English.

Based on the material from these projects, we created configurable process models using a specialised form of a so called mereological graph. This is based on an earlier paper (Meerkamm, 2010) but will also be introduced here. This kind of model is intended to implement the configurability by explicitly modelling so called *variation points*. These are the places where the models differ as a result of several so called *variant options*. An example would be where a process can be executed from person A or B; A and B are the variant options, whereas the organizational aspect itself from the process step is the variation point. The graph also considers dependencies between the different variant options. Based on this special data model, the individual variants can be derived by resolving a set of variation points, which are relevant for a specific variant. This is effected by deleting the attached variant options, that are no longer relevant for the demanded variant.

The deletion of unnecessary parts of a process model is a complex task. Modelling experts have to ensure that the resulting model remains valid. Furthermore, knowledge of the process modelling language used is necessary. In order to avoid this it would be useful to give the user some support and to abstract away from the modelling language. Thus we created a so called “variant list”. This

describes the possible variants and their characteristics so that the user only has to select one of them, while the underlying configuration process remains concealed.

The results of the two case studies are presented in this paper which is structured as follows: Section 2 provides an introduction to our approach of configurable process models and the “variant list”. Section 3 reports the generation of the configurable process models and their configuration in the context of our two case studies. Our practical experiences are summarized in Section 4 and an overview of other concepts of configuration management is given in Section 5. The paper concludes with Section 6 in which we present the limitations of the case study and also give an indication of future research.

## 2 Background

Lots of different configurable process modelling methodologies exist such as, for example, C-EPC (LaRosa, 2008; LaRosa, Lux, Seidel, Dumas, & Hofstede ter, 2007; Rosemann & van der Aalst, 2007) or the Provop approach (Hallerbach, Bauer, & Reichert, 2008, 2009) which will, among others, be presented in Section 5.

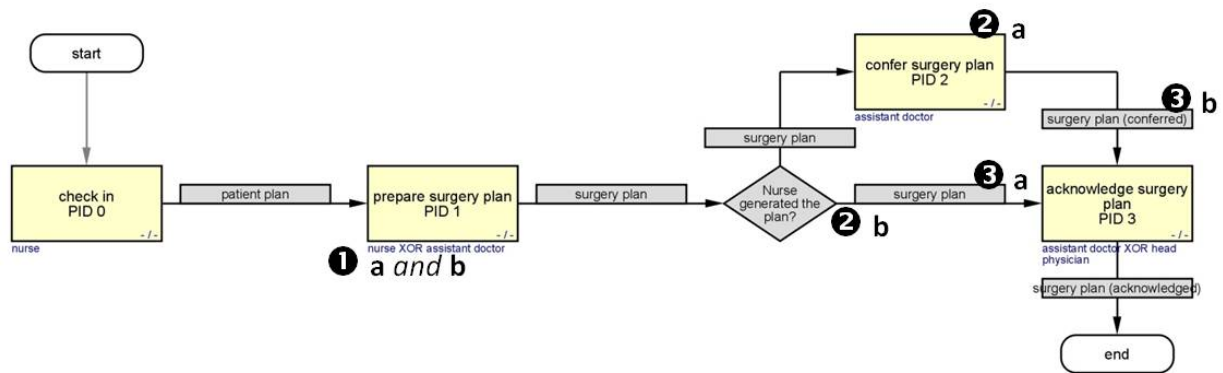


Figure 1. example POPM-model

For our project we decided to use the Perspective Oriented Process Modelling (POPM) framework (Jablonski, 1996) which covers all aspects of process in a modularized manner. An example of a process model can be seen in Figure 1, which already integrates two different variants. The five main perspectives of POPM are: *functional perspective* in order to identify the activities, which is strongly related to the *behavioural perspective* defining the order of the process steps; the *data perspective* to define the input and output of the process step and finally *the organization and operational perspective* in order to identify by whom, and by means of which tools or systems, the activities are executed.

Four so called variation points, which make the differences between the two integrated variant are informally highlighted by numbers. Even although this is only a small example, the main disadvantage is evident: Due to the number of variation points and the dependencies (they are not indicated Figure 1) between the variant options the user can easily be overwhelmed by the quantity of decisions. Thus, there is a need for a data model which implements the variability and configurability. We developed a special data model (see also (Meerkamm, 2010)) which we would like to introduce in the following sub-section.

### 2.1 Generic data model

The basic elements of the generic data model are nodes and leaves. We use the example from Figure 1 to illustrate our concept.

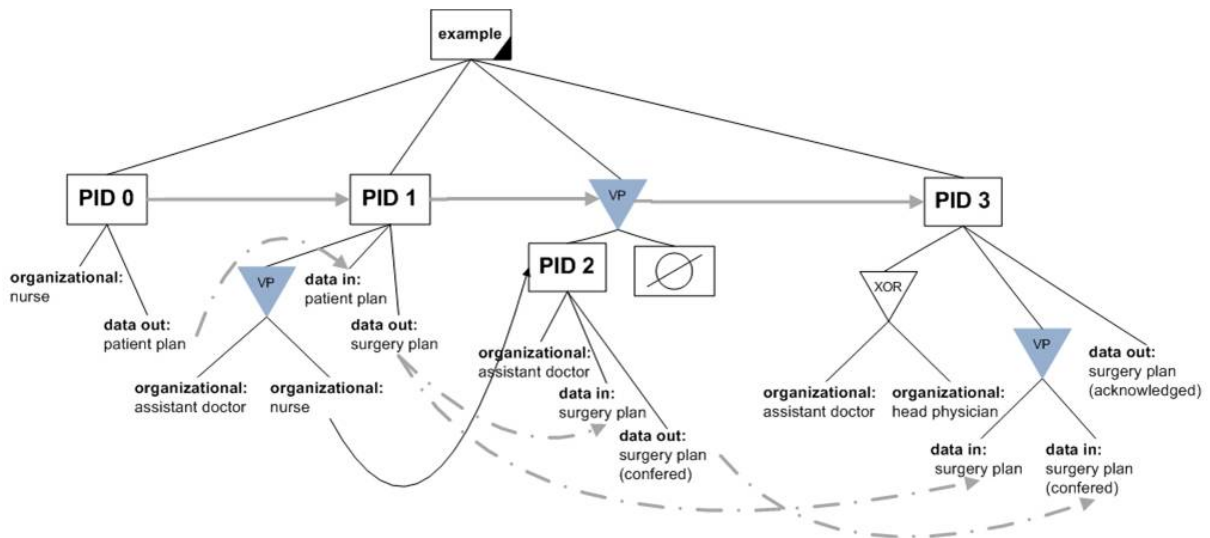


Figure 2. mereological graph

Processes are symbolized as nodes. In case of *elementary processes* they are not further decomposable (Figure 2, PID 0-3). *Composite processes* demonstrate that several processes are composed into a higher level. The composition can be done repeatedly resulting in several hierarchical levels within the tree. The root of the tree always has to be a composite process. We further define a *control flow* between the processes. As process can be more complex, as in our example, an explicit control flow is introduced which is demonstrated by the grey dashed arrows between the process steps. *Alternatives (OR) and exclusive alternatives (XOR)*, modelled as nodes, aggregate several process elements. In order to distinguish between optional and mandatory alternatives an “Empty Element” is introduced, illustrated by the crossed circle within a process. *Conjunctions (AND)* symbolize the parallel execution of the processes.

This concept leads to a functional structure of our configuration concept – which is a common method of handling the complexity of a configurable object (Mittal & Fraymann, 1989; Riitahuhta, 2001; Rosemann & van der Aalst, 2007).

We want to enlarge the concept with *aspect oriented nodes*. The elementary process is connected with this additional sort of nodes or rather leaves. The most common aspects are the organizational, operational and data oriented aspects (Blecker, Abdelkafi, Kreuter, & Friedrich, 2004; Curtis, Kellner, & Over, 1992; Scheer, 2000). For each aspect the relevant types are defined, e.g. nurse as organizational aspect for process step 0. AND, OR and XOR illustrated logical conjunctions and alternatives in terms of the types.

The *data-oriented* nodes distinguish between input and output data of a process step. In order to know exactly which output data is consumed by the next process step as input data an explicit *data flow* is defined. This is for example demonstrated by the grey dashed and pointed arrow between output data element “patient plan” of process step 0 and input data element “patient plan” of process step 1.

Last but not least we define the *variation points*, the coloured triangles named VP in our example. The attached nodes are the possible variant options. The variation points clearly have to be separated from a normal XOR and OR. XOR and OR is part of the behavioural aspect of the process model, thus only functional nodes can be attached.

In order to restrict the domain of possible variants and to ensure that the configured process finally realizes its function *implications* are necessary. These specify dependencies between and within process elements and elements of the aspects and capture the consequences of the variant option. For example, process step 1 implicates the execution of process step 2 in case the nurse is executing process step 1, which is modelled by a thin black arrow.

This generic model captures the configurable aspects of the processes together with related choices and consequences. This empowers the derivation of variants. However, the model bears limitations: Existing process models have to be transferred to the tree. The manual execution of this task is both time-consuming and error-prone, thus it is recommended that the transformation is automated. In this case there will be adequate validation of the correctness of the transformation programme which is much more efficient. The validation of the initially modelled tree itself is, of course, also necessary.

## 2.2 Variant List

As integrated process models are complex and the stakeholders cannot be modelling experts themselves the process must be made to give them access to the information they require (Hammer & Champy, 1993; Recker, Rosemann, & von Uthmann, 2000; Reijers, Mans, & Toorn, 2009). Thus, after the definitions of the integrated process model, including all variants and variation points, the latter as well as the attached variant options have to be organized accordingly; this is documented in a so called “variant list”.

The integrated process model from our example in Figure 1 contains two variants: (1) “check in with conferred surgery plan” and (2) “check in”. The procedure is as follows:

1. A domain expert has *to decide which variation points* are relevant for the concrete variant.

For the variant “check in”, we selected for example the three variation points concerning the organizational, functional and data-perspective, as it can be seen in Figure 2.

2. Afterwards the domain expert has *to define which variant options* of the selected variation points are relevant for the concrete variant.

In our example we selected “assistant doctor” for the organizational variation point, skipping PID 2 and with this, as there is a dependency, the data “surgery plan”.

The same procedure is followed for the variant “check in with conferred surgery plan”. In generating the “variant list” the variants are listed, including the relevant variant options. The process owner only has to select one of the variants from the variant list. In order to facilitate the decision some additional information is supposed to be given with each variant. Afterwards, the process model is adapted, viz. configured, in the “background” as we will also see later presenting the case study. The results are then presented to the process owner.

The advantage of this concept is as follows: Instead of deciding several variation points separately and step by step (here we have only three variation points, each with two variant options; but we have examples which are much more comprehensive) now the process owner is only confronted with the selection of the variant itself from the presented list. This reduces the number of decisions and the process owner does not have to consider all the dependencies which can occur between the different variation points and their variant options.

## 3 Generation of the Different Process Models

As already indicated, we had two projects at our chair, one in the medical domain, the other one within the administration at our university affecting the so called “student life-cycle”. On the basis of the generated process models we want to test our variability concept which was explained in 2.1. The whole procedure is explained in this Section.

**Procedure (i) – Data collection:** In order to generate the process models, which could then be used for our test, we visited the hospital and the administration of our university. In both cases we organized meetings with the relevant people in order that they could explain to us the manner in which they carry out their activities. This phase of data collection was done by means of qualitative analysis methodology (Lamnek, 1995; Yin, 2003) in the form of guided interviews (Witzel, 2000). In our

context the guide line arises from the perspective oriented modelling language we were using as may be seen in the introduction to Section 2.

We finally modelled the collected data with the modelling tool i>pm (ProDatO, 2005). Each of these processes comprises about 150-200 process steps. In order to make sure that we correctly depicted the processes, at the end of this phase we asked the process owner to validate the process models. The comments were discussed and integrated into the process models accordingly.

**Procedure (ii) – Identifying the variants and integration:** We then explicitly identified the variants including their variation points and variant options in each of the process models. Some variants were already accidentally modelled as an integrated model; the other variants are modelled separately, in some cases even at different positions in the process models, viz. not at the same “process sheet”.

From each project we selected three use cases (viz. three sets of variants) in order to test our configurability concept. In total we created six redesigned integrated i>pm process models that incorporate all of the variants. We selected the following three medical processes:

- (1) *Admission procedure:* This process is executed by a doctor to allow people to enter the hospital.
- (2) *Discharge procedure:* This process is executed to officially allow a person who was ill at the hospital to go home.
- (3) *Issuing an invoice:* This process includes all of the steps which are necessary to list the various work and treatments carried out during the patients stay in hospital.

An example from the medical domain is shown in Figure 3 with two variants modelled separately, and Figure 4 illustrating the corresponding integrated process model from these two variants. In our consideration of the “student-life-cycle” we created and/or selected for the following three processes:

- (1) *Preparation for the resolution procedure:* This process illustrates the tasks which are necessary to prepare the resolution procedure at the different hierarchy levels of the university in order to introduce a degree program.
- (2) *Application procedure of the students:* This process illustrates how the different types of students (national/ international, graduate/ undergraduate ...) have to apply for a place at the university in cases where no free enrolment is possible.
- (3) *Examination procedure:* This process illustrates the organization and accomplishment of oral or written tests during the period of the students study.

Note that there are more variants in both process models, but for this case study we focused on this sample, as they are the simplest ones.

**Procedure (iii) Transfer:** We transferred each of the i>pm process models to the generic data model, the mereological graph model. For the time being the transfer from i>pm to the graph was done manually as no implementation exist. Beside this, we were able to model some additional information, which we gained during our interviews and which is available only in a descriptive and informal way in the i>pm models, onto the graph. An example is the dependencies between the different variant options.

**Procedure (iv) Preparation of the configuration:** The resulting graph models integrating explicitly all variation points, variant options and the dependencies were of course far too complex to be used for configuration by the stakeholders, whether it be the staff from the hospital or the staff from the university. Thus, we created the variant list, introduced in 2.2. By means of this, the configuration of the process model could be done by the stakeholders simply selecting one of the desired variants.

**Procedure (v) Configuration:** Based on the selected variants, for each variant we resolved all the related variation points by selecting the appropriate variant options and deleting these ones which are

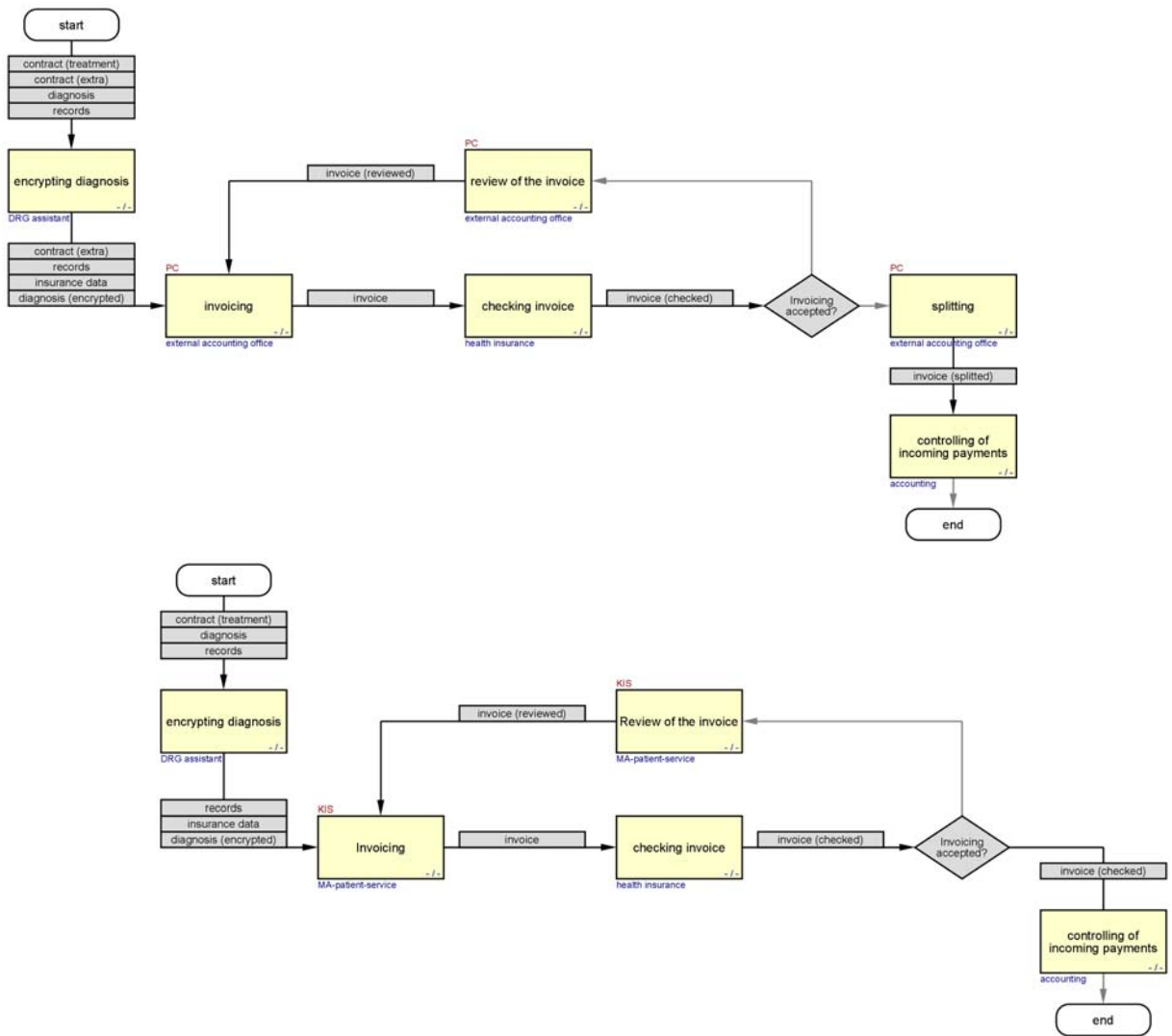


Figure 3. Two process variants of the process "issuing an invoice" (medical domain)

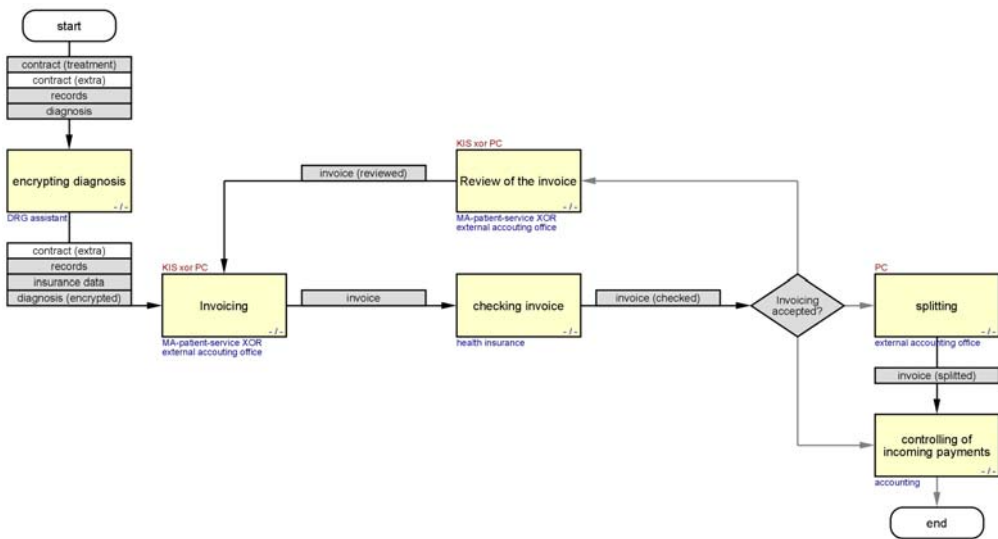


Figure 4. Integrated process model "issuing an invoice"



no longer relevant. After testing the correctness of the resulting graph models, they were mapped to the i>pm models and again checked for correctness.

The advantage of this configuration on the “background” is that the stakeholders do not need to be confronted with the integrated process model and the modelling language; there is no need for the stakeholders to understand the whole configuration process. The stakeholders receive only the individual process models. Thus the process owner directly benefit from the configurable process models.

## 4 Observation

In this Section report our experiences and challenges (including how we addressed them) of the two case studies. We categorise the observations according to the phase of the case studies.

**A. – General results** - First of all it should be mentioned that for all six processes from the two different domains, we were able to create integrated process models from the initial single ones and to generate “variants list” which allows the derivation of the individual process models. For each process of the two different projects we were able to generate a process models equivalent to the initial i>pm process model. This was done by configuring the special graph model according to the selection of the variant list; the resulting graph model was re-mapped to an i>pm process model which was then presented to the stakeholders. This demonstrates that it is possible to integrate several process variants such that afterwards the desired individual variant can be derived from it.

**B. – Challenges during integration of the models** - The generation of the integrated process models involved considerable effort. The identification of the variants was not the problem, but rather the integration of them into a single model. The matching of apparently identical process elements (in particular in terms of process steps and data elements), was only possible after long and deep discussion and reflection. This was necessary due to differences in meaning or the relationship with other process elements. The loss of information by matching elements together had to be carefully avoided. The question of the qualitative aspects of different data elements was a source of much discussion. Furthermore, we had to be particularly careful not forget aspects or process elements from one of the individual process models and this was especially true of the control flow. We had to carefully check if each individual single process could still be “executed” by using the integrated process model.

**C. - Challenge during the transfer** - The manual transfer from the i>pm model to the graph model was time consuming and there was a high risk of making mistakes; thus an automated solution would be preferable. In doing so the validation of the correctness of the transformation program would be sufficient.

Due to the graph structure the mapping of the *control flow* was not easy. This was especially the case when connectors or loops are part of the control flow. Furthermore, we had to be especially careful not to forget a control flow connection which could lead to dead ends in a process. The validation of the correctness of the modelled control flow for the précis runtime routing was even trickier. By re-playing the different initial processes in the integrated model we tried to identify the wrong connections and missing control flows.

For the special modelling element *empty element*, which can be modelled in the generic data model, no corresponding modelling element in the i>pm process model exist; thus we had to derive it from the general context. The manually integration of these information was very time consuming and error-prone due to possible misunderstandings. This also holds for *dependencies*, as dependencies are also not formally modelled in the i>pm model. We carefully re-checked the generic data model with the data from our interviews to make sure that they illustrated the correct circumstances.

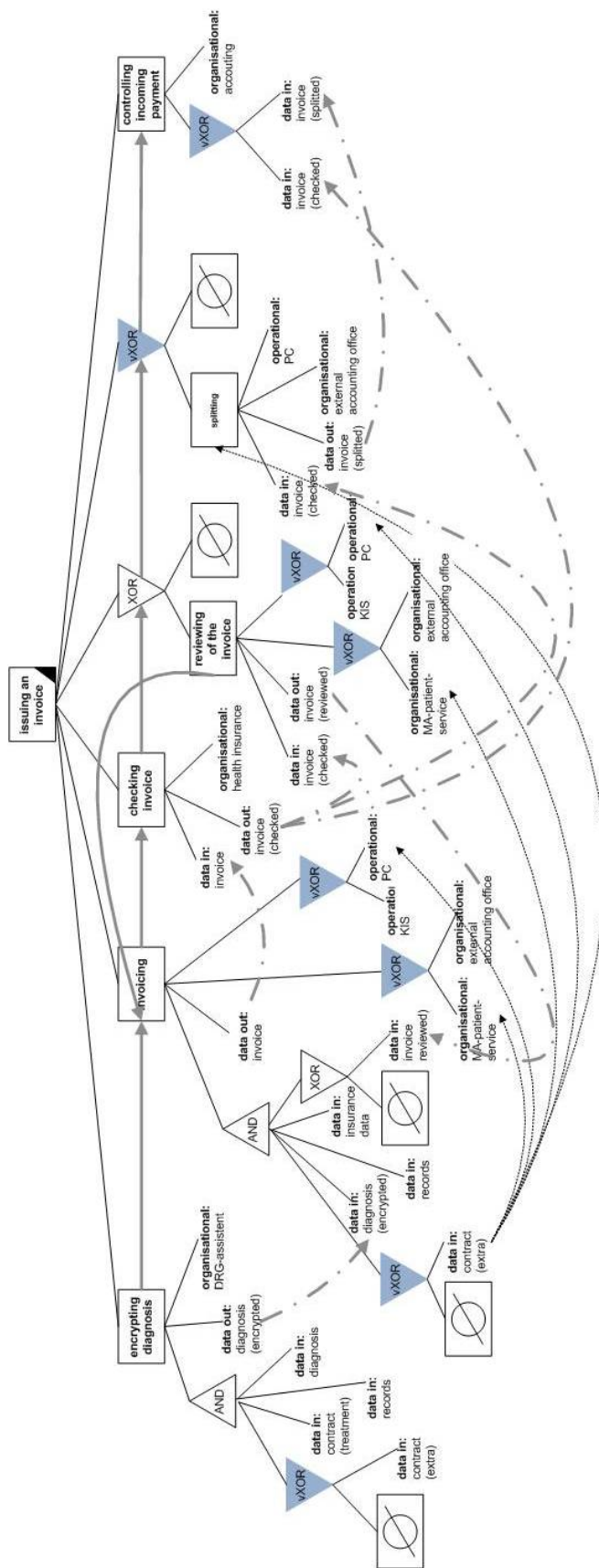


Figure 5. integrated process model II

**D. - Challenge by generating the variant list** – As already stated, the identification of the variants and with this the identification of the variation points and variant options was not a problem, neither was the generation of the variant list. It could simply be seen as a kind of documentation.

**E. - Challenge during configuration** - As the configuration was done manually the main challenge was the correct adaption and reduction of the generic data model and the same holds for the re-transfer to the i>pm process model. Several cycles were carried out, in order to validate the resulting i>pm process models, thus it was also a very time-consuming phase.

The whole procedure was mainly accomplished by the core project member of our chair, and took, including the development of the concept of the configurable generic data model, about nine months.

## 5 Related Work

Process variant management has a high relevance in theory and practice and has been widely studied in different research projects. In this Section we want to give a short overview (which we make no claim to be complete) of existing approaches, including a critical assessment of them from our own point of view.

Explicit **reference modelling and an adaption mechanism** have been introduced by (J. Becker, Delfmann, & Knackstedt, 2007). At the beginning an integrated, multi-perspective process model, a reference model, is generated. Logical terms and attributes are linked to the process elements in order to highlight which elements can be removed in the context of a certain scenario. According to a special context this reference model is configured, viz. unneeded process elements are deleted. The particular from this approach is that they define a staged adaption process. This means that the configuration process is performed repeatedly at several stages. The advantage is that at each stage the process model can become more concrete and the decision for a variant can be done incrementally.

By means of configurable event process chains (**C-EPC**) (LaRosa, 2008; LaRosa et al., 2007; Rosemann & van der Aalst, 2007) a single model approach is implemented. Initially, a whole processes family is generated. The resulting model, called reference model, can be customized to meet specific needs. For this EPC functions and decision nodes are annotated to indicate whether they are mandatory or optional; the definition of constraints is also possible. One of the advantages is the fact that it also focuses on multi-perspectives. Furthermore, the additional questionnaire-driven approach for the configuration process abstract away from the modelling language. This is also very useful as C-EPCs are rather difficult to understand, especially for the end-user with a non-technical background.

The **Provop** (Hallerbach et al., 2008, 2009) approach collects all process variants in a single process model. It applies well-defined change operations (modify, delete, insert) at so called “adjustment points” which are explicitly identified in the basic process model. As it also includes non-functional aspects, it constitutes a very comprehensive approach. Nevertheless, the “change” operations are defined as separate objects with respect to the process model and the objects have to be connected to the model by additional modelling constructs (the “adjustment points”). In doing so the variability is made explicit but outside of the model, which enhances the complexity of the approach.

Process variants are also relevant in the domain of software engineering (M. Becker, Geyer, Gilbert, & Becker, 2001; Halmans & Pohl, 2003). **PESOA** (Bayer et al., 2005; Puhmann, Schnieders, Weiland, & Weske, 2005) provides a concept for variant modelling using UML. Different variability mechanism as inheritance, parameterization and extension points are used during modelling of variants. In doing so there is no separation between the basic process (as source of the variants) and the variants themselves. It is not possible to describe the variants in a uniform manner due to the different possible variability techniques. It focuses on UML, viz. the approach is language-dependent and, furthermore, it does not implement a multi-perspective view of the processes.

## 6 Conclusion

In this cases study we developed configurable process models for a total of six processes from two different domains: the medical domain and the administrative domain. In consideration of the results, the developed technique proved to be suitable for the intended purpose. We were able to derive all the initial variants from the integrated process model by using the generated “variant list” and applying the configuration procedure to the generic data model. However, even though the adaption of the processes was simplified, the creation of the configurable process models involved a great deal of effort as well as necessitating a high degree of domain knowledge and modelling experience. It is obvious that some issues that arose could have been avoided by tool support. This holds mainly for the model transformation. The capturing of dependencies between variation points is still not clear; the offered graphical solution is very confusing. Currently no implementation of the generic data model exists and, while we have begun work on an implementation, this is not yet in presentable form.

## References

- Bayer, J., Buhl, W., Giese, C., Lehner, T., Ocampo, A., Puhmann, F., et al. (2005). *PESOA - Process Family Engineering - Modeling variant-rich processes*. Potsdam, Germany: Hasso-Plattner-Institut.
- Becker, J., Delfmann, P., & Knackstedt, R. (2007). Adaptive Reference Modeling: Integrating Configurative and Generic Adaption Techniques for Information Models In J. Becker & P. Delfmann (Eds.), *Reference Modeling* (pp. 27-58). Berlin: Springer Verlag.
- Becker, M., Geyer, L., Gilbert, A., & Becker, K. (2001). Comprehensive Variability Modeling to Facilitate Efficient Variability Treatment, *4th Int. Workshop of Product Family Engineering (PFE 2001)*. Bilbao, Spain.
- Blecker, T., Abdelkafi, N., Kreuter, G., & Friedrich, G. (2004). Product Configuration Systems: State-of-the-Art, Conceptualization and Extensions *Génie logiciel & Intelligence artificielle. 8th Magrebian Conference on Software Engineering and Artificial Intelligence (MCSEAI2004)*). Sousse, Tunisia.
- Curran, T., & Keller, G. (1997). *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model* New Jersey, USA: Upper Saddle River.
- Curtis, B., Kellner, M. I., & Over, J. (1992). Process Modeling. *Communications of the ACM*, 35(9), 138-146.
- Fettke, P., & Loos, P. (2004). Referenzmodellierungsforschung *Wirtschaftsinformatik*, 46, 331-340.
- Fettke, P., Loos, P., & Zwicker, J. (2005). Business Process Reference Models: Survey and Classification *Workshop on Business Process Reference Models (BPRM'05)*. Nancy, France.
- Fettke, P., & Loose, P. (2003). Classification of reference models: a methodology and its application *Information Systems and e-Business Management*, 1, 35-53.
- Frank, U. (1999). Conceptual Modeling as the Core of the Information System Discipline - Perspectives and Epistemological Challenges *Fifth America's Conference on Information Systems (AMCIS 99)*. Milwaukee, USA.
- Hallerbach, A., Bauer, T., & Reichert, M. (2008). Managing Process Variants in the Process Life Cycle, *10th Int'l Conf. on Enterprise Information Systems (ICEIS'08)*. Barcelona Spain.
- Hallerbach, A., Bauer, T., & Reichert, M. (2009). Capturing Variability in Business Process Models: The Provop Approach. *Journal of Software Maintenance and Evolution: Research and Practice*, 22(6-7), 519-546.
- Halmans, H., & Pohl, K. (2003). Communicating the variability of software-product-family to customer. *Software and System Modeling*, 2(1), 15-36.
- Hammer, M., & Champy, J. (1993). *Reengineering the Cooperation: A Manifesto for Business Revolution*. London: Nicholas Brealy Publishing.
- Jablonski, S., Bussler, C. (1996). *Workflow Management – Modeling Concepts, Architecture and Implementation*. London: International Thomson Computer Press.

- Lamnek, S. (1995). *Qualitative Sozialforschung* (Vol. 1). Weinheim: Springer Verlag.
- LaRosa, M. (2008). *Managing variability in process-aware information systems*. PhD. Queensland University of Technology, Brisbane.
- LaRosa, M., & Dumas, M. (2008). Configurable Process Models: How to Adopt Standard Practices in Your Own Way? *BP trends*, 2008(11), 1-10.
- LaRosa, M., Lux, M., Seidel, S., Dumas, M., & Hofstede ter, A. (2007). Questionnaire driven Configuration of References Models, *19th International Conference on Advanced Information Systems Engineering (CAiSE'07)*. Trondheim, Norway.
- Meerkamm, S. (2010). Configuration of Multi-Perspectives Variants, *1st. Int'l. Workshop on Reuse in Business Process Management (rBPM 2010)*. Hoboken, NJ, USA.
- Mittal, S., & Fraymann, F. (1989). Towards a Generic Framework of Configuration Task *11th International Conference Artificial Intelligence (IJCAI'89)*. San Francisco, USA.
- OGC. (2010). *IT Service Management (ITIL) Home Page*. Retrieved 01-11, 2010, from <http://www.ital-officialsite.com/home/home.asp>
- ProDatO. (2005). Handbuch iPM Integrated Process Manager, Softwaredocumentation (in German). Erlangen, Germany
- Puhlmann, F., Schnieders, A., Weiland, J., & Weske, M. (2005). *PESOA - Variability Mechanism for Process Models* Potsdam, Germany: Hasso-Plattner-Institut.
- Recker, J., Rosemann, M., & von Uthmann, C. (2000). Guidelines of business process modeling In W. M. P. van de Aalst, J. Desel & A. Oberweis (Eds.), *Process Management. Models, Techniques and Empirical Studies* (pp. 30-49). Berlin: Springer Verlag.
- Reijers, H. A., Mans, R. S., & Toorn, R. A. (2009). Improved model management with aggregated business models *Data & Knowledge Engineering*, 68(2), 221-243.
- Riitahuhta, A. (2001). Views and Experiences of Configuration Management. In A. Riitahuhta & A. Pulkinen (Eds.), *Design for Configuration*. Berlin: Springer Verlag.
- Rosemann, M., & Schütte, R. (1997). *Grundsätze ordnungsmäßiger Referenzmodellierung*. Wiesbaden, Germany: Dr. Th. Gabler Verlag.
- Rosemann, M., & van der Aalst, W. M. P. (2007). A Configurable Reference Modeling Language. *Information Systems*, 32(1), 1-23.
- Sabin, D., & Weigel, R. (1997). Product Configuration Frameworks-A Survey. *IEEE Intelligent Systems*, 13(4), 42-49.
- Scheer, A.-W. (2000). *ARIS - Business Process Modeling* (3rd ed.). Berlin, Germany: Springer Verlag.
- Silverston, L. (2001a). *The Data Model Ressource Book, Vol. 1: A Library of Universal Data Models for All Enterprises, revised Edition*. USA: Wiley.
- Silverston, L. (2001b). *The Data Model Ressource Book, Vol. 2: A Library of Data Models for Specific Industries, revised edition*. USA: Wiley.
- Stephens, S. (2001). The Supply Chain Council and the Supply Chain Operations Reference Model. *Supply Chain Management - An International Journal*, 1(1), 9-13.
- Thomas, O. (2006). Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation *Workshop Workshop on Business Process Reference Models Business Process Management (BPRM'05)*. Nancy, France.
- Tiihonen, J., & Soininen, T. (1997). *Product Configurators - Information System Support for Configurable Products*: Hewson Consulting Group.
- van der Aalst, W. M. P., Dreiling, A., Rosemann, M., & Jansen-Vullers, M. H. (2005). Configurable Process Models as a Basis for Reference Modeling *Business Process Management Workshops (BPM'05)*. Nancy, France.
- Winter, R., & Schelp, J. (2006). Reference Modeling and Method Construction - A Design Science Perspective *21st Annual ACM Symposium on Applied Computing (SAC2006)*. Dijon, France.
- Witzel, A. (2000). Das problemzentrierte Interview. *Forum Qualitative Sozialforschung*, 1(1).
- Yin, R. K. (2003). *Case Study Research: Design and Methods*. California: Sage Publications.