

Configurable Process Models: How to Adopt Standard Practices in Your Own Way?

Marcello La Rosa and Marlon Dumas

“The thing can be many in one sense, but also can be one in another sense”

Al-Ghazali: *Revival of Religious Sciences.*

Configurable process models enable a systematic documentation and reuse of standardized “best” practices, while allowing process analysts to understand possible variations contemplated by these standards, and how to link these variations to business decisions. This article discusses the potential benefits of configurable process models and introduces a method and a toolset for process design based on configurable process models.

Mixing Oil and Water

To the experienced BPM practitioner, it is clear that commonalities are frequently found between business processes across business units within the same organization, across companies in a given industry, or even across industries. For example, we often hear the term *order-to-cash* being used to refer to a business process that starts from the moment a purchase order is received by a supplier, to the moment this purchase order has been fulfilled (and the supplier has received the corresponding payment). Virtually all order-to-cash processes include activities related to invoicing, delivery, and payment.

What's frustrating, however, is that, while all order-to-cash processes look the same, they end up being different. For example, an order-to-cash process for the delivery of goods (e.g., delivery of office supplies) is different from an order-to-cash process for the delivery of services (e.g., delivery of consultancy services). In the first case, there is often a physical delivery that happens at a discrete point in time, and the condition of the goods can be checked upon receipt. On the other hand, the delivery of a service may occur over a long period of time (say, 6 months). Over this period, several invoices may be issued for the same original purchase order. Also, evaluating the quality of a consultancy service is often trickier than checking the quality of a box of reams of paper. Not surprisingly, the corresponding order-to-cash process models will have many differences.

Despite such differences, companies have a lot to learn from each other when it comes to analyzing and re-designing their order-to-cash processes. It would be inefficient if every time a company engages in modeling and re-designing its order-to-cash process, it did so “from scratch,” without consideration for how other companies perform their order-to-cash process. Yet, “from scratch” process modeling initiatives are more frequent than one would like to think, and the statement “we are unique” is often used as an argument to avoid adopting standard practices.

Reference process models driven by industry consortia, such as SCOR¹, VICS² or ITIL³, go a long way into identifying common processes, activities, and performance metrics in their respective industries, while APQC's Process Classification Framework⁴ provides an extensive cross-industry classification of processes and activities. In a similar vein, but outside the framework of industry consortia, SAP's reference model [1] captures recurrent processes

¹ Supply Chain Operations Reference Model: www.supply-chain.org

² Voluntary Inter-industry Commerce Solutions: www.vics.org

³ Information Technology Infrastructure Library: www.itil-officialsite.com

⁴ American Productivity and Quality Center: www.apqc.org

supported by SAP's platforms. Any BPM practitioner should consider having these reference process models in his/her toolkit. In fact, tool vendors have already seized this opportunity by integrating popular reference process models in their product portfolio. However, their offerings do not go beyond pre-populated repositories of process models, and provide little guidance and tool support to help analysts to adapt these process models to specific needs and contexts. The fundamental problem is that, beyond independent initiatives aimed at creating reference process models in various industries, there is a lack of standard notations and methods for modeling reference processes and for enabling systematic reuse of reference process models in BPM projects.

One key factor preventing a more widespread adoption and systematic reuse of reference process models is the inherent trade-off between standardization on the one hand, and variation and differentiation on the other. On the one hand, process standardization is desirable because it enables the emergence of best-in-kind designs, it leads to uniform interfaces for customers and business partners, and it helps in creating synergies and economies through sharing of business improvements, resources, and IT assets. It also simplifies training of process workers, and facilitates their re-deployment across business units. On the other hand, different business units (and, of course, different companies) have different needs and priorities, and these needs and priorities evolve in different directions over time. Also, standardization across business units operating in different regions is often hindered by specific requirements, such as a credit check being required in some regions while not in others, or the said credit check being performed in different ways depending on the region. Finally, standardization across companies operating in the same industry often clashes with the imperative for competitive differentiation. Thus, what is needed is a framework that enables the standardization of processes while at the same time enabling variations.

At first, this sounds like mixing oil and water. Yet, it is precisely what process configuration is about. Process configuration essentially deals with the following question: "How to model business processes that are similar to one another in many ways, yet differ in some other ways from one organization, project, or industry to another?" This is achieved by capturing multiple variants of a process in a single *configurable process model*. A configurable process model captures a family of process models and allows process analysts to understand what these process models share, what are their differences, and why and how these differences occur. More importantly, a configurable process model provides a starting point to derive process models for specific companies, business units, or projects.

Configurable process models

To understand what a configurable process model is, let's consider an example. This example is extracted from a repository of process models for the screen business that we have documented in collaboration with several stakeholders in the film production industry. In this industry, it is difficult to come up with standardized process models because virtually every film production project has its own characteristics and, thus, its own business processes. Yet, these processes share many commonalities.

Figure 1, for example, shows two BPMN process models corresponding to two practices in screen post-production: *shooting on tape* and *shooting on film*. Although the two practices differ, some activities occur in both cases, a postproduction process always starts with the preparation of the footage for edit, and continues with an offline edit. This sequence of activities is always followed, regardless of the shooting medium. However, an online edit is carried out if the footage is tape, while a negmatching is carried out if the footage is film. Online edit is a cheap editing procedure that combines well with low-budget movies, typically shot on tape. Negmatching offers better quality results although requiring higher costs; therefore it is more suitable for high-budget productions shooting on film. The choice between online edit and negmatching represents a variability in the post production process: Depending on drivers such as budget, required level of creativity, and type of project, one option or the other needs to be taken.

The question is: How can we represent the post production process in a readable way, but without losing information about possible variants? One option is to build this process around a specific practice – say, Tape shooting (the most common one) – and to stick a note somewhere in the model stating that “when the project is shot on Film, replace activity Online with Negmatching.”

On a larger scale, this option is unsatisfactory: We don’t want a BPMN model full of annotations listing every possible variation, nor do we necessarily want to privilege one practice over another. This is where *configurable process models* come into play. The right-hand side of Figure 1 shows a configurable process model for post-production. This model is a merger between the two process variants. In a configurable process model, variation points are represented by means of *configurable gateways*. For example, the inclusive split gateway marked in red in Figure 1 is a configurable gateway. Unlike a “normal” BPMN gateway, a configurable gateway does not represent a choice or a parallel split behavior that will have an effect when the process is executed or simulated. Instead, a configurable gateway represents a design choice that will need to be made by an analyst in order to adapt the configurable process model for a particular project or organization. For example, the configurable gateway in Figure 1 captures the fact that one needs to choose for a given film project whether to select one path (tape shooting) or the other (film shooting), or possibly both.

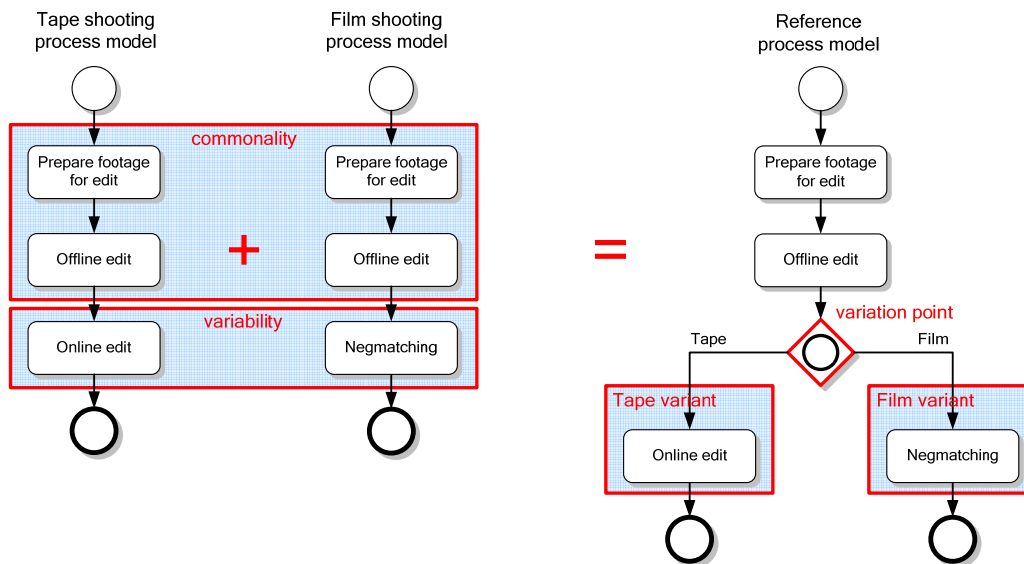


Figure 1. A configurable process model is an integrated representation of multiple process variants.

A configurable process model will typically feature many such variation points, each one capturing a decision that needs to be taken during process design. An analyst can *configure* such a model by picking the most suitable variant for each variation point. Once all these decisions have been taken, the configured process is *individualized* by removing variants that are no longer relevant, leading to an *individualized* process model. The individualized model can be used for further analysis, simulation, or to produce an executable model for a specific set of requirements. Thus, a configurable process model combines a family of similar process models together and is intended to be configured to fit the requirements of specific organizations or projects.

It’s time to make choices

Let’s come back to the difference between a configurable gateway and a traditional gateway. One can ask: Why can’t we simply use standard modeling constructs to capture configuration decisions? As hinted above, the decision associated with a configurable gateway is a design-time decision. It is not based on data values available at runtime (e.g., the amount of an expense in a procurement process), but, rather, on the requirements of the project or organization for which the

model is to be used. Drivers behind the configuration of a process can be the organizational culture, national or regional regulations, compliance requirements, cost, etc.

To exploit configurable process models in the process lifecycle, the traditional “process design” phase is split into two phases: one where the configurable reference model is designed, and another where the reference model is configured and individualized to fit a particular context. This *configuration and individualization* phase precedes the implementation phase, where the derived models are deployed, as shown in Figure 2.

Capturing variation points

The idea of configurable process modeling was introduced by Rosemann and van der Aalst [2], who proposed an extension of the EPC notation called Configurable EPC (C-EPC). In C-EPCs, any function or connector can be marked as being configurable.⁵ Graphically, a configurable element is distinguished by a thicker border than normal elements. A configurable function is a function that can be dropped from the model during configuration (with some restrictions). A configurable connector can be turned into either a different type of connector, or dropped altogether, in which case one of the outgoing/incoming paths is deleted.

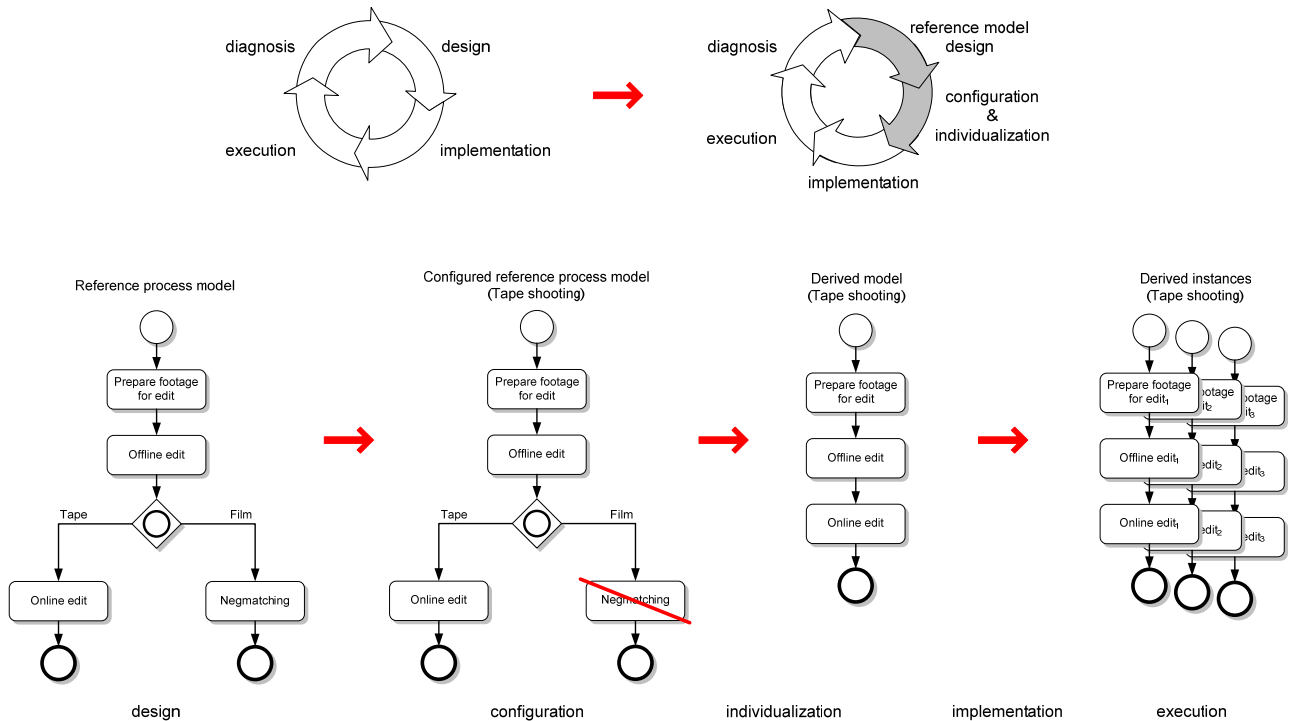


Figure 2. Configuration and individualization of process models in the process lifecycle.

Figure 3.a shows a refined version of the post production process in C-EPC. In this model there are 5 configurable OR connectors and 2 configurable functions. The connectors allow the selection of the medium to be prepared for the editing (which can be tape or film), the editing (online or negmatching), and the type of delivery (tape or film). The functions allow the conversion of media formats: E.g., if the project shoots on film but finishes on tape, we need to transfer the film-edited picture onto tape, via a so-called telecine transfer. In this case, we would need to keep function Telecine transfer and drop function Record Digital Film Master.

⁵ In the EPC terminology, a *function* corresponds to a BPMN task or activity, while a *connector* corresponds to a BPMN gateway.

In Figure 3 we can also see two different individualizations of this process that have been obtained through configuration: the process for low-budget productions (Figure 3.b) and the process for high-budget productions (Figure 3.c). In the first one, we shoot and release on tape and edit online; in the second one, we shoot on film, do a negmatching, and release on both film and tape.

We have also extended the idea behind C-EPC into an even richer language that enables the specification of variation points not only in the control logic of a process (i.e., which activities and process paths can vary), but also in the organizational resources and business objects that participate in the process [3]. For example, in low-budget productions the offline editing is typically carried out by the director and the picture editor, while in high-budget productions this may also involve the post-production supervisor and the producer. Moreover, this activity needs the temp picture from the shooting and produces the edited picture, which may be accompanied by editing notes. Therefore, these variations involving other aspects of a process should also be considered during configuration. For example, Figure 4 depicts an offline editing activity in this new extended language, called C-iEPC (Configurable integrated EPC). In this example, the roles, Supervisor and Producer, and the object, Editing notes, represent variation points – which are denoted by thicker borders for these elements.

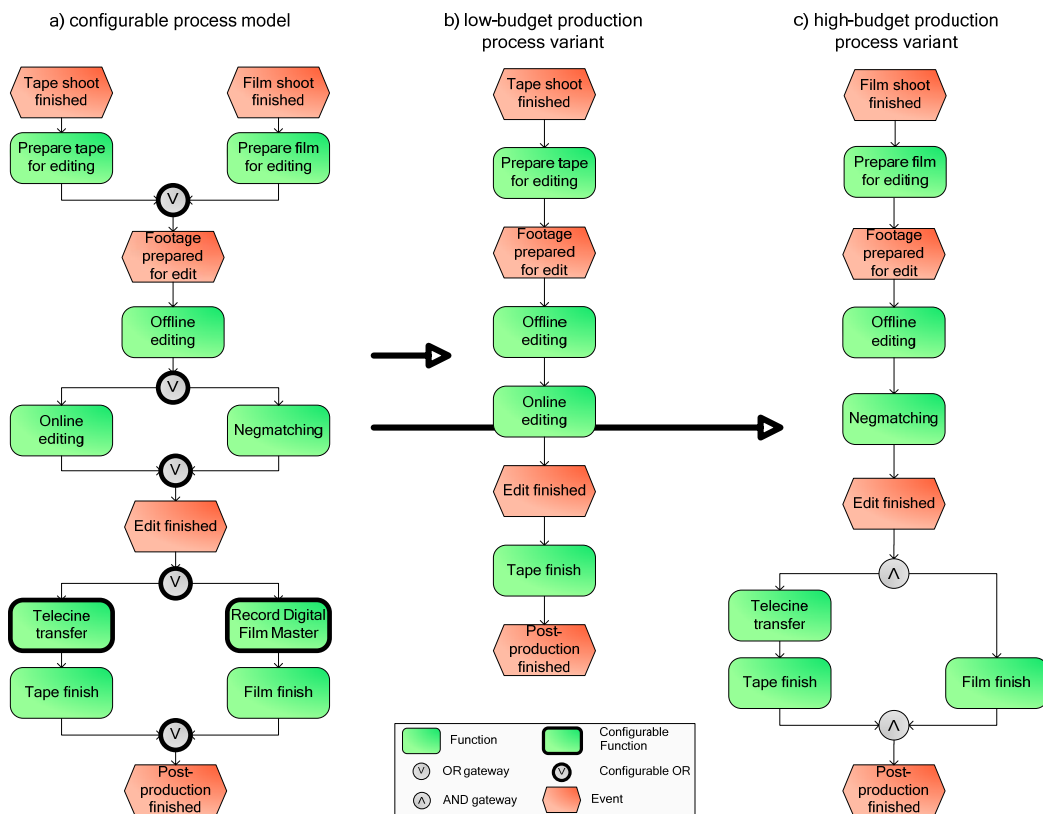


Figure 3. The postproduction configurable process model in C-EPC and two possible individualizations.

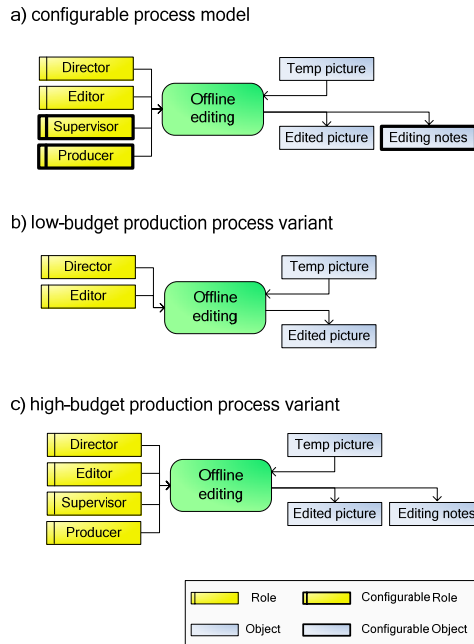


Figure 4. Process variants with human roles and business objects in the C-iEPC language.

Decision support for process model configuration

A configurable process model tells us “how the process varies” but not “why it varies.” Clearly, the variation points of Figure 3 cannot be set freely. As we said, we do a telecine transfer only if we shoot on tape, and similar interdependencies can be drawn for the other variation points. Indeed, constraints like this one depend on business parameters, such as budget and type of project. If, say, we decide to shoot on film, transfer in telecine, and release on tape, it’s because we are producing a project that demands these requirements; e.g., a TV program.

All the more so, if we modeled a real configuration scenario, such as the order fulfilment process from the VICS reference model, we would have to use numerous variation points with complex and intricate interdependencies. And this would very likely make the whole configuration process complex and error-prone.

So another aspect to consider is the provision of decision support during configuration. That is, “How can we abstract from the variation points of a process model and reason in terms of the domain concepts involved?” The idea is to represent these domain parameters in a separate artefact – a questionnaire model – and to link the latter to the variation points in the process model [4]. According to the answer given to a question, one or more variation points are automatically configured in the linked process model. For example, in postproduction, we can ask the question, “What is the type of budget?” with possible answers – “Low,” “Medium,” and “High” – and link these answers to the variation points of the C-EPC model of Figure 3, such that by simply answering “Low,” we would obtain the process variant for low budget productions (Figure 3.b). Other questions for postproduction can be related to the distribution channel (e.g., TV, cinema, home), the delivery medium (e.g. tape, film, DVD), etc. An extract of this questionnaire model is shown in Figure 5.

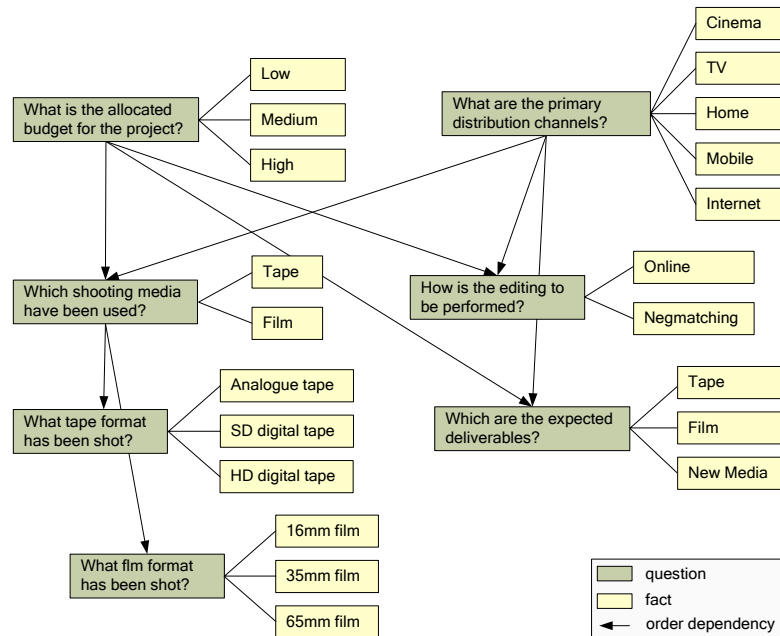


Figure 5. Extract of the questionnaire model for postproduction.

A questionnaire model essentially contains questions, their possible answers, order dependencies, and domain constraints. Order dependencies are used to constrain the order in which the questions should be posed, while domain constraints capture relationships between the possible answers to different questions. Domain constraints are determined by the business rules of the domain and are declarative in nature [5]. For post-production, examples of domain constraints include “negmatching can only be chosen if the project shoots on film” and “a delivery on film for cinema is not suitable for low budget productions.”

We have developed a toolset called Synergia to assist analysts during the configuration of process models defined in C-iEPC through the use of questionnaires. Quaestio – the front end of Synergia – generates an interactive questionnaire from a questionnaire model where questions are posed to the user according to the dependencies; answers that may lead to invalid configurations are avoided by automatically enforcing the domain constraints. For each question, the tool also provides guidelines and suggestions to aid the user in answering the questionnaire (these guidelines are extracted from the configurable process model itself). A screenshot of Quaestio is shown in Figure 6. Here the user is prompted with the question about the budget, but the option “Low” has been grayed out since it is not compatible with the previous answer given that indicated “Cinema” as distribution channel. Note that this questionnaire interface is automatically generated from the information contained in the questionnaire model and the configurable process model.

This approach based on the use of interactive questionnaires supports the analyst in deciding how the configurable process model should be individualized to meet specific requirements. Moreover, the abstraction from a specific modeling notation (e.g., C-EPC) can make the configuration of (complex) process models more accessible to a non-skilled audience. In fact, while it is normal to assume that the modellers who design the process model are familiar with the notation in question, it is less realistic to assume that those who provide input for configuring these models (e.g., a screen director or other domain expert) are sufficiently proficient with the process modeling notation.

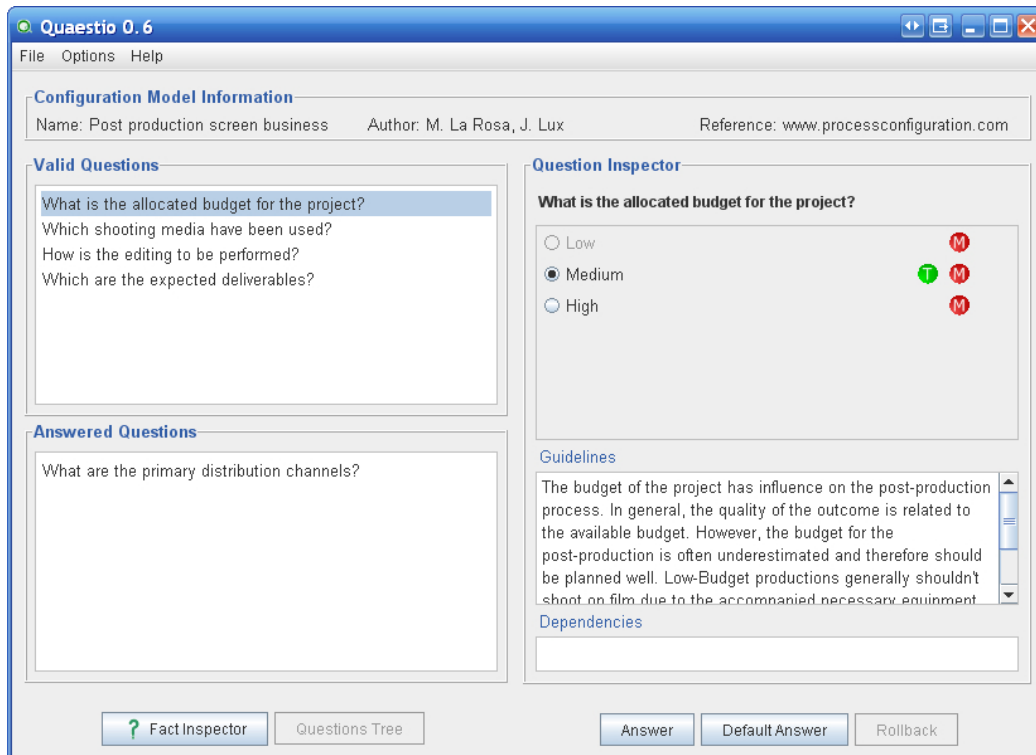


Figure 6. The Quaestio interactive questionnaire tool for process model configuration.

Configuring while preserving correctness

While a questionnaire-driven approach provides guidance to analysts during configuration, it doesn't guarantee that the individualized models are structurally and semantically correct. For example, if a model element or an entire path in a configurable process model is removed during individualization, the remaining model elements need to be re-connected to maintain a correct structure. Also, the configuration of variation points attached to parallel splits, decision points, and synchronization points in a configurable process model may lead to the introduction of semantic errors, e.g., deadlocks. And if the individualized process model contains such errors, it needs to be fixed manually. For example, by wrongly configuring the connectors of Figure 3, we could easily end up with a disconnected model like the one shown in Figure 7.

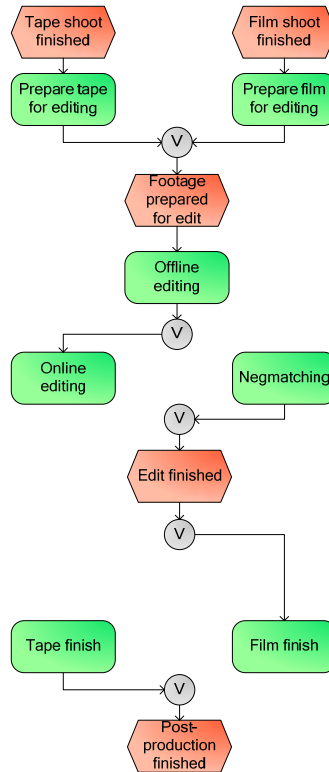


Figure 7. An example of incorrect individualization.

Therefore, the last aspect we have to consider when dealing with configurable process models is the correctness of the individualized models. In other words, “How can we guarantee our resulting process will be correct, given a correct configurable process model in the first place?”

We have devised a set of techniques [4, 6] to address this question as part of our questionnaire framework. In particular, we have developed a technique to automatically derive process constraints from a configurable process model that, if satisfied, guarantee the correctness of any resulting model. These process constraints are combined with the domain constraints of a questionnaire model such that whenever an answer is given to a question, a value is assigned to one or more variation points, and the unified set of constraints is evaluated. If the constraints are satisfied, the answer is accepted. If on the other hand the constraints are violated, additional variation points are identified that need to be configured simultaneously in order to preserve the process correctness. For example, if an edge in the process model is removed, all nodes in a path starting with that edge will also be removed.

By enforcing domain and process constraints, the framework guarantees that the individualized process model is correct and aligned to the requirements of the business domain.

Epilogue

To sum things up, configurable process models provide an appealing alternative to designing process models from scratch. They can be used as templates to derive solution-specific models while fostering the adoption of standardized or “best” practices within an organization. However, the field still suffers from a lack of notations and methods that would allow providers of reference models to document variations in their models (in addition to commonalities), and that would allow process analysts to understand which variants exist in a model, and how to select the variant that best fits a given set of business requirements.

We contend that the lack of such languages and methods is hampering wider adoption and systematic reuse of existing reference process models. For example, possible variations in ITIL reference models are captured in natural language, diluted in tons of heavy text, while SAP’s

reference process models are captured using plain EPCs – which do not allow one to represent variations in a systematic manner. As a result, the individualization of a reference process model is manual and error-prone. Analysts take these models merely as a source of inspiration, but ultimately, they design their own models, since they are given little guidance as to which model elements need to be removed or modified to address a given requirement. This limitation could potentially be addressed by explicitly capturing variations in these references models using configurable process models and linking them to questionnaire models.

To prove that the ideas behind configurable process models are more than wishful thinking, we have implemented and made freely available the Synergia toolset for designing configurable process models using the C-iEPC notation and the questionnaire framework discussed in this paper. This toolset and its associated documentation can be found at www.processconfiguration.com.

While this process configuration toolset is based on the C-iEPC notation, tool vendors will probably have no troubles in adapting the underlying concepts to BPMN or other notations. As a matter of fact, the idea of extending BPMN to capture process variants is already up in the air.⁶

References

- [1] T. Curran, G. Keller and A. Ladd. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*, Prentice Hall PTR, Upper Saddle River, 1997.
- [2] M. Rosemann and W.M.P. van der Aalst. A Configurable Reference Modeling Language. *Information Systems* 32(1): 1–23, March 2007.
- [3] M. La Rosa, M. Dumas, A. ter Hofstede, J. Mendling, F. Gottschalk, Beyond Control-Flow: Extending Business Process Configuration to Roles and Objects. In *Proceedings of the 27th International Conference on Conceptual Modeling (ER'2008)*, Springer-Verlag 2008.
- [4] M. La Rosa, F. Gottschalk, M. Dumas and W.M.P. van der Aalst. Linking Domain Models and Process Models for Reference Model Configuration. In *Proceedings of the Business Process Management Workshops*. Lecture Notes in Computer Science Vol. 4928, pp. 417–430, Springer-Verlag 2008.
- [5] R. G. Ross, Premise and Conclusion – Rules vs. Processes (Again). *BPTrends*, May 2008.
- [6] W. van der Aalst, M. Dumas, F. Gottschalk, A. ter Hofstede, M. La Rosa, J. Mendling, Correctness-Preserving Configuration of Business Process Models. In *Proceedings of the International Conference on Fundamental Approaches to Software Engineering (FASE'2008)*, Lecture Notes in Computer Science Vol. 4961, pp. 46–61, Springer-Verlag 2008.

Authors

Marcello La Rosa is a research assistant and PhD student in the BPM research group, Faculty of IT, Queensland University of Technology, Brisbane, Australia. He can be best contacted at m.larosa@qut.edu.au.

Marlon Dumas is a Professor at the Institute of Computer Science, University of Tartu, Estonia and part-time Associate Professor with the BPM research group at Queensland University of Technology, Brisbane, Australia. He can be best contacted at marlon.dumas@ut.ee.

⁶ <http://vishals.blogspot.com/2008/07/what-else-in-bpmn-20-process.html>