

**Complementing Business Process Verification by
Validity Analysis: a Theoretical and Empirical
Evaluation**

Research paper

**Revised and resubmitted to the Journal of Database Management
November, 2009**

Complementing Business Process Verification by Validity Analysis: a Theoretical and Empirical Evaluation

Abstract

Business processes are designed to achieve business goals. As business processes become increasingly automated through process aware information systems, the quality of process design becomes crucial. While verification of process models has gained much attention over the years, their validation has hardly been addressed. The paper suggests that validity criteria, which relate to the reachability of the process goal, should be used at process design. Since these criteria are intended to be used by human analysts, we propose to use them in combination with automated verification methods. This proposition is supported by a theoretical analysis which shows that the two approaches are complementary in nature, and by an empirical evaluation of the effectiveness of the validity criteria.

Introduction

As business processes become increasingly automated through process aware information systems, the quality of process design becomes crucial. In the life-cycle of a business process, designed process models can be transformed into executable process models (Zur-Muhlen & Rosemann, 2004). As is the case with artifacts in various domains (e.g., software, product, service), problems are easier and cheaper to fix at the early development phases than afterwards (Bray, 2002). Furthermore, unattended design flaws will result in an execution model which preserves the same flaws.

In the area of software engineering, quality assurance entails validation and verification. Validation, often referred to as “building the right system”, relates to whether the system meets the customer’s requirements, while verification, often referred to as “building the system right”, addresses the technical correctness of the system’s operation (Sommerville, 2007).

In analogy between software functional requirements and the goal of a business process, validation of a business process can relate to its ability to achieve its goal. However, most process modeling languages do not entail a goal construct. Rather, they mainly focus on

control-flow structures. As a consequence, the main focus of quality assurance in process modeling has been on verification of structural properties of process models.

The verified properties stand for the model's ability to be executed without reaching situations where the execution cannot complete (e.g., deadlocks, livelocks). Algorithms have been developed for verifying the existence of these properties in process models, usually related to specific modeling languages. Currently there is a variety of verification techniques which can automatically be applied to a designed process model. However, while these can be applied to a process model based solely on its structure, validation of the model requires the understanding of the business domain (Aalst, 2002; Sadiq et. al., 2004). Typically, a process model can be validated by domain experts through simulation (Aguilar-Saven, 2004). However, this requires the process to already be implemented in some simulation tool and does not support the early phase of design. At that phase, validation can only be accomplished as a human based task. Since, as mentioned, most process modeling languages do not entail a goal construct, no structured validation procedure is practiced, thus the task remains to the intuition and common sense of the human analyst. In many cases validation per se is ignored, and verification of control-flow properties is considered as sufficient for determining whether the quality of a process model is satisfactory.

Goal-oriented approaches to process design (e.g., the Generic Process Model – GPM (Soffer & Wand, 2004; Soffer & Wand, 2005)) entail criteria for goal reachability (also termed process validity) in a process model. However, these criteria are theoretical and abstract, and do not constitute a structured methodology to be followed. Furthermore, they are still not widely accepted in practice. The application of these criteria relates to the business logic of the process rather than to its structure. Currently, it is only based on human reasoning, not supported by automated algorithms.

This paper investigates the need for improving the current support to business process validation at design time. In particular, it investigates whether the commonly practiced verification needs to be complemented by validation based on goal reachability. As mentioned, validity criteria address goals, but can be applied by humans rather than in an automated manner. In contrast, verification methods can be performed automatically but without explicitly addressing goals. Hence, we propose to use the validity criteria while the process is being designed, and complement them with an automated verification of control flow properties.

We show that this combination is needed as follows. First, we theoretically analyze and compare the validity criteria and the verification-related properties, and show that they are

complementary rather than equivalent. Second, we empirically test the effect of applying the validity criteria and their contribution to a designed process.

As mentioned, verification methods are language-specific. Hence, our investigation should relate to a specific modeling language. To this end, we decided to use Event-driven Process Chains (EPC) for two main reasons. First, it is a highly popular modeling language used for process design. Second, there is a body of literature dealing with its formalization and verification, thus there are a number of approaches for verifying EPC models. EPC has evolved as a semi-formal language, whose formalization has been the subject of ongoing efforts over the years. Its syntax allows the modeler some degree of freedom, e.g., in deciding whether to explicitly represent external events or to “hide” them. The rationale for hiding external events is twofold: first, representing external events may result in overloaded models, and second, some of the verification methods entail hiding external events (Aalst, 1999). These different representation options may affect the way validity is assessed. Hence, the empirical study reported addressed two research questions. The main research question relates to the applicability and contribution of validity criteria to process design, and the secondary question relates to the effect of explicit process model representation, particularly when using validity criteria. Therefore, this study examines not only the need for validity analysis, but also the settings in which this can be accomplished effectively.

The paper is organized as follows: The next section provides details about GPM and its validity criteria, and about EPC verification methods. These two are compared, and their complementary nature is assessed. Afterwards we present the setting and findings of the empirical study that evaluates the validity criteria, and discuss the findings. Finally, conclusions and future research are presented.

Theoretical background

This section reviews different approaches for business process quality assurance, concentrating on the goal oriented GPM's validity criteria and on a set of methods for verification of specific model properties. Note that while these properties relate to possible behavior of the modeled process, we refer to them as being structural properties. The properties (e.g., soundness) are derived from some token-based semantics, not anchored in the specific domain which the model depicts. As a result, verification methods can be applied to a model whose elements are not even labeled to denote the specific real world elements they represent (an “empty” model). This is in contrast to GPM's validity assessment, which

can only be applied to a fully specified model, bearing the full information about the specific behavior of the modeled domain.

GPM and its validity criteria

This section introduces the GPM framework and its derived criteria for analyzing process validity. The presentation here is mostly informal, and relates to the main concepts and principles of GPM, whose formal definitions are given in (Soffer & Wand, 2004; Soffer & Wand, 2005).

GPM is a set of concepts which extends Bunge's ontology (Bunge, 1977; Bunge, 1979), as adapted for information systems modeling (e.g., Wand & Weber, 1990; Wand & Weber, 1995; Weber, 2004), and for incorporating business process related issues. It looks at a process defined over a *domain*, which is a *composite thing*, a part of the world of which we have control. The *state* of the domain is the set of values assigned to its *properties* at a moment in time. These properties are expressed as *state variables*. The state of the domain can be *stable* or *unstable*. An unstable state is a state that must change by *law*, and these state changes are termed *events*. A stable state is a state that can only change as a result of an event external to the domain. A sub-domain is defined by a subset of the domain state variables. Its state is a projection of the state of the domain, and it can be stable while other parts of the domain are unstable.

A *process* is a sequence of unstable states, transformed by law until a stable state is reached. The definition of a process over a domain sets the boundaries of what is in a stable or an unstable state.

A process model in GPM is a three-tuple $\langle L, I, G \rangle$, where L is the law, specified as mapping between subsets of states; I is a subset of unstable states, which are the possible initial states of the process after a triggering external event has occurred; G is a subset of stable states on which the process should terminate, namely, the goal of the process. Subsets of states are specified by conditions or predicates over values of the state variables of the domain. Hence, a process starts when a certain condition on the state of the domain holds, and ends when its goal is reached, i.e., when another condition specified on the state of the domain holds. As an example, a production process starts in a state where an order is given and all the resources are available, and ends in a state where the product is in finished goods inventory.

We briefly summarize this informal presentation by some formal notation.

Let $(x_1, x_2 \dots x_n)$ be the state variables representing the process domain, $C_1(x_1, x_2 \dots x_n)$ and $C_2(x_1, x_2 \dots x_n)$ be predicates, and $S_1 = \{s | C_1(x_1, x_2 \dots x_n) = \text{TRUE}\}$, $S_2 = \{s | C_2(x_1, x_2 \dots x_n) = \text{TRUE}\}$

sets of states of the domain. Then the law $L:S_1 \rightarrow S_2$ is a mapping, which can also be specified as an operator $L(s_1)=s_2$, where $s_1 \in S_1$, and $s_2 \in S_2$. Given a predicate $C_G(x_1, x_2 \dots x_n)$, which specifies the business condition for process termination, $G=\{s \mid C_G(x_1, x_2 \dots x_n)=\text{TRUE}; L(s)=s\}$.

GPM's goal orientation is the basis for its validity analysis, presented in (Soffer & Wand, 2004; Soffer & Wand, 2007), where validity is considered as goal reachability. A process model is termed valid iff every process path leads to a goal state. Three types of problems are identified as sources of process invalidity, and establish the criteria for validity assessment:

(1) Incompleteness of the process definition: A process definition is considered complete iff the law is defined for every combination of state variable values that may be reached from process states by law or by external events.

Formally: Let S be the set of possible states in a process. The process definition is incomplete iff $\exists s \in S$, such that $\neg \exists L(s)$.

An incomplete law definition might lead to a state where the process does not have a defined path by which to proceed and reach its goal. For example, consider a request that needs to be approved by two managers, and assume that the law is specified for the cases where both managers approve the request or reject it, but not for the case where one approves the request and the other rejects it. Completeness criteria are (a) completeness with respect to internal events, and (b) completeness with respect to external events. The analysis of completeness with respect to internal events should establish that the initial set of states at every step is reached as a final set of states at a previous step. Completeness analysis with respect to external events relates to a set of expected events (Soffer & Wand, 2007). The difficulty is that these events are not within the control of the process, and their outcome may be subject to uncertainty. Hence, whenever the process is affected by an external event, it must be verified that every possible outcome of that event is addressed by the law. When incompleteness of the definition is detected, it can be resolved by modifying the law so as to address the situations that were missing in its definition.

(2) Inconsistency between the law and the goal definition: It is possible that as the process progresses, it reaches a state from which it cannot proceed further to reach a goal state.

Formally: Let S be the set of possible states and G the goal set of a process. The process law is inconsistent with the goal iff $\exists s \in S$ and $\neg \exists n$ such that $L^n(s) \in G$.

Two possibilities exist here, resulting in two consistency criteria. First, the law may keep causing transitions without reaching a stable state. If the state space is finite, this would imply the process has entered an "infinite loop". Second, it is possible the process has

reached a stable state not in the goal set for which there is no external event that can change it to an unstable state. The first case can be resolved by modifying the law to exit the loop under conditions that are certain to materialize. The second case may stand for a real exceptional situation. For example, in a sales process it may be found that the customer's credit card is not valid, nor does he have any other means for payment. Then the process must terminate without achieving its goal (sell the goods). Such stable states must be added to the goal set of the process (which denotes when the process terminates) as a special exception subset.

(3) Dependency of the process on external events: The process might be in a stable state which is not in the goal set with respect to the domain law.

Formally: $\exists s \in S$ such that $s \notin G$ and $L(s)=s$.

As opposed to the case of inconsistency discussed above, where no conceivable external event can change that state, here the process can and is expected to be resumed when the state is changed to an unstable state. By definition, this can only be the outcome of an external event. In fact, the process is "waiting" for an external event to reactivate it. However, since external events are not within the control of the process, there is no guarantee the event will occur, and the process might remain "hanging". For example, a purchasing process waits for goods to arrive from a supplier. Goods arrival is expected, but is not certain to occur. A process which includes such stable state is termed *non-continuous*, and the stable state is termed a *discontinuity point* in the process. No modification of the law can gain control over external events. Nevertheless, the process model can become valid by (a) Modifying the law so that the occurrence of the external event is *monitored*, i.e., the state becomes unstable by a time-related event. (b) The law should be adjusted to map the new unstable state to a process path (e.g., reminding external actors to generate the expected event, or selecting a different path by which the goal can be reached). (c) Defining conditions under which the stable state is considered an exception state to be added to the goal set. These conditions specify when it is apparent that the external event will not occur and the process must terminate.

The GPM validity criteria are generic, so they can be applied even when the process is not specified in GPM terms. In our study they are applied with respect to EPC models.

EPC formalization and verification

This section presents formalization and verification approaches defined for EPC. EPC (Scheer, 1998) is a popular modeling language used for process design. It can refer to

various views of the process: data view, organizational view, functional view, and control flow, or to combine them together. The control flow of EPC consists of three main constructs: function, event, and logical connector. Functions model the tasks or activities within the organization and focus on transformations from an initial state to a resulting state; events describe under what circumstances a function or a process works and in which state a function or a process results; logical connectors (AND, XOR, OR) make it possible to split the process from one flow to two or more flows and to join the process from two or more flows to one flow.

EPC's syntax has originally been semi-formal, hardly constraining the construction of models, and without precise semantics. Due to the popularity and intuitiveness of EPC as a process design language, much effort has been made to provide it with formal semantics, so mathematically proven verification procedures of EPC models can be developed. Nevertheless, while some restrictions were added to EPC syntax during the years, some degrees of modeling freedom still remain. For example, the modeler can decide whether to explicitly represent external events that occur during the process (as events without an incoming arc) or to hide them.

The commonly accepted formalization of EPC defines it as a five-tuple (E, F, C, T, A) where E, F, C are finite sets of events, functions and logical connectors respectively, T is a function which maps each connector onto connector type (AND, XOR, OR) and A is a set of arcs linking functions, events and connectors (Aalst, 1999; Kindler, 2004; Aalst, Desel & Kindler, 2002). The syntax of EPC includes the following restrictions, as summarized in (Aalst, 1999):

- An arc cannot connect two functions or two events.
- There is at least one start event and at least one final event.
- For each event, the number of input and output arcs is no more than one; for each function, the number of input and output arcs is no more than one; for each connector, the number of input and output arcs is at least one.
- For each join connector, the number of input arcs is at least two and the number of output arcs is one; for each split connector, the number of input arcs is one and the number of output arcs is at least two.

As mentioned above, the formalization of EPC serves as a basis for model verification methods, where the main property addressed by EPC verification is *soundness*. Soundness was originally defined for Workflow-nets (WF-nets), which are a specific form of strongly connected Petri-nets, having one initial place and one final place (Aalst, 1998). Considering

WF-nets, soundness satisfies three conditions that ensure the proper termination of the represented process, which should reach its final place and stop being active. The property of soundness is, in essence, applicable to various modelling languages, as demonstrated by Hee et. al. (2008). However, it should rely on an accurate semantics assigned to these languages, usually depicting model behaviour in terms of token transitions and distribution.

The application of soundness to EPC had to solve several semantic difficulties. For example, soundness is based on a defined initial state and a defined final state in a model, whereas EPC may have multiple initial events (events that have no input arcs) and final events (events that have no output arcs). To resolve these difficulties, a number of approaches were proposed (e.g., (Kindler, 2004; Kindler, 2006; Verbeek & Aalst, 2006; Verbeek, Aalst & Hofstede, 2007)), relying on different semantic interpretations assigned to EPC. Most of the approaches require the EPC not to explicitly represent multiple initial and final events (namely, to “hide” them), before or during the verification procedure. Recently, a soundness definition was proposed, where multiple initial and final events are taken into account (Mendling & Aalst, 2007). Informally summarized, this definition of soundness requires the following three conditions: (1) The occurrence of every initial event is possible, (2) a final state is reachable from every state which is reachable from an initial state, and (3) every possible final state of the process is such that no other parts of the process are active when it is reached, and this does not hold for any EPC state that is not final. Based on this definition, a verification algorithm can be applied to EPC models which explicitly represent external events and do not “hide” them. Nevertheless, hiding these events is still commonly practiced, where the main motivation is to concentrate on the activities performed within the process and keep the representation from being overloaded. Hence, explicitly representing external events or hiding them remains a choice made by the process designer.

Note that there are other properties which are defined for EPC and related to soundness, such as relaxed soundness (Dehnert & Rittgen, 2001) and well-structuredness (Aalst, 1999). However, relaxed soundness is weaker than soundness, and well structuredness is not a sufficient condition for soundness, hence we do not address them in detail. Also note, that there are different methods for verification of soundness and of other properties (e.g., based on reachability graph (Aalst, 1999) or on reduction rules (Dongen et. al., 2005)). We address the verified property (namely, soundness) rather than the verification method.

A different property that can be verified regarding an EPC model is *robustness* (Dehnert & Aalst, 2004). While the soundness-related properties address the process as it should be executed by a workflow management system, robustness relates to the interaction between

the process and its environment. An EPC is robust if its final event is reachable for every possible input from its environment, where the inputs can be results of external events or of evaluation of external information (i.e., decisions). In order to check an EPC for robustness, it should be transformed into a WF-net, and possible external events (or transitions controlled by the environment) should be identified. According to Dehnert and Aalst (2004), these can be identified when examining an EPC whose multiple initial events are explicitly represented.

Robustness may be comparable to the controllability property (Lohmann et. al, 2008), which is defined for Open WF-nets (not for EPC). Open WF-nets are WF-nets that include an explicit specification of the interaction with the environment. Controllability verifies the absence of deadlocks in a model (including a lack of response from the environment). However, controllability does not address livelocks, hence it does not ensure the reachability of the final state. In that sense, it is weaker than robustness.

Comparing verification-related properties and validity

The validity criteria are generic and not designed for a specific modeling language. Furthermore, they are not a technique that can be structurally or automatically applied. In this section, we start by relating the notion of goal to EPC models; then we examine the existing EPC verification techniques and show they cannot ensure a valid model.

Relating process goal to an EPC model

As a first step towards identifying and evaluating the specification of the process goal in EPC, we provide an interpretation of a state in EPC, and distinguish stable from unstable states. We use the following notation taken from Mendling and Aalst (2007). The sets of incoming arcs and outgoing arcs of an event e are marked e_{in} and e_{out} , respectively. The set of events in an EPC includes three subsets: E_s the start events, E_{int} the intermediate events, and E_e the end events.

As mentioned, events in EPC describe pre- and post-conditions of functions. Hence, despite their name, they are not really “events” in the traditional meaning of the word (a momentary occurrence), but rather represent a state which might last for an unlimited period of time. In GPM terms, an event is equivalent to a set of states of a sub-domain, while arcs and functions reflect the law¹. This interpretation is consistent with the GPM-based semantics of

¹ Functions, which are actions that lead to state transformation, are abstracted from in GPM

WF-nets (Soffer, Kaner & Wand, 2008). The state of the entire domain is defined differently by different semantic interpretations of EPC. While all these interpretations are token-based, some address the state of an EPC as the token distribution among the nodes of the EPC (in correspondence to WF-nets, e.g., Dehnert & Rittgen, 2001), while others relate to the token distribution among the arcs (Kindler, 2004; Mendling and Aalst, 2007). Without going into the details of these different semantics, we extend our interpretation of an event, so the state of the entire domain at a moment in time is determined by all the events that are active (“holding tokens”) at that moment.

Lemma 1 (state of sub-domain in an EPC): Let e be an event, and D_e be the sub-domain over which e is defined, then

- (a) If $e \in E_e$ then D_e is stable in e ,
- (b) If $e \in E_s$ or $e \in E_{int}$ then D_e may be stable or unstable in e .

Proof: (a) For $e \in E_e$, $|e_{out}|=0$, so there is no law that maps e to a different state. Hence every state $s \in e$ is a stable state of D_e . (b) Proof by example: Figure 1(a) demonstrates that both are possible.

In the example of Figure 1(a), E_s has three events. *Order received* denotes an unstable state for the sub-domain of the order, while *Item available* and *Item unavailable* include stable states with respect to the inventory sub-domain, and can transform only when an order is received (in the other sub-domain). E_{int} has two events, *Order confirmed* and *Payment received*. *Payment received* is an unstable state, which should transform to a state where the order has been delivered. In contrast, *Order confirmed* is a stable state, which can only transform when the customer pays for the order. Since this is an external event, the intermediate state of *Order confirmed* is a stable state where the process is “waiting” for an external event to proceed. As mentioned, EPC allows the modeler to decide whether to “hide” external events or to specify them, and Figure 1(a) demonstrates such “hiding”. Figure 1(b) shows the same process without hiding the external event of the payment, and it is easier to see that the process cannot progress until the occurrence of the external event.

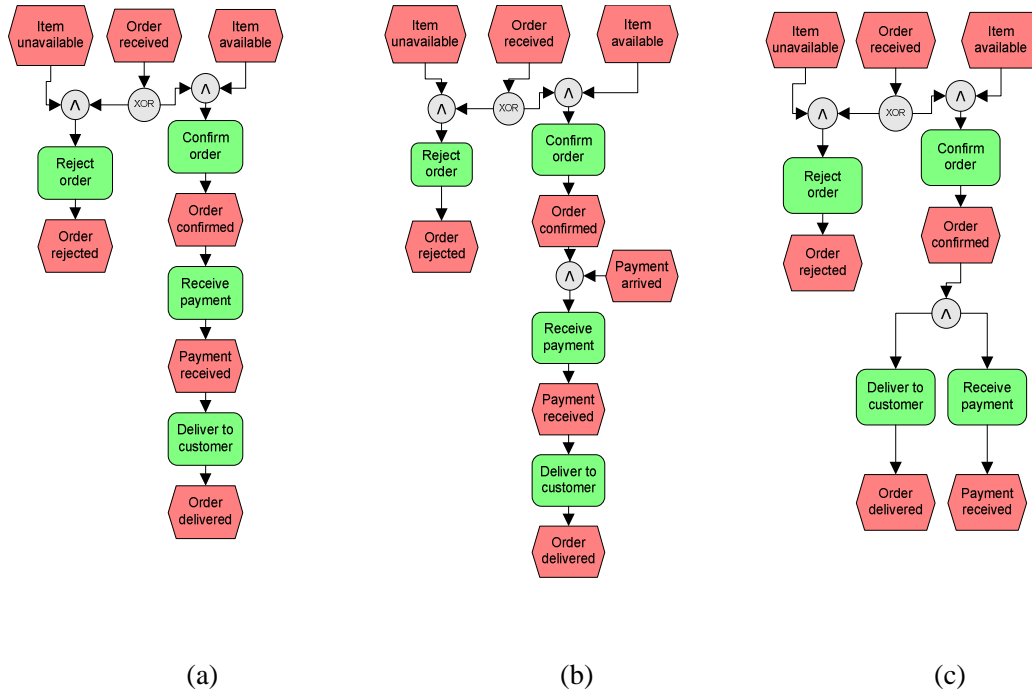


Figure 1: Order delivery process examples

According to GPM, the process goal G is a set of states which are stable for the entire process domain.

For an EPC to be valid, its possible final states must be in the defined goal of the process. Assume an EPC has only one final event e , then it must satisfy $e \subseteq G$. In case an EPC has more than one final event, these can be the endings of alternative paths, or of paths that should be performed in parallel. In the example of Figure 1 (a) and (b), assume the goal is defined with respect to two state variables, the order status and the payment status, so $G = \{s \mid ((\text{Order}=\text{delivered} \wedge (\text{payment}=\text{completed})) \vee (\text{order}=\text{rejected}))\}$. The two final events, *Order rejected* and *Order delivered*, which can be denoted as $e_1 = \{s \mid \text{Order}=\text{rejected}\}$ and $e_2 = \{s \mid \text{Order}=\text{delivered}; \text{Payment}=\text{completed}\}$, satisfy $e_1 \subseteq G$ and $e_2 \subseteq G$. These events are on alternative paths, and only one of them should be reached by the process. As a second example, consider the process shown in Figure 1(c), and assume its goal is the same as the previous one. In that case the EPC has three final events, $e_1 = \{s \mid \text{Order}=\text{rejected}\}$, $e_2 = \{s \mid \text{Order}=\text{delivered}\}$, and $e_3 = \{s \mid \text{Payment}=\text{completed}\}$. The possible final states for the EPC to achieve its goal are e_1 or $e_2 \cap e_3$.

Summarizing this discussion, the combination of final events that form the goal of an EPC can be identified as follows².

Definition 1: Let G be the goal set of a process P , let \mathcal{F} be the power set of E_e of an EPC that represents P , and let m be an element in \mathcal{F} . m includes a *goal-fulfilling* final state of the EPC if $\bigcap(e_i \in m) \subseteq G$, and there is no $m' \in \mathcal{F}$, such that m' is a goal-fulfilling final state and $m' \subseteq m$.

Validity assessment

For a process model to be valid, we should make sure that every possible enactment of that process will reach a goal state (which meets the above condition). However, as mentioned above, the validity criteria provide a list of possible causes for invalidity rather than a structured technique for identifying them in a model. In what follows, we examine the verification techniques of soundness and robustness, to evaluate whether they can be used for assessing the validity of an EPC model.

Soundness relates to the internal structure of the process, while robustness relates to its interaction with the environment. Hence, the combination of soundness and robustness may suffice for validity assessment. However, examining these properties, we identified the following three shortcomings.

First, both soundness and robustness verification relate to the reachability of final events, without explicitly addressing the process goal. Figure 2 presents two example processes, which are both sound and robust. However, considering Figure 2(a), and assuming that the process goal should be $G = \{s \mid \text{Goods}=\text{received} \wedge \text{Payment}=\text{completed}\}$, the final event, $e = \{s \mid \text{Payment}=\text{completed}\}$, does not satisfy $e \subseteq G$. Clearly, the process model is not correct from a business view. Since it is possible to reach a state where payment is made although goods were not received, the process has a validity problem of inconsistency between the law and the goal. Figure 2(b) includes a simplified part of a process taken from the SAP reference model. We assume that its goal should be $G = \{s \mid (\text{Billing document}=\text{completed} \wedge \text{Goods issue}=\text{posted}) \text{ XOR } \text{Delivery}=\text{refused}\}$. However, its possible final states are $S_1 = \{s \mid \text{Billing document}=\text{completed} \wedge \text{Goods issue}=\text{posted}\}$, $S_2 = \{s \mid \text{Billing document}=\text{completed} \wedge \text{terms of credit}=\text{changed}\}$, $S_3 = \{s \mid \text{Billing document}=\text{completed} \wedge \text{terms of credit}=\text{change refusal}\}$.

² Note that we assume that every final event is well defined in terms of the state variables that define the goal.

Clearly, only S_1 is in the goal. Furthermore, S_3 is a state where billing was completed even though the customer is not credit worthy. Note that the SAP reference model, which was constructed as a “best practice” repository, has been subject to structural verification, which has revealed cases of unsound models (Mendling, 2007). Nevertheless, a structural verification which does not relate to the business logic of the process cannot detect this kind of modeling error, and indeed this validity problem was not revealed by it.

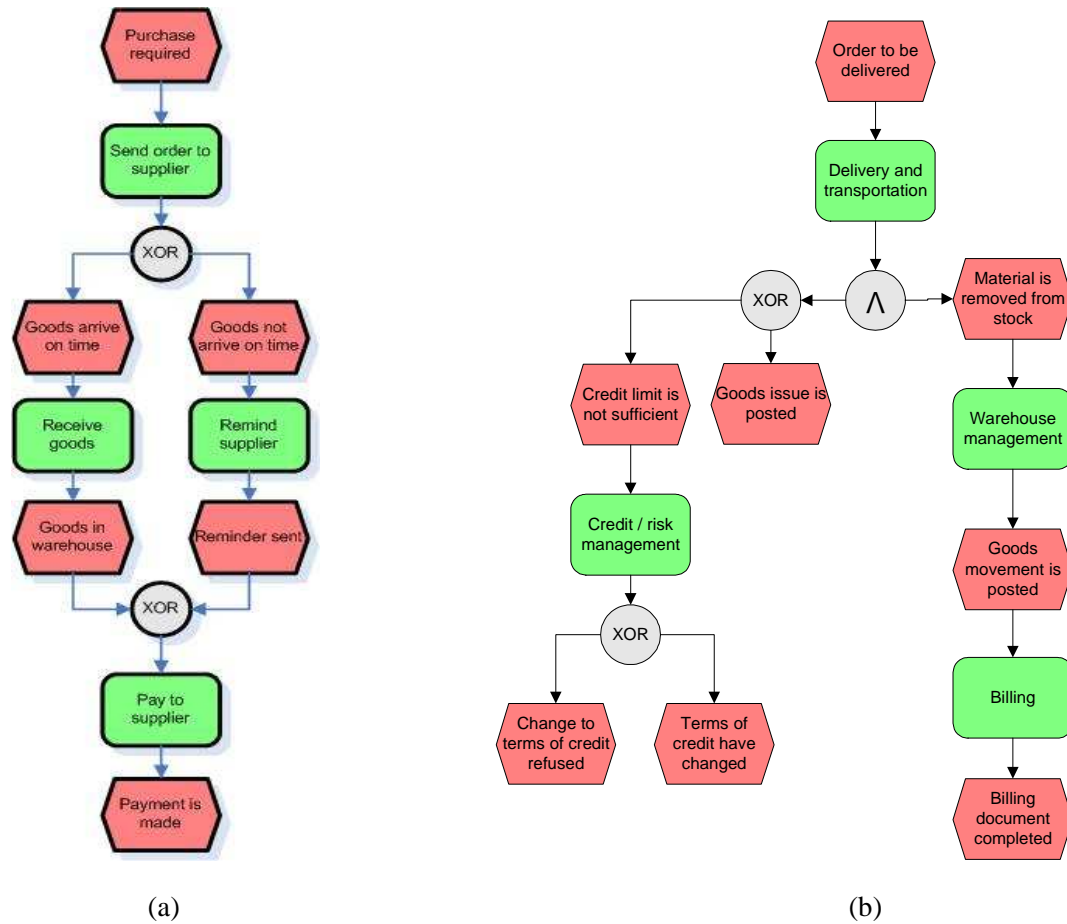


Figure 2: Process examples: sound, robust, and invalid

Second, assume the first shortcoming can be solved by (manually) verifying that the EPC includes a combination of final events that constitute a goal-fulfilling final state. Still, soundness as defined by Mendling and Aalst (2007) requires only the reachability of a final state where no other parts of the process are active. This may be any final state, not necessarily a final state which is in the goal. Note that other soundness definitions (e.g.,

Aalst, 1999) relate to an EPC that has only one final event, which can, in principle, be verified (manually) to be within the goal³.

Third, robustness is a property of a process where, facing all possible inputs from the process environment, the final event can be reached⁴. Robustness verification identifies environment-controlled transitions and analyzes the process reachability graph with respect to the possible external events generated by the actions of the environment. It does not take into account a possibility that the environment may not respond (as is highlighted when analyzing discontinuity points according to GPM's validity criteria). Such possibility should be identified prior to robustness verification and addressed (by monitoring), so that robustness verification may relate to the resulting process definition. The robustness verification procedure operates under the assumption that the human modeler has identified all the possible environment behaviors and incorporated them into the process model. The validity criteria are aimed at supporting the modeler in performing these tasks.

Table 1 summarizes the relation of soundness and robustness verification to the different validity problems of GPM.

Table 1: Summary of validity problems and verification techniques

Validity problem	Soundness verification	Robustness verification
Inconsistency between the law and the goal	Identifies structural problems (e.g., deadlocks), assuming every final event is in the goal	-
Incompleteness of the law (with respect to internal events)	Identifies structural problems (e.g., deadlocks), assuming every final event is in the goal	-
Incompleteness of the law (with respect to external events)	-	Identifies incompleteness if the process includes one final event and it is in the goal
Dependency on external events	-	-

³ These soundness definitions correspond to the WF-nets soundness property. Soffer et al (2008) proved that modeling rules derived from GPM's validity criteria yield sound WF-nets, and that in general, sound WF-nets are not necessarily valid.

⁴ Note that robustness relates to *the* final event, since it is based on a transformation to a WF-net, which has a single final place.

In summary, soundness and robustness verification address structural properties of the process, while the validity criteria relate to its business logic. While structural correctness can be established automatically and definitely by verification algorithms, validity in terms of the business logic and business goal requires human reasoning. The combination of human-based validity analysis with automated verification should be able to address both business logic and structural “correctness” of a process model. In proposing this combination, we assume that the validity criteria can affect the quality of models produced by humans even without a structured application method. However, this is an assertion that needs to be tested. To establish the need for using the validity criteria and their effectiveness in process design, we performed the empirical study reported in the following section.

Empirical study

Aim and model

As suggested in the previous section, we propose to use the validity criteria for supporting the human task of process design. The design of a process model is iterative in nature, where design alternatives are created, evaluated, and modified, until a final alternative is selected. We expect the validity criteria to guide the analysts when evaluating design alternatives and thus to contribute to the design of valid processes. Currently, no such support is available, and designing valid processes relies on the knowledge and expertise of the modeler. Furthermore, as discussed above, the validity criteria do not constitute a structured methodology. Hence, it may be questionable whether the mere awareness of these criteria can have any effect on the quality of models.

The aim of the empirical study reported here was to assess the applicability of the validity criteria and their contribution to model quality. Of the two tasks iteratively practiced at process design, namely, creating a model and evaluating it, validity criteria are expected to affect the latter. Hence, we addressed the evaluation task, where validity problems that exist in a model should be identified.

In addition, since EPC allows different forms of representation of a given process (as discussed in the previous section), the study aimed to find whether these representation possibilities influence differently the human ability to detect validity problems and the effectiveness of the validity criteria. In particular, we addressed the reduced or implicit representation where external events are “hidden”, and compared it to an explicit representation of external events. Our expectation was that 1) the subjects’ awareness of the

generic problems (validity criteria) would better support the identification of these problems in the specific model; 2) an explicit representation of external events would better support the identification of validity problems that are associated with these events.

Considering this task, we asked the following three main research questions:

1. Does the guidance of GPM's validity criteria improve the human ability to identify process invalidity?
2. Does the explicit representation of external events improve the human ability to identify process invalidity?
3. Is there an interaction between the model evaluation approach (with and without validity criteria) and the model representation (explicit and implicit)?

		External event representation	
		Implicit	Explicit
Validity criteria	Provided	Group 1	Group 2
	Not provided	Group 3	Group 4

Figure 3: Four groups – two factors with two levels in the factorial design

Regarding these questions we used a two-factor factorial experimental design (Breyfogle, 2003; Shah & Madden, 2004). The first factor relates to the validity criteria. This factor has two levels: 1) validity criteria were provided; 2) validity criteria were not provided. The second factor relates to the model representation. This factor has two levels: 1) implicit representation of the external events; 2) explicit representation of the external events. Hence four groups of subjects participated in the experiment as illustrated in Figure 3.

We formulated the following sets of hypotheses respectively to the three research questions:

H¹0: There is no effect of GPM's validity criteria on the identification of process validity problems (vs. a positive effect of GPM's validity criteria).

H²0: There is no effect of explicit representation of external events on the identification of process validity problems (vs. a positive effect of explicit representation⁵).

H³0: There is no interaction between the model evaluation approach and the representation of external events (vs. there is an interaction between the two factors).

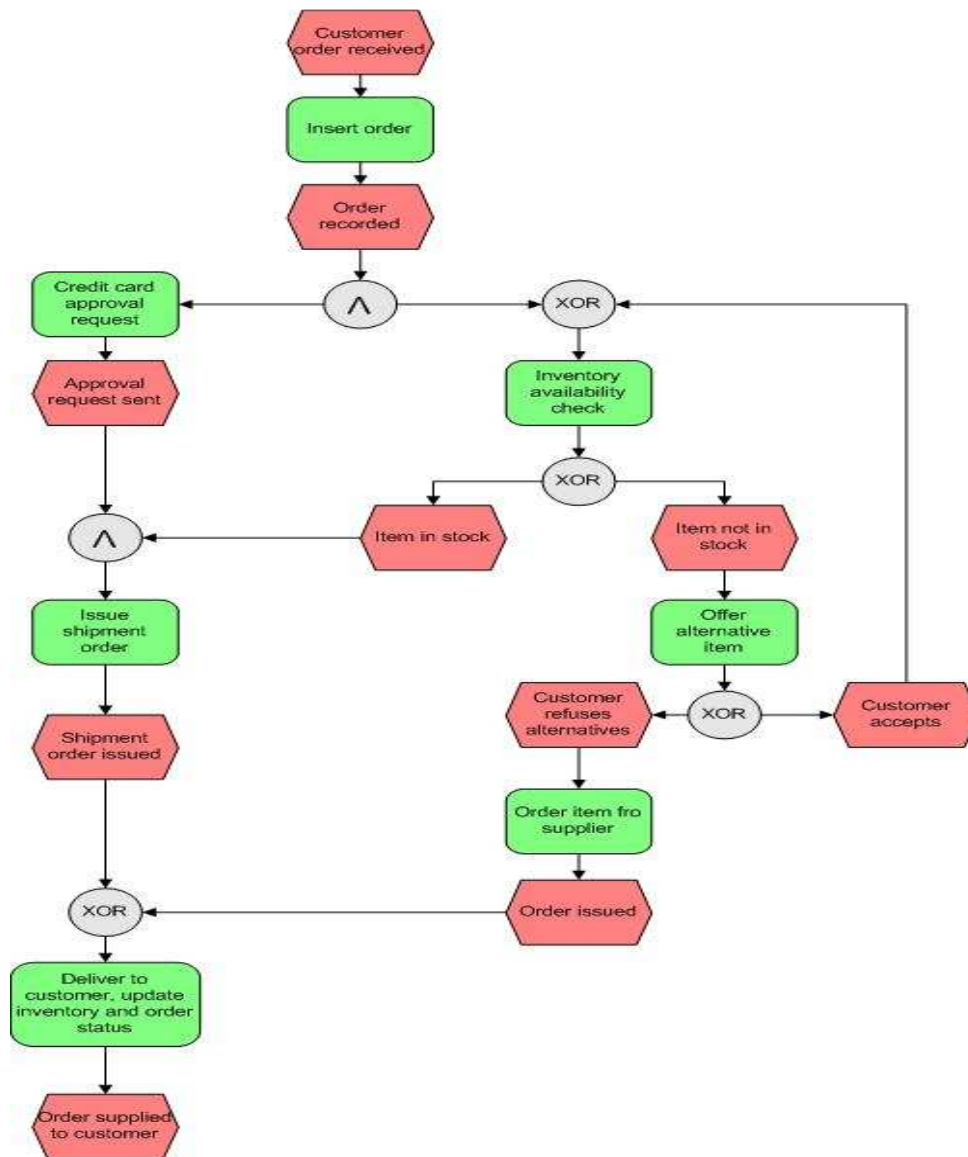
⁵ Note that a positive effect is likely with respect to validity problems that are related to external events. We assume that as a result, the overall performance will be affected.

Experimental settings

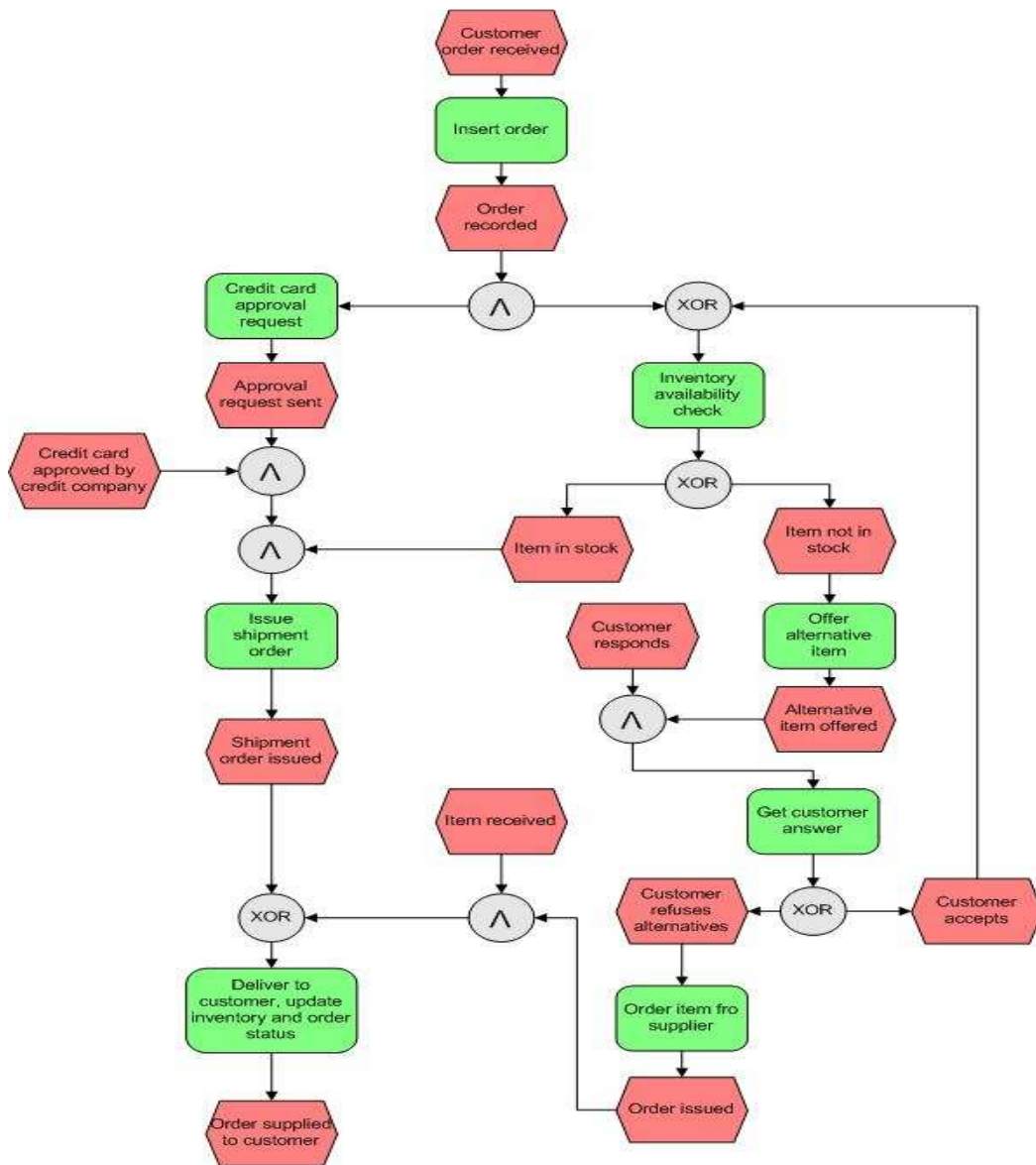
The empirical study was conducted as a laboratory experiment whose subjects were students (novices). This choice of laboratory experiment is similar to the one made in other empirical studies addressing business process modeling (e.g., Sarshar & Loos, 2005; Recker, Rosemann and Aalst, 2005; Vanderfeesten et al., 2008). The participants in the experiment were 80 MIS students in a Systems Analysis and Design course. Each of the participants was arbitrarily assigned to one of four groups as presented in Figure 2: 19 subjects in group 1; 20 subjects in group 2; 19 subjects in group 3; 22 subjects in group 4.

To guarantee that the arbitrary group assignment resulted in groups whose potential performance did not influence the experimental results, we took the following measure. At the end of the course we performed an ANOVA one way analysis of variance to the final grades achieved in the course. We compared means of grades of the students in these four groups. With respect to the hypothesis of no difference between the groups, we received a p-value =0.905 (Levene statistic equals 0.53), so we assume that all groups have the same mean of course final grades.

The students had all learnt and practiced the use of EPC as part of the course before the study was conducted. All subjects were given an EPC model of an order handling process, whose goal was stated as “ordered goods supplied to a customer whose credit card is approved”. The students were instructed to identify all problems which may prevent the process from achieving its goal. The participants in group 1 and group 3 got the implicit representation of the process model as illustrated in Figure 4 (a). The participants in group 2 and group 4 got the explicit representation of the model as illustrated in Figure 4 (b). The validity criteria were given to the subjects in group 1 and group 2 as a list of possible problem categories (see Figure 5), and the students in these groups were instructed to classify each identified problem accordingly.



(a) Implicit model



(b) Explicit model

Figure 4: EPCs in the experimental task

No information was given to the subjects regarding the number of problems they were expected to identify, and no time limitation was placed for performing the task. To increase the motivation of the students, a 5-point bonus in the course grade was promised to the four students whose task performance would be the best. The researchers were present in class at the time of the experiment and answered any question that was raised regarding the understanding of the modeled process.

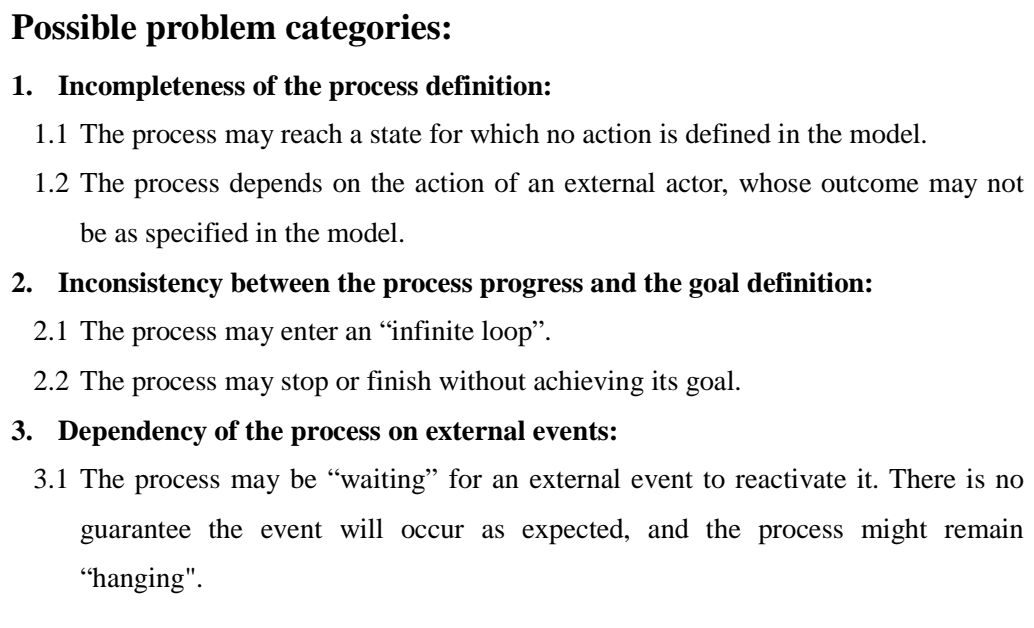


Figure 5: List of invalidity categories derived from GPM, given to subjects in groups 1 and 2⁶

Data analysis

The process model handed out to the students (Figure 4) included the following five invalidity sources, where seven problems can be identified. These, according to Soffer and Wand (2004), represent all the possible problems categories (Figure 5).

1. When a credit card approval request is issued, the process model does not address a possibility that
 - a. the customer’s credit card is not approved (incompleteness – problem 1.2),
or
 - b. that the credit company may not respond (discontinuity – problem 3.1).
2. An infinite loop is possible when alternative items, whose availability is not checked, are proposed to the customer (inconsistency – problem 2.1).
3. The process must wait for the customer’s response to the alternative item offer. The model does not address a possibility that
 - a. the customer may decide to cancel the order when the item ordered is out of stock (incompleteness – problem 1.1), or

⁶ Note that this list is categorized differently than **Table 1**, where problems are associated with verification techniques. Problems 2.1 and 2.2 correspond to the first row in **Table 1**; problems 1.1 and 1.2 correspond to the second and third rows in the table; problem 3.1 corresponds to the fourth row.

- b. the customer may not respond to the offer (discontinuity – problem 3.1).
- 4. The process must wait for the supplier to provide the item, whereas the supplier may not do so (discontinuity point – problem 3.1).
- 5. There is a possibility for the customer order to be fulfilled even when the customer's credit card is not approved (inconsistency – problem 2.2).

Note that soundness verification would identify the process as being sound⁷, and that robustness verification with respect to the external events specified in Figure 4(b) would find the process robust.

Based on these invalidity sources, the subjects' solutions were graded according to a scheme where identification of each problem scored one point out of seven possible.

To analyze this data according to the formulated hypotheses we used non-parametric tests, as the scores are not normally distributed. First, using the Kruskal-Wallis test we analyzed if there is a difference between the performance means of the four groups. The result of $p\text{-value} < 0.0001$ indicates that there is a difference between the performance levels of the groups. In order to analyze the effect of GPM's validity criteria guidance on the identification of process validity problems (H^1_0) we compared the aggregate performance of groups 1 and 2 versus the aggregate performance of groups 3 and 4 using Mann Whitney test. In order to examine if there is a difference in the ability of students to identify problems belonging to different invalidity categories, we distinguished between problems 1 (1a and 1b), 3 (3a and 3b), and 4, which relate to the external environment, and problems 2 and 5 that relate to internal process inconsistency (Figure 5).

In order to analyze the effect of the explicit representation of external events on the identification of process validity problems (H^2_0), we compared the aggregate performance (for different invalidity problems) of groups 1 and 3 versus the aggregate performance of groups 2 and 4 using Mann Whitney test.

In order to test the interaction between the model evaluation approach and the model representation (H^3_0), we analyzed the interaction graphs and used the Adjusted Transform test for analyzing interactions in nonparametric statistics (Sawilowsky, 1990).

⁷ According to the definition of Mendling & Aalst (2007). Note that the corresponding WF-net is not sound, and this verification can identify problem 5, but not the other problems.

Findings

This section presents the findings of the data analysis with respect to the research questions and hypotheses. Table 2 provides the means and standard deviations of the grades achieved in the four groups.

Table 2: Performance means and deviations

Validity criteria provided	Group	Implicit representation			Explicit representation		
		Number of subjects	19			20	
Performance mean		2.16*	0.95**	3.11***	2.6	1.1	3.7
Standard deviation		0.834	0.621	1.1	1.07	0.3	1.14
Validity criteria not provided	Group	3			4		
Number of subjects		19			22		
Performance mean		0.895	0.895	1.789	1.27	0.455	1.727
Standard deviation		0.809	0.567	0.787	1.08	0.596	1.12

*- problems 1a, 1b, 3a, 3b, 4; ** - problems 2, 5; *** - all problems

Research question 1 and the related hypothesis referred the effect of GPM's validity criteria on the identification of process validity problems. We tested whether the aggregate performance mean of the 39 subjects from groups 1 and 2, whose analysis was based on the validity criteria, is better than the aggregate performance mean of the 41 subjects from groups 3 and 4, whose analysis was without the use of validity criteria.

The comparison yielded a highly significant effect with respect to all problem categories (p-value (all)<0.001, p-value (1a, 1b, 3a, 3b, 4)<0.001, p-value (2, 5)=0.002). We can conclude that in general, H^1_0 can be rejected, thus our hypothesis that the validity criteria support better identification of process validity problems is corroborated.

Research question 2 and the related hypothesis addressed the effect of explicit representation of the external events on identification of process invalidity problems. In particular, we tested for a positive effect on the identification of problems related to external effects and hence on the overall performance.

We tested whether the aggregate performance mean of the 38 subjects from groups 1 and 3, whose models included an explicit representation of external events, is better than the aggregate performance mean of the 42 subjects from groups 2 and 4, whose models included

an explicit representation of external events. This test showed that we can reject H^2_0 with respect to the external environment-related problems (p-value [1a, 1b, 3a, 3b, 4] = 0.0975) at a confidence level of 10 %; however we cannot reject H^2_0 with respect to all problems (p-value [all problems] = 0.285). In other words, our hypothesis of a positive effect of explicit representation has been corroborated only with respect to external event-related problems.

To gain a better understanding, we tested the effect of model representation on the performance means of the groups, separating the different modes of evaluation – with and without the validity criteria (for the different problem categories).

Comparing groups 1 and 2 we found a positive effect of explicit representation on identification of problems 1a, 1b, 3a, 3b, 4 (p-value = 0.055) and, as a result, for all problems (p-value = 0.049). When comparing groups 3 and 4, no effect of explicit representation was found; however, the performance means (for problems 1a, 1b, 3a, 3b, 4) of students in group 4 (explicit representation) was higher than in group 3. We can conclude that when the validity criteria are used, an explicit representation supports better identification of validity problems.

Research question 3 and the related hypothesis addressed the interaction between the model evaluation approach (with or without validity criteria) and the representation of external events (implicit vs. explicit). The interaction graphs are shown in Figure 6 (for all problems), Figure 7 (problems 1a, 1b, 3a, 3b, 4), and Figure 8 (problems 2, 5).

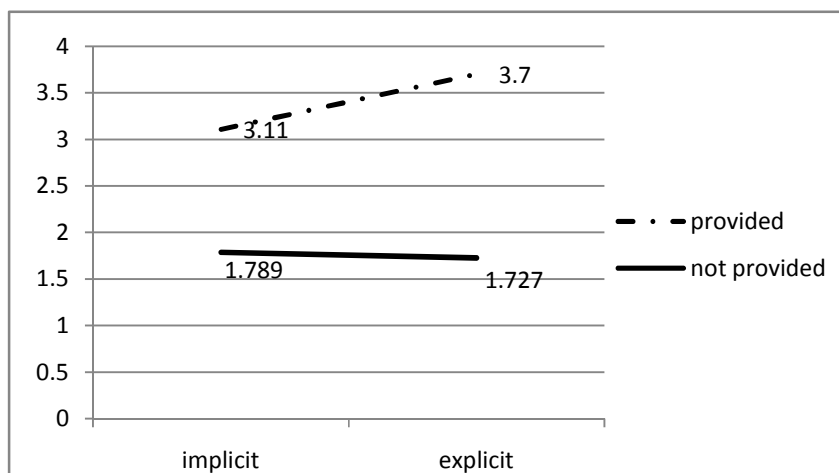


Figure 6: Interaction graph: all problems

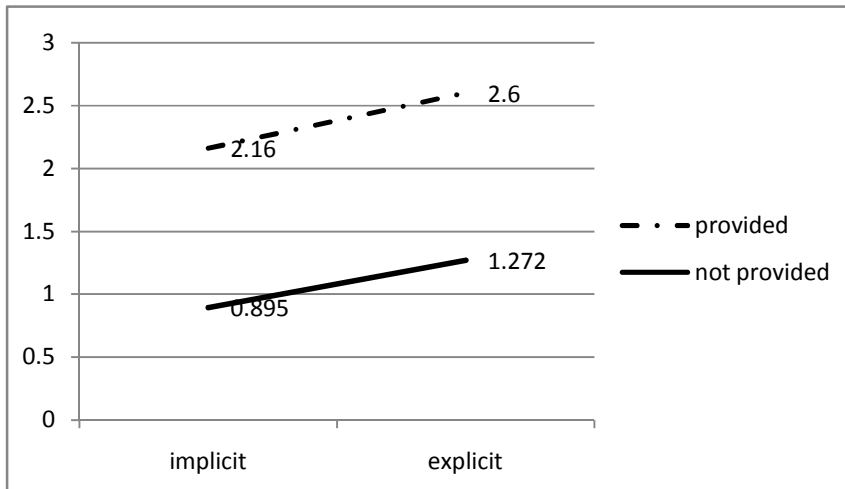


Figure 7: Interaction graph: problems 1a, 1b, 3a, 3b, 4

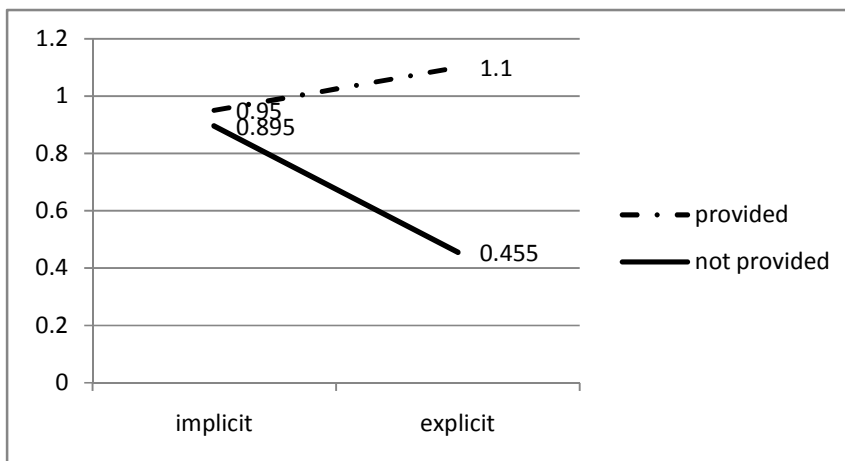


Figure 8: Interaction graph: problems 2, 5

The lines in figure 6 (with respect to all problems) are not parallel, but without a statistical significance. Hence, H^3_0 cannot be rejected with respect to all problems. A separate analysis with respect to the different problem types yielded the following results. The parallel lines in Figure7 show that there is no interaction between the factors, hence H^3_0 cannot be rejected with respect to the external event-related problems. This result can be explained based on the findings mentioned above: for both evaluation modes an explicit representation supports a better identification of external event-related problems – significantly when validity criteria were used, and not significantly but with a higher performance mean when they have not been used.

The non-parallel lines in Figure 8 show the existence of an interaction (p-value =0.006) hence H^3_0 can be rejected with respect to the inconsistency problems. This result can be explained as follows: while no significant difference was found between the performance level of groups 1 and 2; a significant positive difference of implicit representation (p-value=0.009) has been found between the performance means of group 3 and group 4. This may have two possible explanations: (1) an explicit representation of external events yields a more complicated model, which may even become too overloaded; (2) the subjects were not aware of the problems related to external events and hence concentrated on the process logic inconsistency manifested in problems 2 and 5.

Discussion

The empirical findings

The findings of the empirical study indicate the applicability of the validity criteria and their importance in process design. It is shown that relying on human reasoning and common sense is not sufficient for assessing the validity of a process model and identifying invalidity causes. Clearly, when possible error types are given, the analyst knows what to “look for” and is hence capable of identifying problems in a systematic manner. This finding may seem not surprising as it has been found earlier (Grabski, Reneau & West, 1987) that prompting (giving a list of problems) supports better performance than problem recall (determination from one’s memory). However, the GPM validity criteria are at a rather abstract level, not directly related to the modeled domain or operationalized to EPC terms. Hence, their applicability, namely, the possibility of humans to effectively operationalize and apply them to a specific model, could not be taken for granted. In particular, the empirical findings indicate that even novices are able to benefit from the abstract validity criteria and apply them to a concrete situation.

Regarding the effect of explicit representation of external events, an explicit representation highlights external events, whose role in process invalidity is emphasized by GPM’s validity criteria. Hence, when an analyst uses the validity criteria, the explicit representation assists in the identification of validity problems related to unexpected actions of external actors. Yet, when the analyst is not aware of the validity criteria, trouble shooting is done in an unsystematic manner, and the overloaded model resulting from an explicit representation negatively influences identification of problems related to internal events.

Some limitations of the experimental study should be mentioned. First, as already discussed, the population from which the subjects were taken is of students, whose experience is

limited, as opposed to professional analysts. However, the study deals with providing guidance and support to the design process, which seems to be particularly crucial when the analyst's experience is limited. It is possible that experienced analysts have developed reasoning skills which enable them to better identify validity problems in a process model. We believe it would be beneficial to conduct a similar study whose subjects are experienced analysts. Nevertheless, verification of models taken from the SAP reference model, which was constructed by experienced professionals, revealed cases of unsound and non-robust models (Verbeek, Aalst & Hofstede, 2006, Mendling, 2007). The example given in Figure 2(b), taken from that reference model, shows that additional validity problems, undetected by verification algorithms, still exist there. Hence, while the findings of this study are at least applicable for analysts that have a relatively little experience, it is not unlikely that their applicability extends to experienced analysts too.

Second, the study included a model of a specific process in a single modeling language, EPC. The choice of EPC was made for the two main reasons explained before, namely, its popularity and its various verification capabilities. GPM's validity criteria, on the other hand, are notation independent and can be applied to processes modeled in other modeling languages as well. Hence it can be assumed that the findings that address the applicability of GPM's validity criteria are not specific to EPC models, while the findings related to explicit or implicit representations can be applicable only to languages where such different representations are possible. The process model itself was relatively simple and cannot indicate the scalability of the approach, which is yet to be tested. However, the following observations regarding scalability can be made. First, an explicit consideration of the process goal should not be affected by the size of the model. Second, the evaluation of possible results of external events is locally performed for each such event according to the validity criteria, independently of the size of the model.

Last, as opposed to the real-world situation, where the process designer is the one who designs and evaluates the validity of the process design, here the students were given a process model and were required only to evaluate its validity. It may also be argued that the students were not experienced in model evaluation (as their training was in process design). However, as already explained, we view the evaluation of models as being an integral part of the design process, and we focused on this task in the experiment in order to better isolate the effect of the validity criteria.

Implications for practice:

The implications of our findings for the practice of process design relate both to our theoretical analysis and to the empirical findings. The theoretical analysis shows that validity assessment and structural verification of control-flow properties are complementary, both in the issues they address (goals and business logic vs. executability) and in the ways by which they are achieved (human-based vs. automated algorithms). The importance of this finding lies in the fact that the validity criteria are not commonly used, and human intuition is usually relied upon for assessing the reachability of the process goal. Furthermore, while the use of verification techniques may give the impression that the process is "correct", we have shown that they do not consider the process goal and thus cannot replace validation. The empirical findings show that the application of validity criteria by humans is effective in detecting validity problems and that the need for such criteria to guide a systematic analysis of process validity is real. Moreover, most of the problems detected in our study are such that would not be identified by verification procedures. It is also shown that different possible representations may affect the human performance when applying the validity criteria.

The conclusion that follows the findings reported here is that process design should include a step of human-based validity assessment in which the validity criteria are applied and relate to the goal of the process. This step should be followed and complemented by an automated structural verification, which relates to the executability of the process after its business logic has been assessed. When the language used for process design is EPC, where explicit or implicit representation of external events is determined by the process designer, an explicit representation will better support the application of the validity criteria.

Implications for research on process model quality:

Our investigation of the complementary nature of validity assessment and structural verification highlights the need for a comprehensive framework of process model quality. An attempt in that direction has been made by Recker (2007), who proposed a theoretical framework for understanding process model quality. To operationalize such framework, existing approaches and techniques should be mapped to it. Then it might be possible to identify sets of complementing techniques that together relate to all aspects of model quality. The framework of Recker (2007) relates to a process model as a conceptual model, and follows notions developed for conceptual model quality. Most notably, it adapts the quality framework of Lindland, Sindre & Solvberg (1994) to process models. Indeed, at the design phase in its life-cycle, a process model is a specific type of conceptual model, conceptualizing behavior of some part of the real world. Hence, we also use this framework

as a basis for our discussion here. The framework distinguishes three dimensions of model quality: semantic quality, namely the correspondence between the model and the domain, syntactic quality, namely the correspondence between the model and the modeling language, and pragmatic quality, namely the correspondence between the model and its users.

Considering the properties addressed in this paper, namely, validity, soundness, and robustness, they are roughly classified by Recker (2007) as syntactic and semantic quality properties. Soundness is usually considered by the business process research community as standing for semantic quality (e.g., Aalst, 1999, Kindler, 2004, Aalst et. al., 2009). However, the use of the term “semantic” in this context relates to the mathematical token-based semantics of the language rather than to the correspondence to the domain. Recker (2007) interprets executability as syntactic quality. However, we consider this interpretation as going beyond the meaning of “syntax”. With respect to the essential syntactical rules of process modeling languages such as EPC, a model that follows the syntax rules is not necessarily executable.

The analysis in this paper leads to a different mapping of the properties under consideration to the quality dimensions. As indicated by Recker (2007), the application of the quality framework, and particularly the pragmatic quality dimension, should consider the intended use of the model under consideration. Traditionally, pragmatic quality has been considered as the interpretability of a model (Siau & Tan, 2005). However, this consideration is with respect to the intended use of conceptual models, which serve for purposes of communication and understanding (Mylopoulos, 1992). In contrast, the intended use of process models is not only to support communication, but to eventually be executed. Hence, the concept of pragmatic quality can be extended in this respect and include the executability of the modeled process. Considering this interpretation, we claim that the properties of soundness and robustness represent aspects of pragmatic quality. In addition, the verification of these properties ensures syntactic correctness of the model. Validity, on the other hand, relates both to semantic and to pragmatic quality. The semantic aspect of validity is in the assessment of whether the process model corresponds to the goal seeking nature of the real world process. A valid process model represents a real world process that has a defined goal which it is designed to achieve. The pragmatic aspect is in the assessment of the ability of the designed process to achieve its goal.

Based on this mapping, the complementary nature of validity assessment and structural verification is better understood. Yet, this does not mean that all the required quality aspects are covered by this combination. Other aspects of semantic quality, such as model

expressiveness, and of pragmatic quality, such as model understandability, are not addressed here. Based on this discussion, we believe that a correct and consistent mapping of process model quality properties to the above discussed quality dimensions is a step towards the operationalization of the quality framework.

Conclusions

Validity of a process model should be achieved at the design phase, when errors can be easily corrected. Much effort has been made suggesting procedures for automated verification of specific structural properties. Currently, models that satisfy these properties are considered semantically correct. According to our analysis, such verification methods cannot be considered as validation, since they do not address the business logic of the process and its goal. Rather, they implicitly assume that the business logic of the process should be known and well addressed by the human designer. The validity criteria, proposed as part of the GPM framework, provide support to the human reasoning that should address the business logic to be expressed in the process model. As such, these criteria can be considered as complementary to verification.

The paper compares the properties of soundness and robustness, addressed by verification algorithms, to the property of validity, and shows that soundness and robustness, which relate to pragmatic quality of the model (with respect to the purpose of executing the model), do not ensure validity, which relates to both semantic and pragmatic quality. In addition, the paper reports an empirical study which investigated the need for a systematic support to the application of business logic in process design as provided by the validity criteria, and their applicability.

The empirical findings clearly indicate the need for a systematic analysis rather than relying on simple unguided human reasoning. They show that the validity criteria are applicable even at an abstract form, so even novices are able to benefit from them and apply them to a concrete situation. In addition, the empirical findings indicate that when using EPC as a modeling language, an explicit representation provides a better support to the application of GPM's validity criteria, thus to the identification of validity problems.

Future research of the validity criteria may take several directions. First, as discussed above, it would be interesting to perform a similar study whose subjects are experienced analysts. Second, we may experiment with models of different sizes to evaluate the scalability of the approach. Finally, major efforts should be devoted to the development of a structured

methodology, possibly accompanied by tools, for supporting the application of the validity criteria.

References

- Aalst, W.M.P., Dumas, M. , Gottschalk, F. , ter Hofstede, A.H.M., La Rosa, M. & Mendling, J. (2009). Preserving correctness during business process model configuration. *Formal Aspects of Computing* (to appear).
- Aalst, W.M.P. (1998). The application of petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1), 21–66.
- Aalst, W. M. P. (1999). Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10), 639-650.
- Aalst, W. M. P. (2002). Making Work Flow: On the Application of Petri Nets to Business Process Management, *Application and Theory of Petri Nets 2002, LNCS 2360* (pp. 1-22). Berlin: Springer-Verlag.
- Aalst, W. M. P., Desel J. & Kindler E. (2002). On the Semantics of EPCs: A Vicious Circle. In Nüttgens, M. & Rump, F. (Eds.), *EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten* (pp.71-79). Trier: Germany.
- Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, 90, 129-149.
- Bray, K. (2002). *An Introduction to Requirements Engineering*. UK: Addison Wesley.
- Breyfogle, F. W. (2003). *Implementing Six Sigma: Smarter Solutions Using Statistical Methods*. Austin: John Wiley and Sons.
- Bunge, M. (1977). *Treatise on Basic Philosophy: Vol. 3, Ontology I: The Furniture of the World*. Boston: Reidel.
- Bunge, M. (1979). *Treatise on Basic Philosophy: Vol. 4, Ontology II: A World of Systems*. Boston: Reidel.
- Dehnert, J. & Aalst, W. (2004). Bridging the gap between business models and workflow specifications. *International Journal of Cooperative Information Systems*, 13(3), 289-332.

- Dehnert, J. & Rittgen, P. (2001). Relaxed Soundness of Business Processes. In Dittrich et. al. (Eds.), *Advanced Information Systems Engineering (CAiSE 2001)*, LNCS 2068 (pp. , 157-170). Berlin: Springer-Verlag.
- Dongen, B. F., Aalst, W. M. P. & Verbeek, H. M. W. (2005). Verification of EPCs: Using Reduction Rules and Petri Nets. In Pastor A. & Falcao e Cunha (Eds) *Advanced Information Systems Engineering*, LNCS 3520, Berlin: Springer-Verlag, 372-386.
- Grabski, S.V., Reneau J.H., & West S.G. (1987). A comparison of judgment, skills, and prompting effects between auditors and systems analysts. *MIS Quarterly*, 11(2), 151-162.
- Hee K., Oanea, O., Serebrenik, A., Sidorova, N. & Voorhoeve, M. (2008). History-based joins: semantics, soundness and implementation. *Data & Knowledge Engineering* 64, 24-37.
- Kindler, E. (2004). On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. In Desel, J., Pernici, B. & Weske, M. (Eds.), *Business Process Management, BPM 2004*. LNCS 3080. Potsdam, Germany: Springer-Verlag, 82-97.
- Kindler, E. (2006). On the semantics of EPCs: resolving the vicious circle. *Data and Knowledge Engineering*, 56, 23-40.
- Lindland, O.I., Sindre, G. & Solvberg, A. 1994. Understanding quality in conceptual modeling. *IEEE Software* 11 (2), 42–49.
- Lohmann, N., Massuthe, P., Stahl, C. & Weinberg, D (2008). Analyzing interacting WS-BPEL processes using flexible model generation. *Data & Knowledge Engineering*, 64. 38-54.
- Mendling J. & Aalst, W. (2007). Formalization and Verification of EPCs with OR-Joins Based on State and Context. In Krogstie J., Opdahl, A. L. & Sindre, J. (Eds), *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE)*, LNCS 4495 (pp. 439-453). Berlin, Germany: Springer-Verlag.
- Mendling, J. (2007). *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration.
- Mylopoulos, J. (1992). Conceptual modeling and telos. In P. Locoupoulos & R. Zicari (Eds.), *Conceptual modeling, databases, and cases*. New York, NY: John Wiley & Sons, Inc.
- Recker, J. (2007). A socio-pragmatic constructionist framework for understanding quality in process modelling. *Australasian Journal of Information Systems*, 14(2). 43-63.

- Recker, J., Rosemann, M., & Aalst W. (2005). On the User Perception of Configurable Reference Process Models – Initial Insights. *Proceedings of the 16th Australian Conference on Information Systems*. Sydney: Australia.
- Sadiq, S., Orłowska, M., Sadiq, W., & Cameron, F. (2004). Data Flow and Validation in Workflow Modelling. In Schewe K. D. & Williams H (Eds.), *Proceedings of the 15th Australasian database conference – Vol. 27*, Dunedin, New Zealand, Australia: Australian Computer Society, 207 - 214 .
- Sarshar, K. & Loos, P. (2005). Comparing the Control-flow of EPC and Petri Net from the End-user Perspective. In Aalst, W., Benatallah B., Casati F. & Curbera F. (eds.), *Proceedings of the 3rd International Conference on Business Process Management* (pp. 434-439). Nancy, France: Springer.
- Sawilowsky, S. (1990), Nonparametric tests of interaction in experimental design, *Review of Educational Research*, Vol. 60(1), pp. 91-126.
- Scheer, A.W. (1998). *ARIS-Business Process Modeling*. Berlin: Springer-Verlag.
- Shah, D. A. & Madden L.V. (2004). Nonparametric analysis of ordinal Data in designed factorial experiments. *Phytopathology*, 94(1), 33-43.
- Siau, K. & Tan, X. (2005). Improving the quality of conceptual modeling using cognitive mapping techniques. *Data and Knowledge Engineering*, 55, 343-365.
- Soffer, P. & Wand, Y. (2004). Goal-Driven Analysis of Process Model Validity. In Pearsson, A. & Stirna, J. (Eds.), *Advanced Information Systems Engineering (CAiSE'04)*, LNCS 3084 (pp. 521-535). Berlin: Springer-Verlag.
- Soffer, P. & Wand, Y. (2005). On the notion of soft goals in business process modeling. *Business Process Management Journal*, 11(6), 663-679.
- Soffer, P. & Wand, Y. (2007). Goal-driven multi-process analysis. *Journal of the Association of Information Systems*, 8(3), 175-203.
- Soffer P., Kaner, M. & Wand Y. (2008). Assigning Ontology-Based Semantics to Process Models: the Case of Petri Nets. In Bellahsene Z. & Leonard M. (Eds.). *Advanced Information Systems Engineering (CAiSE 2008)*, LNCS 5074 (pp. , 16-31). Berlin: Springer-Verlag.
- Sommerville I. (2007) *Software Engineering* (8th edition). Harlow, England: Addison-Wesley.

- Vanderfeesten, I., Reijers, H.A., Mendling, J., Aalst, W., & Cardoso, J. (2008) "On a Quest for Good Process Models: The Cross-Connectivity Metric", In Bellahsene Z. & Leonard M. (eds) *Advanced Information Systems Engineering (CAiSE 2008)*, LNCS 5074 (pp. 480-494). Berlin: Springer-Verlag.
- Verbeek, H. M. W. & Aalst W.M.P. (2006). *On the Verification of EPCs Using T-invariants*. BPM Center Report BPM-06-05, BPMcenter.org.
- Verbeek, W.M.P, Aalst, W.M.P & Hofstede A.H.M. (2007). Verifying workflows with cancellation regions and OR-joins: an approach based on relaxed soundness and invariants. *Computer Journal*, 50 (3), 294-314.
- Wand, Y. & Weber, R. (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering*, 16(11), 1282-1292.
- Wand, Y. & Weber, R. (1995). Towards a theory of deep structure of information systems. *Journal of Information Systems*, 5(3), 203-223.
- Weber, R. (2004). Conceptual modelling and ontology: Possibilities and Pitfalls. *Journal of Database Management*, 14 (3), 1-20.
- Zur-Muehlen, M. & Rosemann, M. (2004). Multi-Paradigm Process Management. Proceedings of Workshop on Business Process Modeling, Development and Support (BPMDS'04), *CAiSE'04 Workshops Proceedings 2*, 169-175.