Requirements Eng (2005) 10: 184–197
DOI 10.1007/s00766-005-0003-8

ORIGINAL ARTICLE

Anne Etien · Colette Rolland

Measuring the fitness relationship

Received: 31 August 2004 / Accepted: 10 May 2005 / Published online: 19 August 2005

Abstract It is widely acknowledged that the system functionality captured in a system model has to match organisational requirements available in the business model. However, such a matching is rarely used to support design strategies. We believe that appropriate measures of what we refer to as the fitness relationship can facilitate design decisions. The paper proposes criteria and associated generic metrics to quantify to which extent there is a fit between the business and the system which supports it. In order to formulate metrics independent of specific formalisms to express the system and the business models, we base our proposal on the use of ontologies. This also contributes to provide a theoretical foundation to our proposal. In order to illustrate the use of the proposed generic metrics we show in the paper, how to derive a set of specific metrics from the generic ones and we illustrate the use of the specific metrics in a case study.

Keywords Fitness relationship · Criterion · Metric · Alignment · Bunge's ontology

1 Introduction

Although, business process and information technology have been recognised to be natural partners [1–5] their relationship has not been fully exploited [6]. This relationship has been referred to by different terms such as "alignment", "fitness", "match", "adaptation", "correspondence". In this paper we will use the term fitness relationship [4]. We adopt the view of Regev [7], who defines this concept "as the correspondence between a set of components". Following Nadler [8] we consider fitness to be "the degree to which the needs, demands, goals, objectives and/or structure of one component are consistent with the needs, demands, goals, objectives and/or structure of another component".

The foregoing suggests (a) a precise identification of the types of correspondence between components of the business and system models and (b) a measurement of the degree of correspondence. In this paper we attempt to address both of these.

In dealing with issue (a), we base our proposal on the use of ontologies for representing business and system components, respectively. For the former kinds of components we use the Soffer and Wand [9] ontology whereas at the system level we use the Wand and Weber (WW) ontology [10]. These two ontologies are adaptations of Bunge's ontology [11, 12] which are largely recognised for its theoretical foundations. The use of ontologies is a way of becoming independent of the business and systems models, thereby leading to a generic expression of the component correspondence. In addressing issue (b), we propose the use of fitness criteria and associated metrics to measure these. This is in the spirit of work done on impact analysis [13] but takes its roots in metrics definition in software engineering [14, 15]. The result is a set of generic fitness measures associated to correspondences between constructs of the SW and WW ontologies.

Whereas, the key scope of the paper is to present these generic measures and their rationale, we also illustrate their use considering part of an industrial case study. For this purpose, correspondences between the ontologies and each of the two specific models are established. The specific measures are then, derived from the generic measures and applied to the case study.

The rest of the paper is organised as follows. Section 2 presents an overview of the two ontologies that are used. Section 3 defines the correspondences between elements of these ontologies through two types of mappings. In Sect. 4 the fitness criteria and associated metrics are presented. Section 5 illustrates the use of these metrics in a concrete case. Finally, conclusions are drawn in Sect. 6.

A. Etien (✉) · C. Rolland
CRI University Paris1, 90 rue de Tolbiac,
75013 Paris, France
E-mail: aetien@univ-paris1.fr
Tel.: +33-1-44078634
Fax: +33-1-01-44078954
E-mail: rolland@univ-paris1.fr

2 Overview of the ontologies

This section presents an overview of the WW ontology for the system level and the Soffer and Wand (SW) ontology underlying business models.

2.1 Wand and Weber ontology

The WW ontology [10] allows to model a software system. It is the first ontology defined from Bunge's works [11, 12] and using his concepts. Thus, according to this ontology, the real world is made of *things* that (1) possess *properties* and (2) can be composed of other *things* and can thus be qualified as *composite*. Properties can be *intrinsic* (e.g. height of a person), *mutual* to several things (e.g. a person works for a company) or *emergent* which refers to the property of a composite thing but not possessed by individuals composing it.

Properties are perceived by humans in terms of *attributes*, which can be represented as functions on time. For a given *thing*, the set of values of all its attribute function is called its *state*. Properties can change and this is expressed by a state change or *event*. However, as not all states are possible and not all changes can occur, there exist rules governing possible states and state changes called *state laws* and *transition laws*, respectively.

Two things are coupled if the history of one thing depends upon the history of the other. A *system* comprises a set of things where each thing in the set is coupled to at least one other thing in the set. The *environment* is the set of things that do not belong to the system but act on or are acted upon by the system.

To these concepts directly adapted from Bunge's works, Wand and Weber added other ones that are derived from the former. Thus, a class is a set of things that possess a common property. Since an attribute represents a property that all things of a class possess, it specifies a *property in general* of this class. In the simple sales accounting system described by Wand and Weber in [16]. The attribute 'unit-selling price' representing a property that all product items possess, specifies a property in general of the product items class. However, the specific unit-selling price of item C123 (e.g. 0.87€) is a property in particular of that thing [17]. A state is *stable* if it can only change as a result of an action of something outside the domain i.e. the part of the world changes that we want to model. An *unstable state* is a state that must change. A *law* is defined as a function from the set of states into itself. A *transition law* is a set of the possible unstable states into the set of states. A thing is an *input* of a system if it is acted upon by a thing belonging to the environment, whereas, an *output* of a system acts on an environmental thing. An *external event* is a change of state where the reached state is an input. A change of state where the reached state is not an input is an *internal event*.

All these constructs are modelled in Fig. 1 and are detailed in [10, 18, 19].

2.2 Soffer and Wand ontology

The SW ontology [9] allows the description and formalisation of a business model. It relies on the WW ontology but introduces new concepts to deal specifically with some particularity of business processes. The concepts of *things, properties*, *classes*, *attributes*, *states, events*, *laws* and *transitions* as well as the typologies of properties, states and events defined by WW are identical in the SW ontology. New concepts introduced are as follows: a *goal* is a set of stable states; a *process* is a sequence of unstable states leading to a stable states; a goal; *ordered activities* are state transitions caused by transformations defined by the law; *actors* are classes that take actions in response to their state changes and *resources* are classes that take no further actions.

The SW's view of the business world is represented in Fig. 2 and more details can be found in [9].

3 Correspondence links between ontology constructs

In this section we introduce two types of links, namely *maps* and *represents* to establish a construct correspondence across the SW and WW ontologies and suggest a set of correspondence links to *represent* constructs of the SW ontology by constructs of the WW ontology. We base our proposal on a review of link typologies proposed in the literature and considered below.

3.1 Overview of literature

In [20], a relation between two modelElements that belong to different method models can be either an Association, a Composition or an Is-A link. In our case, in so far as the two models represent two different modelling domains, it is clear that the Composition and Is-A links do not hold. However, it does not mean that there necessarily only exists Association links between SW and WW ontology constructs. Indeed, the Association link can be specialised.

In [21], Pohl developed a dependency meta-model with five categories and 18 types of possible dependency links that can exist between any information objects (e.g. agent, product state, process step, requirement...). These links correspond to the way information objects affects other (e.g. positively or negatively and in Pohl's case information objects can be in conflict).

Based on these different proposals, we define two types of correspondence links namely, *maps* and *represents* between a SW ontology construct and a WW ontology construct.

3.2 Maps and represents correspondence link types

In this section, we successively present the *maps* and the *represents* links.

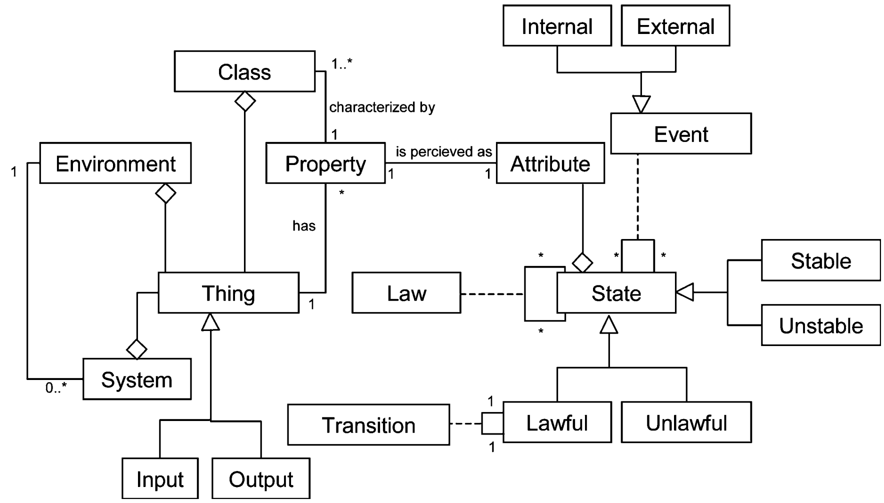

Fig. 1 The Wand and Weber ontology modelled with UML

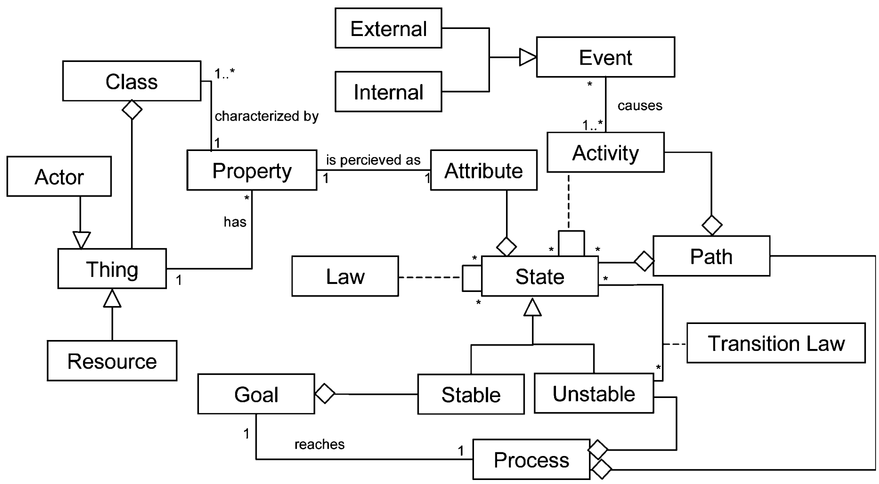

Fig. 2 The Soffer and Wand ontology modelled with UML

3.2.1 Maps link

This link (1) is close to the *similar* link of Pohl and the similarity notion of Bunge and (2) is a specialisation of the association link presented by Ralyté and al.

As each of the two SW and WW ontologies are specialisations of the Bunge's ontology some constructs are present both at the business level and at the system level. It is thus possible to define a mapping between constructs of the same nature.

Thus, one class X maps $\mathscr{M}$ another class Y if there exist an isomorphism between the set of properties of X ($p(x)$) and the set of properties of Y, $p(y)$. In other terms, each property of X corresponds to one of Y even (domains being eventually different). The existence of an isomorphism and not only a homomorphism allows to specify that the classes X and Y play the same role. We use the $\approx$ symbol to specify the existence of such an isomorphism.

$$X\mathscr{M}Y \Leftrightarrow p(x) \approx p(y).$$

From this definition, it can be deduced that this relationship is reflexive and thus if a class X *maps* another class Y then, Y also *maps* X.

Such a mapping can be extended to other constructs than a class. Thus, for example, a state X at the business level *maps* a state Y at the system level if (1) two classes *map* each other and (2) if it exists an isomorphism between the set of values of all the attribute functions of one class with the set of values of all the attribute functions of the other class. This means that not only the set of attribute functions of the two classes correspond (isomorphism between properties) but also the set of values of these attribute functions.

In the same way, two laws or two events *map* each other if each of the two states occurring in the law (respectively event) at the business level *maps* a state of the law at the system level.

3.2.2 Represents link

The *represents* link is a specialisation of the association link between modelElement of different models introduced in Ralyté and al. This link is also close to both the *satisfies* and the *is_based_on* links defined in Pohl.

Thus, a construct of an ontology *represents* ($\Re$) a construct of the other ontology if the existence of the former affects the behaviour, the value or the existence of the latter. $X \Re Y \Leftrightarrow X$, where $\triangleright$ signified that X affects the behaviour of Y.

It is important to notice that two constructs of different nature can be linked through a representation link. For example, a class can *represent* a property.

Clearly, the *represents* link is weaker than the *maps* link. Thus, two constructs that *map* each other also *represent* each other. The opposite is not true because two constructs of different nature can be linked through a representation link even though a mapping link cannot exist.

All possible *maps* and *represents* links constitute the baseline elements to measure the fit or unfit between a business model and a system model. The criteria for measuring and metrics associated with these links are developed in the next section.

4 Fitness criteria and metrics

We use the Cavano and McCall framework [15] to develop a system of fitness measurement. This framework organising software quality measure is divided into three levels, *factors, criteria* and *metrics*. Factors are characteristics that can be appreciated on an external point of view. They can be used as an aid in specifying software quality objectives. These high-level factors are then broken down into criteria that are more software directed. These latter correspond to the product characteristics and represent the internal and technical point of view. Metrics allow to measure a criterion. The Cavano and McCall framework is still now a reference in the domain of the quality measure and has inspired lot of other frameworks as those of Boehm [14], the ISO quality standard [22]...

In our framework, we identify four factors along which the fit can be measured namely, the *intentional factor*, the *informational factor, the functional factor*, and the *dynamic factor*. These factors reflect the four perspectives that have been reported in IS literature, namely, the holistic view brought by the goal-actor-resource-process perspective; the information perspective; the functional, law-based perspective and the dynamic perspective. They can be used to aid in specifying fit objectives. Each factor has associated *criteria* corresponding to fit characteristics. They are in turn, related to *metrics* that allow the actual computation of the degree of fit. Criteria and metrics are based on *maps* and *represents* links between constructs of the SW and WW ontologies as defined in the previous section.

Table 1 provides an overview of the proposed fit measurement system. As shown in the table it comprises 10 criteria and 10 metrics grouped along the four identified factors. Metrics are summed up in the fourth column of the table. Components used in the metrics expression are marked in italics whereas the type of link (*maps* or *represents*) is shown in bold. A formal definition of each metric is given in Appendix.

In the criteria and metrics, the business view is taken as a reference and is compared with the system view. Therefore, in any metric the denominator always refers to constructs of the SW ontology. The value of the metrics can be modified with an evolution of the business or the system. It is true that mostly, in practice, when corrective actions are applied in order to raise up the metrics values, the system is changed. Nevertheless, system can correspond to 'good practice' (for example in case of ERP) or it can occur, for example that, with a platform changes, classes that were not possible before can henceforth be implemented. In those cases, corrective actions occur on business.

In the sequel we consider each factor in turn and comment its associated criteria and metrics. For sake of space (and except for the first criterion) we limit ourselves to comments on the meaning and potential use of the criterion to reason about the system/business fit. The precise definition of the metrics can be found in Appendix.

4.1 Criteria and metrics related to the intentional factor

Along the intentional dimension, the objective is to measure to which extent the system is meeting the business purpose. This is achieved by providing four criteria associated to the *intentional factor* dealing, respectively, with the business activity and the goal support, and the actor and resource representation.

4.1.1 Support ratio

The *support ratio*, similar to the technological coverage criterion defined by Bodhuin [13] is the extent to which business activities are supported by the system. The higher this ratio is, the more automated the activities are. Conversely, a low-support ratio expresses that a large number of business activities are manually carried out. In order to raise this ratio it is necessary to automate some activities and thus, to introduce in the system new classes, properties, laws, events to support these business activities.

An activity is supported by the system if there exists an event in the system that *represents* it. Thus, the support ratio is as follows:

Table 1 Fitness measurement system

Factors	Criteria	Metrica	Comments
Intentional	Support ratio	Activity count	Number of activites represented by system events/Number of activites
	Goal satisfaction	Goal count	Number of goals for which each state in the system/Number of goals
	Actor presence	Actor count	Number of business actors mapping a system class/Number of business actors
	Resource presence	Resource count	Number of business resources mapping a system class/Number of business resources
Informational fit	Information completeness	Business/System class mapping count	Number of business classes mapping system classes/Number of business classes
	Informational accuracy	Business/System state mapping count	Number of business states mapping to system states/Number of business states
Functional fit	Activity completenes	Business/System class mapping count	Same as information completeness but for one given activity
	Activity accuracy	Business/System state mapping count	Same as information accaracy but for one given activity
Dynamic fit	System reliability	Law-mapping count	Number of business laws for which each business state maps a system state and the transformation between business states are possoble between system states/Number of business laws
	Dynamic realism	Path mapping count	Number of paths for which each business state maps a system state and the succession of these system states is possible/Number of possible paths

$$\text{Support ratio (Sr)} = \frac{\text{Number of activities } \textbf{represented} \text{ by system events}}{\text{Number of activities}}$$

Let:

- A_b be the set of business activities (i.e. activities present in the business process), card(A_b) = the number of elements[1] contained in A_b.
- E_s be the set of system events
- A_b^r be the set of business activities for which there exists a system event representing it; $A_b^r\{a, a \in A_b | \exists e \in E_s e \Re a\}$ and card($A_b{}^r$) = number of elements contained of A_b^r

Using these notations, the metric associated to this criterion is:

$$\text{Sr} = \frac{\text{card}(A_b^r)}{\text{card}(A_b).}$$

We defined all metrics (see Appendix) in a similar way this was presented above for the support ratio criterion.

4.1.2 Goal satisfaction

Clearly, evaluating to which degree the system meets the business goals is an essential part of the fitness relationship measurement. The presence of the *goal* construct and its relationship with processes in the SW ontology reflects the importance of goals in business modelling.

The *goal satisfaction criterion* is proposed to measure the extent to which goals are supported by the software system. To evaluate this, we consider that the states associated to goal satisfaction in the business model shall also be represented as states in the system. For example, if for a booking process there is a goal specifying that "booking has to be paid", then the state 'paid' for the class booking must be part of the system model. In the case when it is not, the goal is not supported by the system and the booking process looses its purpose.

The *goal satisfaction metric* is defined as the ratio of goals supported by the system to the goals in the business. A goal is supported by the system if each business state associated to the goal *maps* to a state in the system. A low-goal satisfaction ratio indicates that a number of goals cannot be achieved with the use of the system. On the contrary, a high ratio shows that the system fits the need for business goal satisfaction.

4.1.3 Actor and resource presence

Business processes involve *actors* and use *resources*. Therefore, it seems relevant to measure the extent to which these are represented in the system. We therefore, introduce two criteria namely, the *actor presence* and *resource presence* criteria.

Actor presence According to the SW ontology, an *actor* is a class that triggers a state transition on another class. A client that makes an order in the case of an e-commerce application on the Internet is an *actor*. It provokes the creation of an order (and thus a state change of the class order). However, some of its prop-

[1]If S is a set of elements, card(S) refers to the cardinality of S and corresponds to the number of elements contained in S.

erties can be used in another process as for example, his/her address for delivery. Thus, it seems important to check that business actors are present in the system in order (1) to trigger state transitions and (2) to permit the use of their properties in the system. This is the role of the *actor presence criterion*.

A low value of this criterion means that the causes of some state transitions are not the same in the business and in the system. In order to increase its value, it is possible (1) to remove the business actor who is not present in the system if the check demonstrates that he does play a significant role in the business process or, (2) to introduce the actor in the system.

Resource presence According to SW ontology, a *resource* is a class that takes no further action. Thus, a business resource is supported in the system if it *maps* to a system class that takes no further action. For example, in an ATM project, a credit card is a business resource that does not change state. In order to memorise some properties of the card such as the credit card number and the validity date it is necessary to have a class in the system, which *maps* the credit card resource.

A low value shows that numerous business resources are not mapped to system classes. This means that either some resources play a role in the business which does not need to be known by the system or that the correspondence of some of them in the system is missing. In contrast, a high value demonstrates that the majority of the business resources are known by the system. No indication is given relatively to the way they are used.

4.2 Informational factor

The *intentional factor* and its related criteria defined above are focusing on the evaluation of the degree to which business activities are supported by the system and the extent to which the system allows business goals fulfilment. The *information factor* which we consider now, complements the former by a deeper analysis of the way activities are supported in the system. In order to provide a good fit between the system and the business processes, the system must (1) manipulate all the business process objects and (2) support all the business process object states associated to the business processes. Two criteria have been defined in order to permit such an evaluation, the *informational completeness* and the *informational accuracy*, respectively as shown in Table 2.

4.2.1 Informational completeness

According to the SW and the WW ontologies, a business process and a system manipulate information through classes. Obviously, to avoid a misfit between the business and the system there should be a strong correspondence between the business information and the system information. This can be measured by the fact that there exist classes in the system that *map* business classes. The *informational completeness criterion* measures the proportion of this mapping. If this proportion reaches unity, it means that the information is completely managed by the system; the term informational completeness is then used.

A high value computed by the metric shows that the information system really meets its purpose as it manages the information needed to support business processes. This value might diminish if for example, new classes are introduced in the business. Corrective action should be taken to increase the *information completeness* ratio either by modifying the system to manipulate the new classes or by removing business classes if they demonstrate to be of low value.

4.2.2 Informational accuracy

Clearly, it is important that the business process and the supporting system manage the same information, but it is not sufficient. Intuitively, it seems that the information has to be managed in the "same way" at the business process and system levels. The "same way" means for us, that the states of the business *map* states in the system. Thus for example, in the case of order management, it is necessary for the system to recognise the state "satisfied" of an order because events and thus activities depend on this state. If such a state does not exist in the system, the order will not be delivered, the payment will not be asked for and the stock will not be managed.

A low informational completeness ratio implies low informational accuracy because the *maps* links used for accuracy measurement apply to states of classes and therefore, depend on the ones on classes. Furthermore, if a business class is not supported by the system then neither are its states. However, the contrary is not true. Business classes can be mapped to system classes without a similar mapping for their states. Such a situation detected due to the information accuracy measure implies that corrective actions need to be taken because it shows that lifecycles, business events, transition laws, or paths of the business could not be enacted.

Table 2 The criteria of the informational factor and their associated metrics

Factors	Criteria	Metrica	Comments
Informational fit	Information completeness	Business/System class mapping count	Number of business classes mapping system classes/Number of business classes
	Informational accuracy	Business/System state mapping count	Number of business states mapping to system states/Number of business states

Table 3 The criteria of the funtional factor and their associated metrics

Factor	Criteria	Metrics	Comments
Funtional fit	Activity Completeness	Business/System class mapping count	Number of busniss *classes* mapping system *classes*/Number of bussiness *classes*
	Activity Accuracy	Business/System state mapping count	Number of busniss *states* mapping system *states*/Number of bussiness *states*

4.3 Functional factor

The *functional factor* aims to measure the degree to which activities in the system correspond to business activities. The correspondence is based on involvement of classes and their states in business and system activities, respectively. The functional factor has two criteria associated to it, *activity completeness* (Ac) and *activity accuracy* (Aa). Both apply to each individual activity of a business process, separately. Informal definitions of these two attributes are specified in Table 3.

4.3.1 Activity completeness

This criterion allows to enter into the details of each business activity. It expresses the degree to which a given business activity is supported by the system by measuring the supported classes of the activity.

4.3.2 Activity accuracy

As above for the completeness criterion, this criterion measures the degree to which a given activity is supported by the system. It is based on a *maps* link between a business state and a system state involved in a business process activity and system activity, respectively.

Similar to the *information and accuracy criteria*, there is here, a dependence between *activity completeness* and *activity accuracy* measures as accuracy holds only for those activities which fulfil completeness. The accuracy criterion therefore, provides a finer grained measure of the fit than the completeness criterion. Misfit, according to the *activity completeness* criterion, requires checking actions to verify if the lack of completeness is justified whereas a low value computed by the activity accuracy metric signals some inconsistency in the representation of the activity in system terms that requires to be corrected.

4.4 Dynamic factor

The fourth factor, the *dynamic factor* aims to evaluate the extent to which business processes are reflected in paths of system state transitions. It has two criteria namely the *system reliability criterion* and the *dynamic realism criterion* (cf. Table 4).

4.4.1 System reliability

The two previous factors take into consideration classes and/or the states independently as well as separately. The *system reliability criterion* considers the succession of system states, activities and events to check their fit with the corresponding business succession. The fit is based on the *maps* relationship between these. Indeed, an absence of mapping reveals errors. Assume for example, that in a business process orders are paid only if their state is ‘satisfied’. Even if using previous measures it is known that the system manages orders, knows the states ‘satisfied’ and ‘paid’ for an order, nothing guarantees that the system respects the state constraint for triggering the order payment, i.e. applies a law which *maps* the business law.

The *system reliability* criterion is defined to measure the fit between business laws and systems laws. The associated metric computes the ratio of business laws that are supported in the system, i.e. those for which there exists a *mapping* with a system law.

A low value of the *system reliability criterion* shows that the system will not behave as a mirror of the business performance as business laws are poorly mapped to system laws. Such a situation is critical and calls for corrective design actions.

4.4.2 Dynamic realism

Complementarily to measuring the fit of business and system laws, it is important to evaluate if the succession of

Table 4 The criteria of the dynamic factor and their associated metrics

Factor	Criteria	Metrics	Comments
Dynamic Fit	System Reliability	Law-mapping count	Number of *business laws* for which each business state maps a system *state* and the transformation between business states are possible between system states/Number of *business laws*
	Dynamic Realism	Path mapping count	Number of *paths* for which each business state maps a system *state* and the succession of these system states is possible/Number of possible *paths*

business activities is adequately supported by the system. This leads to look for correspondence between a path of business activities and a path of system state transitions. The *dynamic realism* criterion is defined for this purpose. A path is considered as supported by the system if each state constituting the path *maps* to a state in the system and if the business law *maps* to a system law.

A low level of the value computed by the metric shows that the system does not provide what is expected at the business level. To increase this value, either the business model has to be modified keeping the system constant or vice versa. Besides, it shall be noticed that the reason for a low value might be low values of other criteria. Corrective actions should examine the latter before considering the former.

5 Generating specific metrics from generic metrics

In this section, we aim to illustrate the use of the generic metrics presented in the previous section. Because of their genericity, the 10 metrics defined above cannot be used 'As Is' but must be adapted to the specific constructs and correspondences existing between the two specific models that are used to represent the business and the system, respectively. Our proposal consists in deriving the specific metrics from the generic ones. This derivation is based on correspondences between the two ontologies and the two specific models. The generated specific metrics can be used then, to actually measure the degree to which a system fits its related business. In this section, we consider the MAP [23] model to represent the business and the UML class and state transition diagrams to represent the system. We first show how the specific metrics for these two models can be generated and then, use them to measure the degree of fit in a case study.

5.1 Generating specific metrics

The process to generate specific metrics from the generic ones is shown in Fig. 3. This process proposes to establish a liaison between the set of the specific business model constructs (respectively system model) and the SW ontology set of constructs (respectively the WW ontology) and then, derive the formulation of specific metrics.

We illustrate this process by (a) demonstrating the WW ontology correspondences with the UML constructs embedded in the class and state transition diagrams, (b) summing up the correspondence process for the MAP model and (c) deriving the set of specific metrics for these two models.

5.1.1 Correspondences between WW and UML constructs

Associating the UML class diagram and UML state transition diagram allows us to describe a system from both a static and a dynamic perspectives. The static part provides description of durable links between instances of constructs (i.e. composition) whereas the dynamic part models behavioural interactions inside the system or with its environment (i.e. in response to events). Figure 4 presents the relationship between WW constructs and UML concepts. The names of WW constructs and particularly those involved in metrics are noted in the right corner of the boxes that represent the equivalent UML concepts.

The *class* and *attribute* components in the UML class diagram respectively relates to the WW *class* and *property* constructs. Both *properties* and *attributes* see their *states* change in response to *events*. The *event* in the UML state transition diagram can be *external* (an external event has for origin a stimulus coming from the environment of the system. It is associated to an actor), *internal* (it responds to a particular state change of a system object). The notion of *transition*, which changes states, in the UML state transition diagram corresponds to the *transformation* construct in the WW ontology. Table 5 below sums up the correspondence between the WW constructs involved in metrics and the UML constructs.

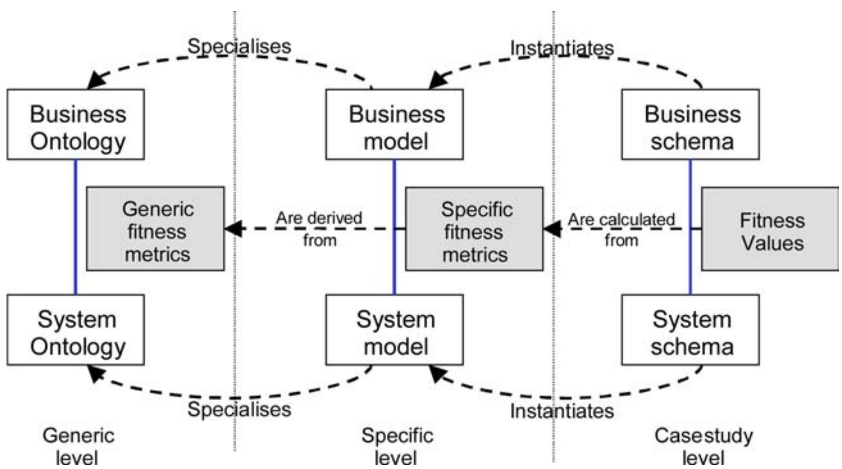

Fig. 3 Framework illustrating the use of the metrics

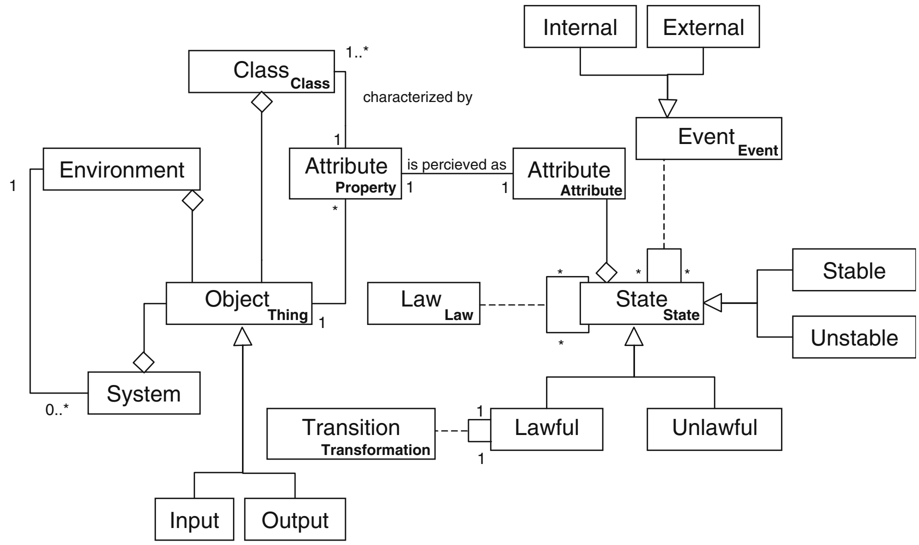

Fig. 4 Relating class diagram and state transition diagram to WW

Table 5 Wand and Weber and UML constructs correspondence

WW constructs	UML constructs
Thing	Class
State	State
Event	Event
Transformation	Transition
Property	Attribute

Table 6 Soffer and Wand and MAP constructs correspondence

SW constructs	MAP constructs
Process	Map
Activity	Section
Goal	Intention
Law	Strategy
Thing	Object
State	State
Actor	Actor
Resource	Resource
Path	Path

5.1.2 Relationships between SW and MAP constructs

Following the same way-of-working, the correspondence between SW constructs and MAP constructs can be obtained as shown in Table 6.

The MAP model allows to model processes in intentional terms. The key concepts of MAP are (1) *intentions* (goals to achieve or maintain) that are formally defined as desirable states and corresponds to the SW goal construct; (2) *strategies* (means or manners to attain a goal), which ensures a mapping between subsets of states, hence it specifies the SW *law* and (3) *sections* (triplets $< I_i, I_j, S_{ij} >$ where $I_i$is the source intention, I_j the target intention and S_{ij} the strategy to attain when I_i has been achieved) that relates to the SW notion of *activity*. MAP provides a representation mechanism based on a non-deterministic ordering of intentions and strategies. A map is a labelled directed graph with intentions as nodes and strategies as edges between intentions. The directed nature of the graph shows which intentions can follow which one. An edge enters a node if its strategy can be used to achieve the intention of the node. The *map* construct in MAP corresponds to the *process* construct in SW. *Objects* appear in the map specification as parameters of the intention and strategy linguistic formulation [24]. A process model expressed as a map contains discontinuity related to *events* that can be *external* (a map is triggered by an *external event* arising in a class and resulting from the action of some other class called *actor)* or *internal* (section triggering). In both models the process is viewed as a *path*.

5.1.3 Metrics specific to UML and MAP

Based on the relationship between the constructs of the SW ontology (respectively WW) and the MAP concepts (respectively UML) captured in Table 6 (respectively in Table 5), the conversion of generic metrics is easy to carry out. It simply consists in replacing in the metric formula the generic construct by its corresponding specific construct. Table 7 sums up the 10 specific criteria and metrics to measure the fit between a business expressed in MAP terms and the system specification designed with the UML class and state transition diagrams.

Let us consider the specific *Support ratio* metric in comparison to the generic one given in Sect. 6.

In generic terms, the support ratio metric measures the degree of correspondence between the number of activities represented by system events and the total number of activities in the business process. At the specific level, the measure is established between the number of *map sections* ***represented*** by *events* and the total number of *map sections*. The generic formula provided in section IV is adapted as follows: Let:

Table 7 Specific metrics for measuring business/system fit modelled in MAP and UML terms

Criteria	MAP constructs	UML constructs	Comments
Support ratio	Section	Event	Number of sections represented by events/Number of sections
Goal satisfaction	Intention	state	Number of intentions for which each state maps a state in the system/Number of intentions
Actor presence	Actor	Class	Number of business actors mapping a system class/Number of business actors
Resource presence	Resource	Class	Number of business resources mapping a system class/Number of business resources
Information completeness	Object	Class	Number of business objects mapping system class/Number of business objects
Information accuracy	State	State	Number of business states mapping to system states/Number of business states
Activity completeness	Objects	Class	Same as information completeness but for one given section
Activity accuracy	State	State	Same as information completeness but for one given section
System reliability	Law	State	Number of business laws for which each business state maps a system state and the transformation between business states are possible between business states are possible between system states/Number of business laws
Dynamic realism	Path	State	Number of paths for which each business state map a system state and the succession of these system states is possible/Number of possible paths

– S_b be the set of business sections card(S_b) = the number of elements contained in S_b.
– E_s be the set of system events
– S_b^r be the set of business sections for which it exists event representing them; $S_b^r = \{s, s \in S_b | \exists e \in E_s \wedge e \Re s\}$ and card(S_b^r) = number of elements contained in S_b^r

$$\mathrm{Sr} = \frac{\mathrm{card}(S_b^r)}{\mathrm{card}(S_b)}$$

The nine other metrics can be precisely generated in a similar way. We illustrate the use of them in the following Renault/DIAC case study.

5.2 Illustrating the use of metrics

DIAC is a subsidiary of the financial branch of Renault, the French car constructor. The overall objective of the DIAC branch is to sell financing products (credits and leases) to help customers buying Renault vehicles. DIAC main business goal is split into pre-sales (concerning catalogues, marketing, and making contracts) and post-sales (i.e. treasury, coordination with partners, customer management). We consider here part of the post-sales activity, the repayment of loans.

Each contract defines a schedule of monthly repayments. DIAC monitors repayments, checks late or incomplete payments and may impose penalties. A contract can be (1) closed normally, i.e. after total loan repayment, (2) terminated by anticipation (when, for example, the client asks for paying several monthly repayments at the same time) or (3) cancelled at the client initiative within the legal revocation period. In every case, ending a contract triggers different activities such as recording administrative data, managing residual debt either by friendly agreement or by bone of contention, archiving contracts to respect the 10 years legal storage obligation.

After noticing a real drop in productivity, the DIAC executive managers decided to carry out an internal audit. We suggested to use metrics in order to evaluate the extent to which problems are related to a low level of support of the business by the legacy information system. As the current situation was poorly documented, revision was first necessary to get to-date models representing the As-Is situation. We proposed to develop a high-level, intentional view of the current situation using MAP. A complete documentation was produced in about a month. The document is around 120 pages long and describes 34 models. The map in Fig. 5 is an excerpt of this document that models the loan repayment process in intentional terms.

From the models of the business and those of the system, we identified each instance of the concepts involved in the different metrics. Thus, for example, 'client' and 'contract' are Map business objects, "manage the schedule repayment by payment" is an activity 'payment' and 'contract' are classes and so on. We proceeded in this way for all elements. As a result, the process is composed of activities; goals are defined in terms of objects states, etc... The metrics can then be applied based on their definitions. For instance, for the information completeness metric, we count the total number of business objects and compare it to the business objects mapping a system class. In our case study, each business object has properties that correspond to those of a system class. This means that each business object maps a system class. The information completeness equals one. On the 12 sections of the map presented in Fig. 5 only 10 are *represented* by system events. Indeed, the two archiving activities are manual and thus

Fig. 5 Manage the customer relationship by loan recovery

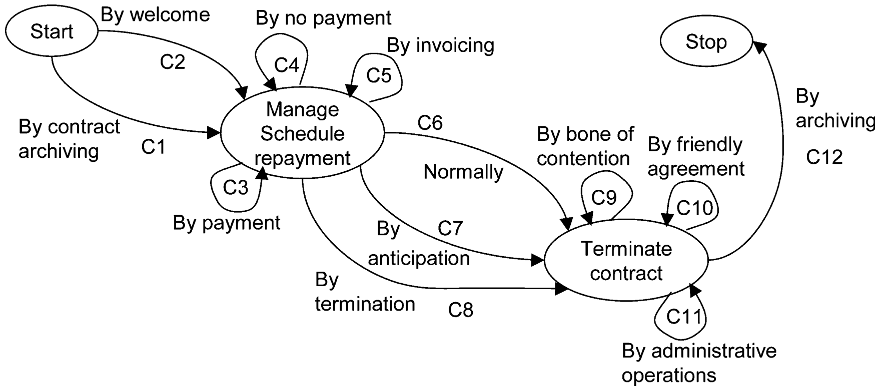

Table 8 Fitness measures obtained in the DIAC project

Criteria	Measures
Support ratio	0.83(10/12)
Goal satisfaction	0.67(4/6)
Actor presence	1(2/2)
Resource presence	0.75(3/4)
Information completeness	1(7/7)
Information accuracy	0.8(24/30)
Activity completeness	1(6/6)
Activity accuracy	0.85(11/13)
System reliability	0.82(27/33)
Dynamic realism	0.60(55/92)

not supported by the system. The support rate thus equals 10/12. Furthermore, some business states do not map any system state as the states archived or anticipated of the contract, the state reimbursed for the payment etc. This impacts several criteria as the Information accuracy or the Dynamic Realism. A detailed study of the latter allows to realise that only 55 among the 92 business processes are implemented in the system. A similar approach is used to calculate the other values of Table 8.

Clearly, a rapid overview of Table 8 shows that a good fit is not reached. A number of measures are inferior to one. The *support ratio* shows that two activities (concerning archiving, represented in Fig. 5 by sections C1 and C12) are not supported by the system. Indeed, archiving is currently manually done: a person has to copy and class contracts. Complementary, the measure of *activity accuracy* (calculated for the payment activity) highlights that (1) some business states are not supported by the system and (2) even if, business classes *map* system classes (*information* and *activity completeness* equal one) their state mapping is low: for example, the system cannot consider a monthly due repayment as 'partially paid' whereas such situations occur in practice. The absence of mapping between system/business states in this case has costly implications (1) the sales person must trigger a no payment procedure (modelled in section C4) and (2) employees must carefully look for identifying these situations and taking actions. This misalignment has consequences on the dynamic fit as well; this can be deduced from the measures of the *system reliability* and *dynamic realism* as shown in Table 8.

An analysis triggered by the low value of the *dynamic* realism metric shows a low fit between business practice and system implementation. It was noticed that the invoice is issued before the payment is handled whereas in practice the latter can precede the former. This situation complicates the work of clerks in charge of contract payments as they must manually carry out corrective actions.

The case study demonstrated that the proposed measures helped stakeholders to identify problems related to the lack of information system support to some parts of the business process. Based on the reasoning triggered by the measures, it was possible to identify possible improvements and make decision accordingly. It was for example, decided (a) to buy an electronic data manager in order to electronically archive contracts and (b) to manage partial payments and compensations with the system support. Besides, based on the analysis of the dynamic factor measures, the implementation of other business laws was decided. In addition to triggering direct decision for improvements, the fit analysis was also, indirectly useful. Indeed, it helped eliciting business changes that could benefit to the company. Providing to car buyers additional financial services such as personal loans in addition to car loans is an example of these.

6 Discussion and conclusion

The four factors introduced in this paper reflect the four perspectives that have been reported in IS literature, namely, the holistic view brought by the goal–actor–resource–process perspective; the information perspective; the functional, law-based perspective; and the dynamic perspective. These factors attempt to go across all the constructs of the SW ontology and establish a correspondence with those of the WW ontology. The criteria and metrics developed to support these factors adopt fit measurement that takes the business view as a reference and compare the system view with it. Therefore in any metric, the denominator always refers to constructs of the SW ontology.

The use of the WW and the SW ontologies allows to formally define a generic fit measurement system. Indeed, on the one hand, the metrics are not only expressed in natural languages, but rely on formal concepts and links that are clearly and precisely defined. On the other hand, the ontologies are not used as model but as meta-model to represent the system. The fit measurement system is thus defined at a generic level. The paper briefly illustrated the use of these metrics. It was shown (1) how to customise the metrics to given models by linking the ontologies with two specific models and adapting the metrics accordingly and (2) how to use them to discuss the level of fit in a concrete information system application. The reported experiment showed that measures can help relating problems which are felt in the daily business life to lack of system support and then, making decisions to overcome these problems.

The measures rely on models and thus depend on their pertinence and how we can trust them. They allow to identify misfit problems between models and not directly between the business and the system. We are conscious that such models do not ever exist in companies; however, it seems that the use of models is spreading in the management world. The fit or unfit in our approach is measured on models and therefore, their effectiveness is subject to the fact that the models represent accurately the reality of the business and the system. Process mining and delta analysis [25] can be used to define models corresponding to what is really used. In the brief case study, we propose, first of all, to bring up to date the models in order they correspond as precisely as possible to the system or the business.

The metrics allow to put figures on an abstract concept, the alignment between business and system. They thus help to identify what is aligned and not and in which ratio. It is extremely rare that every measure equals one. This does not obviously imply that corrective actions are necessary. Indeed, thresholds introduction [13] can be pertinent and allow a better appreciation of the fit relationship. Different values can be affected to the threshold but correspond to the stakeholders' vision of the alignment. Thresholding helps the stakeholders to establish priority between the different measures. Thus, if the metric determines a value lower than the associated threshold, the business process and supporting software system are misfiting, motivating corrective action. This action can be a modification of either the business model or the system (even if concretely, it rather corresponds to modification of the system except when this latter represents "good practices" as ERP). Thresholding leads to a beneficial cycle of design-measurement-design for better fit. In our case study, the stakeholders could have considered that the system fits enough the business processes. Thresholding allows to put alignment into perspective and not to restrict fitness to cases where each fitness measure equals 1.

The measures described in this article give an image of the alignment at a given point of time. They help in the choice of design option, or in determining when corrective actions are necessary. However, they should not be the unique decision criterion of the stakeholders. Indeed, fit has not only to be reached but also and above all to be maintained. Thus, it is clear that a system perfectly supporting a business but not being flexible at all would perhaps not have the preference of the stakeholders on a system more flexible but less aligned with the business.

In order to better take into account the characteristics of each project in the calculation of the fitness measures, we intend to explore the weighting technique found in Software Quality. This weighting should be based on the relative importance accorded in the organisational context, to the different constructs of the SW ontology. For example, take the weighting of activity in the computation for the *support ratio*. Its definition considers the number of automated activities and ignores the relative value addition of activities in the business. Thus, it can happen that the support ratio is high but the most value adding activities are not automated. Appropriate weighting obviates this problem. Weights should be determined based on the experience of the stakeholders.

The metrics defined in this paper help (1) to better understand the fit concept, (2) to determine in what extend, in a given project, the business and the system are aligned (3) to identify corrective actions to undertake, (4) to reduce the lost of performance of the organisation resulting of a misfit. They constitute a help decision criterion.

Our research agenda relies on three crucial points, which will be the subject of further investigations: (1) more empirical assessments of these criteria; (2) the development of a tool to support design based on fit measurement and (3) improvement of the metrics to take into account partial support.

Acknowledgments Sincere thanks are to the anonymous reviewers who provided constructive criticism that has made improvements possible.

7 Appendix

Criterion	Metrics	Data
Support ratio (Sr)	Number of activies represented by system events/Number of activies Sr = card(A_b^2)/card(a_b)	- A_b be the set business activies - E_r be the set of system events - A_b^r be the set of business activities represented by a system event; $A_b^r = \{a, a \in A_b \| \exists e \in E_r \wedge e\Re a\}$
Goal satisfaction (Gs)	Number of goals for which each state maps a state in the system/ Number of business goals Gs = card $(G)_b{}^m$/card (G_b)	- G_b be the set of business goals - S_b be the set of business states - S_r be the set of states of the system $G_b{}^m$ be the set of business goals supported by the system $G_b^m = \{g \in G_b \| \forall s, s \in S_b \wedge s \in g \Rightarrow \exists s' \in S_s \wedge sMs'\}$
Actor presence (Ap)	Number of business actors mapping asystem class/Number of business actors Ap = card($Ac_b{}^m$)/card(Ac_b)	- $Ac{-}b$ be the set of business actors - C_s be the of system classes - $Ac_b{}^m$, be the set of business actors mapping a system class that triggers a stste transition on another system class. $Ac_b^m = \{a, a \in A\infty \| \exists c, c' \in C_s \wedge s \in c \wedge s' \in c' \wedge < s, s' > \wedge aMc\}$
Resource presence (Rp)	Number of business resource maping asystem class/Number of business resources Rp = card (R_b^m)/card (R_b)	- R_b be the set of business resources - L_s be the set of system laws - R_b^m, be the set of business resource mapping a system class for which there does not exist a stste change. $R_b^m = r, r \in R_b \| \exists c \in C_s \wedge \forall s \in S_s \wedge s \in c \wedge \forall l \in L_s \wedge l(s) = s \wedge rMc$
Informational completeness	Number of business class mapping system classes/Number of business classes Ic = card ($T_b{}^m$)/card (T_b)	- C_b is the set of business classes - $C_b{}^m$ is the set of business classes mapping a system class $C_b^m\{c, c \in C_b \| \exists c' \in C_s \wedge cMc'\}$
Informational accuracy (Ia)	Number of business states mapping system states/Number of business states card ($S_b{}^m$)/card (S_b)	- $S_b{}^m$ be the set of business states mapping a system states: $S_b^m = \{s, s \in S_b \| \exists s' \in S_s \wedge sMs'\}$
Activity completeness (Ac)	Number of business classes mapping classes/Number of business classes Ac = card (T_a^m)/ card(T_a)	- C_a be the set of business classes involved in a business activity a and - C_a^m the set of business classes involved in a business activity a and mapping a system class. $C_a^m = \{c, c \in C_a \| \exists c' \in C_s \wedge cMc'\}$
Activity Accuracy (Aa)	Number of business ststes mapping system states/Number of business states Aa = card ($S_a{}^m$)/ card (S_a)	- Sa be the set of business states involved in a business activity a, - $S_a{}^m$ be the set of business states involved in a business activity a and for which there is a maping with a system states. $S_a^m = \{s, s \in S_a \| \exists s' \in S_s \wedge sMs'\}$
System Reliability (Sre)	Number of business laws for which each business states maps a system states and the transformation between business states are possible between system states/Number of business laws Sre = card ($L_b{}^m$)/card (L_b)	- L_b be the set of business laws - l_r be the set of system laws - $L_b{}^m$ be the set of business laws mapping asystem laws $L_b^m = \{l, l \in L_b \| \exists l' \in L_s \wedge lMl'$
Dynamic realism (Dr)	Number of paths for which each business state map a system states and the succession of these states is possible/Number of possible paths Dr = card ($P_b{}^m$)/ card (P_b)	- P_b is the set of paths - $P_b{}^m$ the set of paths supported by the system

$$P_b^m = \{p, p \in P_b \wedge p = \wedge \forall i, \quad s_i \in S_b \wedge s_i \neq s_i \wedge \exists l \in L_s \wedge s_{k+1} = l(S_k) | \exists l' \in L_s \wedge lMl' \wedge \forall i, \exists s'_m \in S_s \wedge s_i M s'_m\}1$$

References

1. Davenport TH, Short JE (1990) The new industrial engineering: information technology and business process redesign. Sloan Manag Rev 31(4):11–27
2. Giaglis GM (2001) A taxonomy of business process modelling and information systems modelling techniques. Int J Flexible Manufact Syst 13(2):209–228
3. Arsanjani A, Alpigini J (2001) Using grammar-oriented object design to seamlessly map business models to component-based software architectures, proceedings of the international symposium of modelling and simulation, Pittsburgh, pp 186–191
4. Salinesi C, Rolland C (2003) Fitting business models to systems functionality exploring the fitness relationship. In: Proceedings of CAiSE'03, Velden, Austria
5. Fifth workshop on business process modeling, development, and support BPMDS'04, Riga, Latvia, 2004
6. Eatock J, Giaglis GM, Paul RJ, Serrano A (2000) The implications of information technology infrastructure capabilities for business process change success. In: Henderson P (ed) Systems engineering for business process change. Springer-Verlag, London, pp 127–137
7. Regev G, Wegmann A (2004) Remaining fit: on the creation and maintenance of fit. In: Proceedings of BPMDS'04, Riga, Latvia
8. Nadler D, Tushman ML (1980) A congruence model for diagnosing organizational behavior. In: Miles R (ed) Resource book in macro organizational behavior, Goodyear, Santa Clara, pp 30–49
9. Soffer P, Wand Y (2004) Goal-driven analysis of process model validity. In: Proceedings of CAiSE'04, Riga, Latvia
10. Wand Y, Weber R (1992) An ontological model of an information system. IEEE Trans Software Eng pp 1282–1292
11. Bunge M (1977) Treatise on basic philosophy: ontology I. The furniture of the world, Reidel

12. Bunge M (1979) Treatise on basic philosophy: ontology II. A world of systems, Reidel
13. Bodhuin T, Esposito R, Pacelli C, Tortorella M (2004) Impact analysis for supporting the co-evolution of business processes and supporting software systems. In: Proceedings of BPMDS'04, Riga, Latvia
14. Boehm B, Brown JR, Kaspar H, Lipow M, McLeod G, Merritt M (1978) Characteristics of software quality. North Holland
15. Cavano JP, McCall JA (1978) A framework for the measurement of software quality. In: Proceedings of the software quality and assurance workshop, San Diego, CA, ACM SIGMETRICS and SIG-SOFT, pp 133–139
16. Wand Y, Weber R (1989) An ontological evaluation of systems analysis and design methods. In: Falkenberg ED, Lindgreen P (eds) Information system concepts: an in-depth analysis. Elsevier, Amsterdam, pp 79–107
17. Rosemann M, Green P (2002) Developing a meta model for the Bunge-Wand-Weber ontological constructs. Inf Syst J 27:75–91
18. Wand Y, Weber R (1993) On the ontological expressiveness of information systems analysis and design grammars. J Inf Syst 3(4):217–237
19. Wand Y, Weber R (1995) On the deep structure of information systems. Inf Syst J 5:203–223
20. Ralyté J, Rolland C, Deneckère R (2004) Towards a meta-tool for change-centric method engineering: a typology of generic operators. In: Proceedings of CAISE'04, Riga, Latvia
21. Pohl K (1996) Process-centered requirements engineering. Wiley, New York
22. ISO/IEC 9126: Software Engineering—Product quality—Part 1: Quality model (2001)
23. Rolland C, Prakash N (2001) Matching ERP system functionality to customer requirements. In: Proceedings of the 5th international symposium on requirements engineering (RE'01), Toronto, Canada, pp 66–75
24. Prat N (1997) Goal formalisation and classification for requirements engineering. In: Proceedings of the third international workshop on requirements engineering: foundations of software quality REFSQ'97, Barcelona, Spain, pp 145–156
25. van der Aalst WMP (2004) Business alignment: using process mining as a tool for delta analysis. In: Proceedings of BPMDS'04, Riga, Latvia