https://doi.org/10.1007/978-3-642-01862-6_7

Vertical Alignment of Process Models - How can we get there?

Matthias Weidlich[1], Alistair Barros[2], Jan Mendling[3], and Mathias Weske[1]

[1] Hasso Plattner Institute, Potsdam, Germany
{matthias.weidlich,weske}@hpi.uni-potsdam.de
[2] SAP Research, CEC Brisbane, Australia
alistair.barros@sap.com
[3] Humboldt-Universität zu Berlin, Germany
jan.mendling@wiwi.hu-berlin.de

Abstract. There is a wide variety of drivers for business process modelling initiatives, reaching from business evolution and process optimisation over compliance checking and process certification to process enactment. That, in turn, results in models that differ in content due to serving different purposes. In particular, processes are modelled on different abstraction levels and assume different perspectives. Vertical alignment of process models aims at handling these deviations. While the advantages of such an alignment for inter-model analysis and change propagation are out of question, a number of challenges has still to be addressed. In this paper, we discuss three main challenges for vertical alignment in detail. Against this background, the potential application of techniques from the field of process integration is critically assessed. Based thereon, we identify specific research questions that guide the design of a framework for model alignment.

Key words: process model alignment, business-IT gap, model consistency, model correspondences

1 Introduction

The broad field of application of Business Process Management (BPM), from process analysis to process enactment, results in a variety of requirements for BPM methods and techniques. In particular, there is a huge difference in the appropriate level of abstraction of processes, as well as the assumed perspective. Both, abstraction level and perspective, depend on the purpose of the model and the involved stakeholders.

Evidently, real-world scenarios require multiple process models, each of them created for a specific objective. Such a model has to be *appropriate* in the sense that it incorporates a reasonable level of detail, focus on certain properties, and neglects unrelated aspects. As diverging modelling purposes cannot be organized in a strict top-down fashion, it is unrealistic that the corresponding models can always be derived through hierarchical refinement. Consequently, and most likely, there will be a variety of differences between models. Arguably, these *mismatches* are in the nature of process models that serve different purposes. Thus, avoidance of mismatches might not only be impossible in certain scenarios, it might also be unnatural and counter-productive. That is to say that a resolution of these mismatches might impact the adequacy of a process model in a negative manner.

A widely known example for the problem of aligning high-level and low-level models is the missing fit between business process models and workflow models. For more than a decade, this notorious 'Business-IT Gap' has motivated various researchers to investigate a better alignment of such models [1, 2, 3, 4, 5]. The prominence of this mismatch has somewhat hindered the discussion of the problem in a more general setting. Due to a similar difference in purpose, we observe that process models that are created to reflect control objectives for Sarbanes-Oxley compliance can hardly be used for process reengineering. In the same vein, SIPOC process diagrams are hardly informative to workflow implementation projects. While process modelling builds on a certain core in terms of task description, the diverging application scenarios for these models (see [6]) result in models that cover accounting operations, web service invocations, control activities, or strategic to-dos.

This paper argues that various aspects of an alignment of process models have not yet been investigated in a sufficient manner. Results from various research fields, for instance process integration and behaviour inheritance, might be adapted for alignment purposes. However, the scope of model alignment goes beyond the requirements that have typically to be satisfied in these research fields. Therefore, this paper elaborates on the challenges for vertical model alignment in detail and outlines the steps to be taken in order to achieve a mature solution. Albeit complicated by the usage of different modelling approaches (with potentially varying expressiveness), the problem of vertical model alignment is independent of any language. For illustration purposes, we use the Business Process Modeling Notation (BPMN) [7] throughout this paper. In order to clarify our point, we explicitly exclude mismatches from the discussion that stem from a mismatch between different modelling languages (such as BPMN and BPEL).

Against this background, our contribution is twofold. First, we motivate the need for vertical alignment and elaborate on three major challenges in detail. Second, we discuss why existing techniques are not sufficient in order to address these challenges and identify open research questions. The remainder of this paper is structured accordingly. The next section introduces a motivating example along with the major use cases for an alignment. Subsequently, Section 3 reviews related work. In Section 4 we elaborate on the major challenges for an alignment of process models. Based thereon, a set of research questions that need to be tackled is presented in Section 5. Finally, Section 6 concludes the paper.

2 Motivating Example and Use Cases

In order to illustrate the need for vertical alignment of process models, Figure 1 depicts two process models describing a lead management process, which we encountered in the course of an industry corporation. The upper model shows

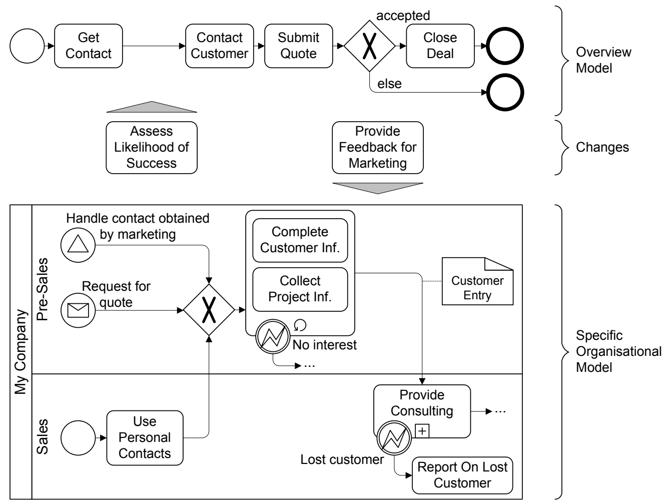

Fig. 1. A lead management scenario, described by two models that need to be aligned

solely the major activities, from getting a customer's contact details to arranging a deal with them. Here, an intuitive overview of the major processing steps, independent of any concrete organisational or technical environment, is in the centre of interest.

At the other end of the line, processes are specified in a fine-grained manner. They might aim at capturing technical aspects, such as the treatment of exceptional cases or data mediation. Furthermore, low-level models often also focus on the relation between the process and its execution environment. Organisational units that are mandated to execute the tasks and information systems that support their execution are assigned to certain parts of the process. The lower process in Figure 1 is an example for such a low-level model. It provides not only a more fine-grained view, but also relates activities to organisational roles.

Granted that there are multiple process models as described before, vertical alignment of process models is mainly driven by three use cases.

Validation. In various situations as, for instance, related to the 'Business-IT Gap' one process model is utilized as a specification against which a second, often more fine-grained model is validated. However, validation is not restricted to technical models. The upper model in Figure 1 might also be interpreted as a specification for the implementation of the process in a certain organisational environment, that is, the lower model.

Inter-Model Analysis. Process optimisation often requires an analysis across multiple process models. With respect to the exemplary processes in Figure 1, one might want to identify all roles that are involved when a customer is contacted. Starting from the activity *Contact Customer* of the high-level model, this information depends on one or more low-level models.

Change Propagation. Once potential improvements have been identified, all related models have to be updated accordingly. This can imply that changes in one model have to be propagated to the other models, and vice versa. While automatic change propagation appears to be unrealistic, the identification of affected processes or process regions, respectively, would already be a major benefit. Changes in process models can origin from all abstraction levels. Strategic management decisions will typically be reflected as changes in high-level models, whereas the replacement of a technical system enforces an adaptation of a low-level model. Consequently, change propagation has to happen top-down as well as bottom-up. Figure 1 illustrates both cases.

Addressing these use cases, any alignment has to embrace means for correlating elements of different models. These correspondences, in turn, have to respect certain consistency criteria in order to be exploited for model validation, analysis, or change propagation.

3 Related Work

Our work relates to the various research areas, namely *integrated system design*, *process integration*, *measures for process similarity*, and *behaviour inheritance*.

Integrated system design relates to various approaches that have been proposed to derive technical realisations from business requirements. In this case, consistency is achieved by deriving *information system models* directly from *business models*. In [1], the author raises the awareness for interdependencies between such models and introduces the notion of *vertical integration*, which comprises refinements for data objects and their relationships, as well as activities and their life-cycles. Considering also transactions, *realisation types* [4] that transform a business model into a technical model are another approach to derive technical models from business requirements. Bergholtz et al. [8] advocate the usage of communication patterns that guide the creation of process models from business requirements. This work has later been extended towards a framework, in which process models are derived from business models via activity dependency models as an intermediate step [9]. Due to the focus on the system development from-scratch, the aforementioned approaches are limited to rather strict refinements and do not deal with detection or resolution of inconsistencies. Taking existing informations systems into account, business-driven development (BDD) [5] aims at seamless transition from business-centred *analysis models* to technology-centred *design models*. Here, the authors describe transformation steps concerning the control flow, data representation, and service landscape in order to realise this transition. Other authors introduced a *process support layer* [10] realising common mismatch patterns to bridge the gap between process models and existing service landscapes. These patterns focus on differences related to service granularity, ordering, and interaction behaviour. Still, these approaches assume comprehensive derivation of technical models from business models, which implies a rather tight-coupling of these models.

Process integration assumes that process models originate from different sources and, therefore, are different yet similar. Common integration approaches for process models aim at unification of multiple views on a process, process harmonisation after an organisational merger, or the evolution of existing processes using reference models. Various publications define a merge operation for behavioural models based on model correspondences [11, 12, 13]. Nevertheless, this operation typically considers solely the control flow dependencies. A systematic classification of differences between similar processes has recently be published by Dijkman [14]. This work describes mismatches related to the control flow, resource assignments, and activity correspondences between two models that should be integrated. For control flow mismatches, a detection technique has also been presented [15]. Although process integration methods show how certain mismatches can be detected and resolved, they typically focus on *very similar* processes on the same level of abstraction. Thus, these models differ only slightly. The same delimitation holds for existing approaches to integrate different behavioural views, for instance [16], in which enterprise and computational views are aligned under the assumption of hierarchical refinement.

Measures for process similarity are related to our work, as vertical alignment assumes models to be similar to a certain extent. The authors of [17] present such a measure based on the enforced execution constraints. Moreover, a similarity measure might also be grounded on change operations [18]. Aiming at querying of models that are similar regarding their structure but reside on different levels of abstraction, Soffer introduced *structural equivalence* [19]. Still, focus is on hierarchical refinements between these models.

Behaviour inheritance aims at applying the idea of inheritance known from static structures to behavioural descriptions. In [20], Basten et al. introduced different basic notions of behaviour inheritance, namely *protocol inheritance* and *projection inheritance* based on labelled transition systems and branching bisimulation. A model either inherits the behaviour of a parent model, if it shows the same external behaviour when all actions that are not part of the parent model are *blocked* (protocol inheritance) or *hidden* (projection inheritance). Similar ideas have been presented in [21], in which the authors distinguish *invocation consistency* and *observation consistency*. These notions correspond to the notions of Basten et al. mentioned above [20]. Focussing on object life cycles, Schrefl and Stumptner built upon this work and argued that there is no exclusive choice between invocation consistency and observation consistency [22]. They also further distinguished *weak invocation consistency* and *strong invocation consistency*. The former implies inheritance of the interface, while the latter also enforces that added activities do not interfere with the inherited interface.

4 Challenges for Vertical Alignment

In this section, we discuss what we see as the major challenges for vertical alignment of process models. Section 4.1 first identifies the spectrum of differences before Section 4.2 discusses challenges of defining model correspondences. Finally, Section 4.3 describes requirements for measuring a degree of consistency.

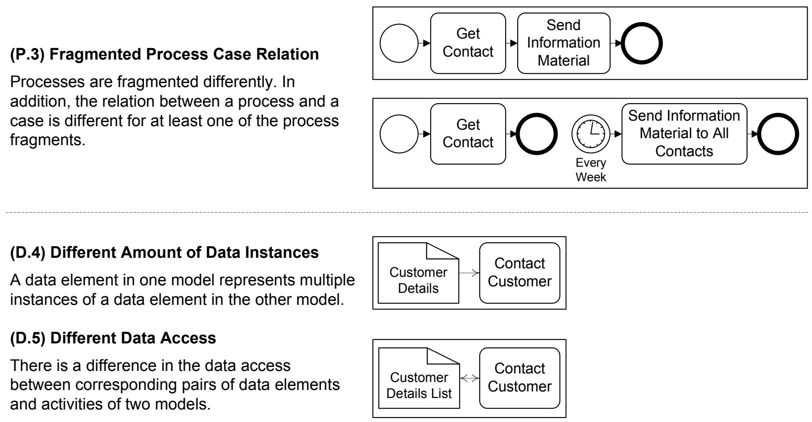

Fig. 2. Examples for differences in process slicing, data access, and instance correlation

4.1 A Variety of Differences

Process models describing a scenario on different abstraction levels and from different perspectives, naturally show various kinds of differences. As mentioned above, there is related work on differences between quite similar processes. Nevertheless, existing classifications focus on resource assignments, activities, or the control flow, and neglect the process, and data perspective.

For instance, the upper part of Figure 2 illustrates differences related to the process perspective. Here, the slicing of processes is different as a process in one model is split up into two processes in the other model. In addition, we encounter differences with respect to instance correlation. In contrast to the most upper process, sending of information material is not an atomic activity in the process below. Here, sending is done via batch processing. As these mismatches cannot be traced back to elements of the process model, but refer to sets of process models, they are said to relate to the process perspective. The same kind of instance correlation issue can also arise with activities or data objects, illustrated in the lower part of Figure 2. Moreover, this example shows differing data access. While the first activity has only read-access, its counterpart might modify the respective data object. Due to space limitations, we have to restrict the discussion to these exemplary differences in this paper and refer to a technical report for an informal description of more differences relevant for vertical alignment [23]. An assessment of existing classifications of differences against our set of differences is shown in Table 1. This reveals only partial support for the differences that we identified and, therefore, motivates further investigation. The reason for the limited support is a predominant focus on comparison of rather similar processes. As these processes typically reside on the same level of abstraction, some of our differences are of minor importance for the purpose of process integration.

The variety of differences illustrated in table 1 raises the question of how they can be classified and formalized in a systematic manner. The most extensive

Table 1. Differences of process models (informal descriptions can be found in [23]) and how they are considered in existing classifications

	ID	Mismatch	Henkel [4]	Decker [10]	Dijkman [14]
Proc.	P.1	Process Fragmentation	–	–	–
	P.2	Process Case Relation	–	–	–
	P.3	Fragmented Process Case Relation	–	–	–
Activity	A.1	Activity Fragmentation	+	+	+
	A.2	Partial Activity Equivalence	–	–	+
	A.3	Non-Covered Activity	–	–	+
	A.4	Activity Iteration	–	–	+
	A.5	Activity-Case Relation	–	+	–
Flow	C.1	Different Causal Dependencies	+	+	+
	C.2	Rerouting	+	–	+
	C.3	Alternative Merge	–	–	+
	C.4	Decision Distribution	–	–	–
Data	D.1	Data Element Fragmentation	+	–	–
	D.2	Partial Data Element Equivalence	–	–	–
	D.3	Non-Covered Data Elements	+	–	–
	D.4	Different Amount of Data Instances	–	–	–
	D.5	Different Data Access	–	–	–
Resource	R.1	Resources Fragmentation	+	–	+
	R.2	Partial Resources Equivalence	–	–	+
	R.3	Non-Covered Resources	–	–	–
	R.4	Contradicting Resource Assignments	–	–	+
	R.5	Additional Resource Assignments	–	–	+

collection of differences, published by Dijkman [14], is based on the notion of *black-box equivalence* and *white-box equivalence*. The first requires the effects of two related units of work to be the same, whereas the second criterion also requires the way these effects are achieved to be the same. Although it is mentioned that equivalence is defined between sets of activities, phenomena that result from different abstraction levels are not further investigated. However, in our context, we have to consider these effects. Therefore, we advocate to extend the classification of differences from two dimensions, i.e. *what* is specified and *how* it is achieved, with a third one, which takes the level of detail into account. Thus, differences can be clustered according to one of the following aspects, *model coverage*, *behavioural contradictions*, and *information density*.

Model coverage relates to the question, whether there is a difference in what is described in two models. That is, the process models are examined regarding the coverage of functionality and descriptions of data and resources. In other words, to which extent is the scenario described in one model reflected in the other model? An example is given in Figure 3. Compared to process (A), the process (B) contains an additional activity, i.e. *Notify Candidate*. Differences in model coverage can be coarse-grained (whole process parts of one model are without counterpart in the other model), as well as fine-grained (activities or data elements without counterpart).

Behavioural contradictions relates to the question of how certain behaviour is achieved. Even in case the same functional part of a business scenario is captured by two models (no difference in model coverage), the realisation of this part might be different. For instance, there are differences in the execution order

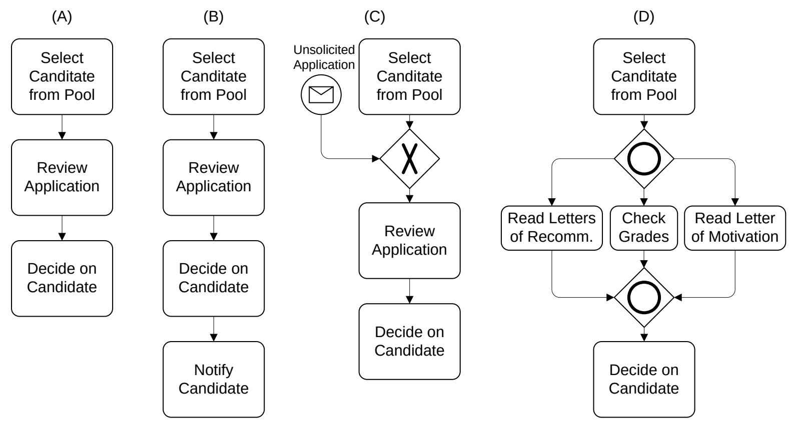

Fig. 3. A base process (A) and three process variants that differ with respect to model coverage (B), behavioural contradictions (C), and information density (D)

of corresponding activities, there is differing data access between corresponding activities, or a resource assignment in one model contradicts the one in another model. Again, Figure 3 illustrates such a difference with process (C) that specifies another entry point compared to process (A).

Information density relates to the question of how detailed the process is described. Two process parts realising the same scenario (no difference in model coverage) in the same way (no difference with respect to behavioural contradictions) might be specified in a different level of detail. Here, a typical example would be the refinement of an activity, as illustrated with process (D) in Figure 3, again compared to process (A). Different non-conflicting resource assignments of corresponding activities are another example for such a difference.

We summarize that vertical alignment has to deal with a broader variety of model differences compared to the existing work regarding process integration. Here, it is interesting to notice, that certain differences between processes that have been observed in practise, for instance in terms of enterprise integration patterns [24], have not yet been considered in the detection of differences to the best of our knowledge. Thus, the challenge is a comprehensive classification and formalisation of model differences. Such a formalisation might be inspired by the notions of *refinement* and *extension* as introduced for object life cycles [22].

4.2 Model Correspondences

A substantial requirement for vertical alignment of process models are means to correlate model elements. These *correspondence links* associate one or more elements of one model, with its corresponding elements of a second model. Any validation and inter-model analysis, as well as change propagation depends on these connections. Concerning model correspondences, we can identify two major challenges. First, the question how correspondences are established has to be addressed. Second, semantics of correspondences have to be defined.

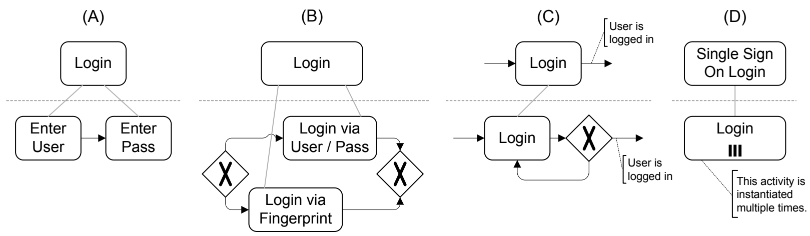

Fig. 4. Examples for model correspondences with different semantics

The question of the origin of model correspondences is crucial for the applicability of vertical model alignment. For real world scenarios, correspondences cannot be defined manually, owing to the pure number of models and model elements. Therefore, techniques that allow for automatic or at least semi-automatic definition of correspondences need to be exploited. Linguistic analysis of element naming, domain specific ontologies, or analysis of data dependencies are just a few examples of techniques that might be applied. It might also be necessary to select a set of related models from a repository prior to determining correspondences between them. That, in turn, results in additional efforts.

Besides their implications on techniques for finding correspondences, the aforementioned differences raise the question of semantics of correspondences. In other words, what is the meaning, if two (sets of) model elements *correspond* to each other. Figure 4 illustrates this challenge by four exemplary process pairs. We see that a 1-to-n correspondence might be interpreted such that the conjunction of n model elements corresponds to the single model element (A). On the other hand, it might be interpreted in way that the correspondences are mutually exclusive (B). Theoretically, it might even be the case that m-out-of-n model elements together correspond to an element in the other model. Thus, the latter element corresponds to more than one (i.e. different to case (B)), but not all (i.e. different to case (A)) of the model elements connected via correspondences. Sure enough, the same questions regarding correspondence semantics arise for fragment-to-fragment correspondence. It might be the case that for two elements of one model, the sets of corresponding elements of the other model are overlapping. Moreover, semantics of a correspondence might be that one activity instance in one model corresponds to all instances of the respective activity in the other model (C). While this scenario assumes sequential iteration of a corresponding activity, a correspondence might have also been defined between activities with a different notion of a case (D). Here, one activity is instantiated for a set of logins, whereas the other is instantiated multiple times, for each login. Thus, semantics of the correspondence are that one activity corresponds to multiple concurrent instantiations of the other activity.

Semantics for correspondences were proposed in the context of process integration. However, the semantic relationships observed during process integration, for instance counterpart-related processes [25], do typically not appear between processes that should be aligned. Correspondences, as whose by Dijkman that we discussed above [15], might be seen as a starting point, but are still not able to capture the examples of Figure 4. Please note that although this figure illustrates the ambiguity of link semantics only for activities, similar problems arise for other kind of process elements, e.g. data objects or resources.

4.3 The Notion of Consistency

Meaningful analysis across multiple process models has to be related to a certain degree of consistency between these models. However, there is no commonly agreed on definition of consistency for models on different abstraction levels that also assume different perspectives. Above, we discussed that differences between process models can be clustered according to the aspect they relate to, i.e. model coverage, behavioural contradictions, and information density. It seems reasonable to assume that differences in information density do not affect consistency. In other words, consistency is independent of the level of detail in which a process is specified. Consequently, we assume models to be consistent, if they cover exactly the same part of a scenario and there are no behavioural contradictions between them. Starting with this informal definition, formalisation of the coverage criterion seems to be straight-forward. In contrast, a formalisation of the second criterion, the absence of behavioural contradictions, i.e. behavioural consistency, appears challenging.

In Section 3, we discussed related work from the field of behaviour inheritance. Inheritance notions typically focus on the so called *visible* behaviour, while internal behaviour is neglected. Thus, we have to clarify the notion of visible behaviour for the purpose of vertical model alignment. Considering only the interactions with partners of a process might not be sufficient, as an interchanged order of corresponding internal activities of two processes might not be detected. Nevertheless, such a contradiction affects consistency in a negative manner, as it hampers change propagation. Depending on the purpose of the alignment, there might be no invisible behaviour.

Despite that, behaviour inheritance notions are too restrictive and support only a limited variety of mismatches. The authors of the most liberal notion, namely *life-cycle inheritance*, list a set of *inheritance preserving transformation rules* [20]. The insertion of activities between existing ones or the addition of loops containing new actions are examples for these rules. Everything that goes beyond these rules, for instance differences in the process instantiation mechanism, does not preserve inheritance and is inconsistent. Thus, these notions assume that behaviour is *added* in a *structural way* (e.g. iteration, choice, sequential or parallel composition) in the course of refinement of process models. An assumption that does not hold for vertical alignment. Moreover, behavioural contradictions that relate to the data or resource perspective, for example differing data access and conflicting assignment of activities to resources, must be taken into account.

Even in case existing inheritance notions would be weakened to some extent, most of the real world alignment scenarios would probably be still inconsistent. Thus, a single Boolean answer to the question of consistency is not sufficient. Instead, consistency should either be assessed based on a set of distinguished criteria (similar to the different soundness criteria for the verification of control flow) or measured in a metric way. The former would be similar to the different soundness criteria for the verification of control flow or the *realisability levels* [26] that have been proposed for the alignment of business and technical models. Obviously, a pure metric (i.e. non-stepwise) consistency measure would have to be relative with respect to certain properties, e.g. size of models or the abstraction level. We consider the latter to be intuitive, as a big difference in the level of detail of two models might legitimate a certain degree of differences regarding model coverage or behavioural contradictions. Such a notion would ease change propagation, as the less invasive out of a set of change operations can be identified.

5 Empirical Research Questions

In the previous section, we outlined the major challenges for vertical model alignment. In order to address these challenges, this section identifies research questions that need to be answered through empirical research.

Specific Analysis Questions. In Section 2, we introduced three major use cases for model alignment. In case of change propagation, requirements for a model alignment framework are easy to derive. In contrast, the use case of inter-model analysis needs to be further refined. The *usefulness* of specific analysis questions has to be evaluated empirically. These analysis questions could be clustered according to the process perspective (e.g, activity or data perspective) or the difference categories (e.g., model coverage).

Synthesis of Model Correspondences. We mentioned before that it does not seem to be realistic to assume manual syntheses of model correspondences for real world scenarios. On the other hand, it also seems to be naive to assume that automatic techniques for deriving correspondences can approach the quality achieved by human-beings with specific domain knowledge. Therefore, the effort process modellers would be willing to invest needs to be analysed. In terms of the technology acceptance model [27], the potential *ease-of-use* of a framework for vertical model alignment needs to be investigated. There might be a trade-off between these results and the refined analysis use case; certain analysis questions might require a certain degree of manual alignment efforts.

Perception of Consistency. In order to shape a requirements framework for consistency notions applicable in the context of vertical model alignment, we need to know, which differences between processes affect consistency in a negative manner. First, our hypothesis on differences related to information density—we consider these differences to have no impact on consistency—has to be corroborated. In addition, the impact of the remaining differences on the *perceived consistency* of process models has to be further investigated. It seems reasonable to assume that certain differences are more likely to be tolerated than others. In contrast to an interchanged order of activities, a sequentialisation of concurrent activities might not be seen as a behavioural contradiction. Empirical evidence on the perception of consistency is therefore needed to define gradual or even metric consistency notions.

6 Conclusion

The need for an alignment of business-centred and IT-centred process models has been identified over a decade ago. In this paper, we argued that this alignment problem has to be generalised to more than two abstraction levels and two perspectives. That results from different drivers for process modelling, which requires an alignment of models serving a variety of purposes. Based on three use cases, we elaborated on three major challenges for model alignment, that is the characteristics of mismatches, the semantic ambiguity of model correspondences, and the definition of a consistency notion. Our main contribution is the assessment of existing techniques from the field of process integration in order to address these challenges. It becomes evident that these techniques cannot be applied in a straight-forward manner. Instead, they have to be extended and adapted in order to cope with the requirements for vertical model alignment.

On the one hand, some of the identified white-spots can directly be addressed in future work. For instance, mismatches that are not covered by existing work have to be formalised and classified. Subsequently, techniques for identifying differing semantics of correspondences have to be investigated. On the other hand, for other open issues, it is uncertain how existing techniques should be extended or adapted. In this paper, we pointed out three research questions that have to be answered as a prerequisite for the definition of an alignment framework. Currently, we are addressing these questions empirically. As a result, we hope to clarify the requirements framework for reasonable vertical model alignment.

References

1. Ramackers, G.J.: Integrated Object Modelling. PhD thesis, Leiden University, Thesis Publishers Amsterdam (1994)
2. Grover, V., Fiedler, K., Teng, J.: Exploring the Success of Information Technology Enabled Businessprocess Reengineering. IEEE Transactions on Engineering Management 41 (3), 276–284 (1994)
3. Rolland, C., Prakash, N.: Bridging the Gap Between Organisational Needs and ERP Functionality. Requirements Engineering 5 (3), 180–193 (2000)
4. Henkel, M., Zdravkovic, J., Johannesson, P.: Service-based processes: Design for business and technology. In: Aiello, M., Aoyama, M., Curbera, F., Papazoglou, M.P. (eds.) ICSOC, pp. 21–29, ACM (2004)
5. Koehler, J., Hauser, R., Küster, J.M., Ryndina, K., Vanhatalo, J., Wahler, M.: The Role of Visual Modeling and Model Transformations in Business-driven Development. Electr. Notes Theor. Comput. Sci. 211, 5–15 (2008)
6. Rosemann, M.: Preparation of Process Modeling. In: Process Management: A Guide for the Design of Business Processes. pp. 41–78, Springer (2003)
7. OMG: Business Process Modeling Notation (BPMN) 1.1. (January 2008)
8. Bergholtz, M., Jayaweera, P., Johannesson, P., Wohed, P.: A Pattern and Dependency Based Approach to the Design of Process Models. In: ER. LNCS, vol. 3288, pp. 724–739, Springer (2004)

9. Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T., Johannesson, P.: A Declarative Foundation of Process Models. In: Pastor, O., e Cunha, J.F. (eds.) CAiSE. LNCS, vol. 3520, pp. 233–247, Springer (2005)
10. Decker, G.: Bridging the Gap between Business Processes and existing IT Functionality. In: Proceedings of the 1st International Workshop on Design of Service-Oriented Applications (WDSOA), pp. 17–24, Amsterdam, The Netherlands (2005)
11. Frank, H., Eder, J.: Towards an Automatic Integration of Statecharts. In: ER. LNCS, vol. 1728, pp. 430–444, Springer (1999)
12. Mendling, J., Simon, C.: Business Process Design by View Integration. [28] pp. 55–64
13. Küster, J.M., Koehler, J., Ryndina, K.: Improving Business Process Models with Reference Models in Business-Driven Development. [28] pp. 35–44
14. Dijkman, R.M.: A Classification of Differences between Similar Business Processes. In: EDOC, pp. 37–50, IEEE Computer Society (2007)
15. Dijkman, R.M.: Diagnosing differences between business process models. In: BPM. LNCS, vol. 5240, pp. 261–277, Springer (2008)
16. Dijkman, R.M., Quartel, D.A.C., Pires, L.F., van Sinderen, M.: A Rigorous Approach to Relate Enterprise and Computational Viewpoints. In: EDOC, pp. 187–200, IEEE Computer Society (2004)
17. van Dongen, B.F., Dijkman, R.M., Mendling, J.: Measuring Similarity between Business Process Models. In Bellahsene, Z., Léonard, M., eds.: CAiSE. LNCS, vol. 5074, pp. 450–464, Springer (2008)
18. Li, C., Reichert, M., Wombacher, A.: On Measuring Process Model Similarity based on High-level Change Operations. In: ER. LNCS, vol. 5231, pp. 248–264, Springer (2008)
19. Soffer, P.: Refinement equivalence in model-based reuse: Overcoming differences in abstraction level. J. Database Manag. 16 (3), 21–39 (2005)
20. Basten, T., van der Aalst, W.M.P.: Inheritance of Behavior. Journal of Logic and Algebraic Programming (JLAP) 47 (2), 47–145 (2001)
21. Ebert, J., Engels, G.: Observable or Invocable Behaviour - You Have to Choose. Technical Report 94-38, Department of Computer Science, Leiden University (December 1994)
22. Schrefl, M., Stumptner, M.: Behavior-consistent specialization of object life cycles. ACM Trans. Softw. Eng. Methodol. 11 (1), 92–148 (2002)
23. Weidlich, M., Decker, G., Weske, M., Barros, A.: Towards Vertical Alignment of Process Models - A Collection of Mismatches. Technical report, Hasso Plattner Institute (2008) `http://bpt.hpi.uni-potsdam.de/pub/Public/BptPublications/collection_of_mismatches.pdf`.
24. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley (2003)
25. Grossmann, G., Schrefl, M., Stumptner, M.: Classification of business process correspondences and associated integration operators. In: ER (Workshops). LNCS vol. 3289, pp. 653–666, Springer (2004)
26. Henkel, M., Zdravkovic, J.: Supporting development and evolution of service-based processes. In: ICEBE, pp. 647–656, IEEE CS (2005)
27. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly (Sept.), 319–339 (1989)
28. BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings. In: BPM Workshops. LNCS, vol. 4103, Springer (2006)