Enterprise Information Systems, 2013
Vol. 00, No. 00, 1–29, http://dx.doi.org/10.1080/17517575.2013.857792

Improving the process of process modelling by the use of domain process patterns

Agnes Koschmider[a]* and ~~H.~~A. Reijers[b]

AQ1

[a]*Institute for Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Karlsruhe, Germany;* [b]*Eindhoven University of Technology, Eindhoven, The Netherlands*

(*Received 21 December 2010; accepted 30 September 2013*)

The use of business process models has become prevalent in a wide area of enterprise applications. But while their popularity is expanding, concerns are growing with respect to their proper creation and maintenance. An obvious way to boost the efficiency of creating high-quality business process models would be to ~~re use~~ relevant parts of existing models. At this point, however, limited support exists to guide process modellers towards the usage of appropriate model content. In this paper, a set of content-oriented patterns is presented, which is extracted from a large set of process models from the order management and manufacturing production domains. The patterns are derived using a newly proposed set of algorithms, which are being discussed in this paper. The authors demonstrate how such *Domain Process Patterns*, in combination with information on their historic usage, can support process modellers in generating new models. To support the wider dissemination and development of Domain Process Patterns within and beyond the studied domains, an accompanying website has been set up.

Keywords: process model reuse; process design; patterns; process modelling support; guidance

1. Introduction

Conceptual models of business processes are gaining popularity because of their versatile support for a range of business objectives. Such *business process models* enable the common understanding of the kind of business activities underlying a business process, including their mutual relationships. This type of insight is crucial in those situations, for example, when people are trained for a new job, when one wants to demonstrate a company's compliance with a set of regulations or when an existing way of developing a particular product or service must be improved. Additionally, business process models – or *process models*, for short – can be useful in the configuration of information systems. Workflow Management systems, Enterprise Resource Planning systems and Customer Relationship Management systems are all examples of systems that are either implicitly or explicitly driven by a model to support, control and/or monitor a described business process.

Whether it is for organisational or technical purposes, the act of business process modelling is not at all a trivial exercise. This act is defined as the effort to capture one or more business processes in the form of a conceptual schema, which serves a particular

*Corresponding author. Email: agnes.koschmider@kit.edu organisational or technical purpose, and which adheres to a particular modelling notation (EPCs, BPMN, YAWL, etc.). Rosemann, for example, provides a sobering list of 23 pitfalls based on his observations of business process modelling projects (Rosemann 2006a, 2006b). From this list, one can sense the trouble that many modellers have to come up with a process model of sufficient quality, either in a syntactical or semantic sense, within a reasonable amount of time. Most notably, modellers 'get lost in detail', trying to capture all possible scenarios rather than the relevant ones, or fail to connect to business representatives who hold the domain knowledge that needs to be captured.

1.1. Motivation

If one considers that many organisations are nowadays in the possession of hundreds or even thousands of process models on the one hand (Weber and Reichert 2008; Reijers, Mans, and van der Toorn 2009) and that various vendors such as SAP and Oracle offer reference models for all kinds of domains, it seems worthwhile to investigate opportunities that foster *reuse* of process model content.

This paper presents a new approach that aims to facilitate and simplify the reuse of process model content. The line of thinking of the authors to focus on domain-centered reuse of process model content has been particularly motivated by the practical experience with the development of a process modelling support tool (Koschmider, Hornung, and Oberweis 2011; Koschmider, Song, and Riejers 2010). This tool suggests process models or groups of (logically) related process activities – called a process model part – that can be re used to build a process model. To populate the repository for the modelling support tool, a large number of process models have been fragmented into their constituent process model parts. ~~Wile~~ doing so, striking similarities between ~~these~~ on an abstract level were noticed. In other words, similar sets of process activities occurred, suggesting their generic value in a particular modelling domain.

1.2. Example

Consider, for instance, a semiconductor company that wishes to describe its purchasing process of silicon components. The traditional approach is the following (Sharp and McDermott 2008): A project team of analysts and business representatives is assembled, they interview a range of stakeholders (management, operational staff, administrative staff, etc.), disclose their findings in an initial model, verify and validate this model with stakeholders and potentially ~~re iterate~~ these steps until a model of sufficient quality is established. Depending on the circumstances, such a procedure may take weeks or even months. Presumably, however, the company itself (or one of its subsidiaries) has already captured purchasing processes in models before, which may cover common elements. Also, the company perhaps could gain access to third party process models of this particular procedure. In other words, ~~re using~~ process model content, in part or in full, could speed up the process of process modelling considerably.

1.3. Approach

In this work, these similarities will be substantiated by proposing a set of *Domain Process Patterns* (DPPs). A DPP captures a process model part with a particular structure, which represents a specific function of it in a modelling domain. To arrive at these DPPs, 190 process model parts from the order management and the manufacturing production domains were analysed. Both domains have been identified as highly structured business processes (Davenport 1993; Norman 1996), which facilitates the derivation of reasonably generic tasks. Due to the structuredness of these process models, algorithms were defined that support the fragmenting of business process models into meaningful model parts. For these two domains, a set of generic tasks were identified that subsumes a group of process activities. Additionally, the authors found out that some tasks are exclusively used in one of these two domains – an insight that is useful to support process modellers in actual modelling efforts.

1.4. Application

Domain Process Patterns raise various application scenarios. The main scenario is process reuse that is exemplified in Section 6.1. The idea is that, parts of process models are particularly valuable to be reused in other process models when they concern the *same functional domain*. Stated differently: To create a process model for a purchasing procedure, it is more valuable to look at process model content from other purchasing process models than, for example, customer support or manufacturing models. This also makes intuitive sense given, for instance, the success of the Information Technology Infrastructure Library (ITIL) which prescribes a set of recommendations only for a limited number of processes, such as incident and problem management. The same idea is also central to the development of MIT's process handbook, which allows companies to compare their processes with those of other parties in the same functional area (Malone, Crowne, and Herman 2003).

Another appropriate scenario is process model analysis. DPPs allow for analysing the flow of process activities with respect to their reasonableness from a content perspective. Such a semantic-based verification of process models is only investigated in a limited fashion.

Domain Process Patterns are also suitable as foundation to measure process model similarity or to evaluate the understandability of process models (see Section 6).

1.5. Contribution

The contribution of this paper can be seen on two levels: the tangible and more abstract. With respect to the former, the proposed set of DPPs can be considered as a valuable form of reusable process model content for process models that are to be created within the order management and the manufacturing production domains. The usage of DPPs is then beneficial in at least two respects. First of all, their usage does not depend on a particular tool or modelling language; DPPs are modelling-language independent and can be used as a basis for modelling support in any modelling environment. Second, DPPs facilitate the reuse of process model parts, which is a common technique in order to build up new process models by assembling already designed ones. This type of process reuse potentially reduces the business process modelling effort. It will be demonstrated how the use of DPPs can be exploited within a concrete modelling environment.

On a more abstract level, this contribution lies in the explanation of how DPPs can be derived, so that it can be followed up for other domains by using the procedure and algorithms as described. The authors also contribute to this more abstract line in setting up an open and extensible website, which can be used for the storage and dissemination of existing and new DPPs.[1] Instructions on how to use DPPs and how to contribute to their further development can be found on this web page.

The remainder of this paper is structured as follows. Section 4 presents DPPs in detail. Initially, a categorisation for DPPs will be sketched, Section 4.2 describes attributes characterising a DPP. Subsequently, an algorithm to detect DPPs will be described in Section 4.3. DPPs will be presented in detail in Section 4.4. Section 5 describes empirical findings that give evidence to the existence of DPPs. Section 6 illustrates an application scenario for DPPs. Section 3 compares the approach suggested in this paper with related work. The paper concludes with a summary and an outlook on further research work.

2. Difference to other process pattern types

The use of DPPs can be seen as distinctively different from available approaches. Most notably, DPPs differ from the so-called *workflow patterns* as reported in the literature. Figure 1 shows differences between workflow patterns and DPPs. For instance, Control Flow Patterns (Van Der Aalst et al. 2003) cope with the syntactical structure of process models and describe syntactical relationships between process activities (e.g., the *Sequence Pattern* is used to describe a series of consecutive tasks). However, Control Flow Patterns do not examine functions of process activities. For this, Activity Patterns (Thom, Reichert, and Iochpe 2009) have been suggested, which focuses on the business function of a process activity (e.g., process activities enclosing a XOR split imply a decision; process activities that are connected in one direction denote a unidirectional performance). Still, Activity Patterns do not focus on the content of process activities. DPP, in contrast to Activity Patterns, abstract from the business function of a workflow activity and consider the content of process activities in a modelling domain. And where Control Flow Patterns provide insight into the syntactical needs for process modelling languages, DPPs facilitate reuse from a content perspective.

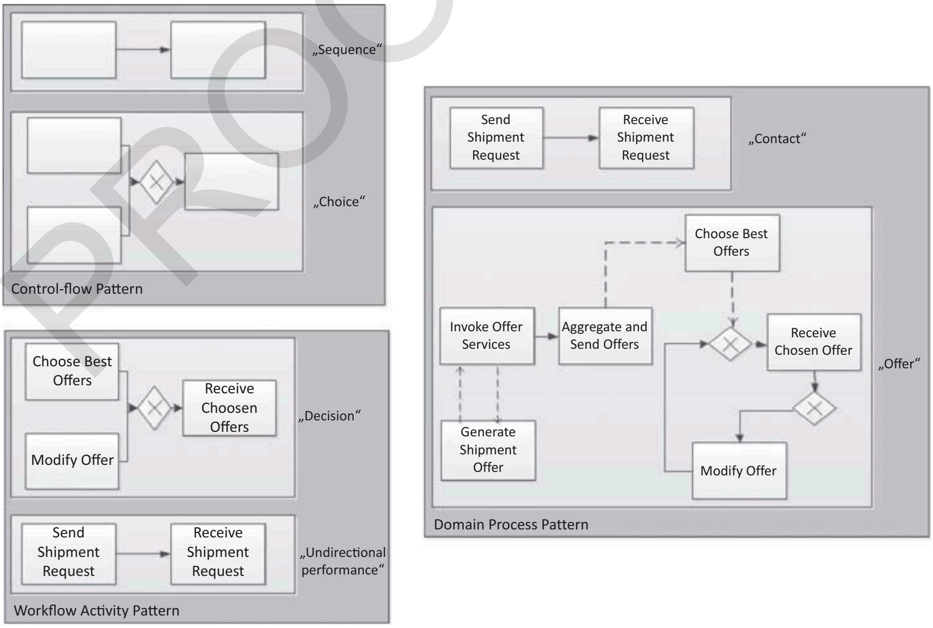

Figure 1. Relationship between several workflow pattern types and Domain Process Patterns.

Consider the following scenario. A person intends to create a purchase order process. He or she may either have some notion of the process or have no clue at all. The different process pattern types then may help the modeller in creating the process model by serving as potential, suitable process activities. Alternatively, the modeller may use the patterns for analysis purposes. In the first case (modelling support) the modeller can use the recommendations to complete the process model. In the second case (analysis), control-flow patterns are used to detect syntactical errors (e.g., the detection of a deadlock, which occurs because an XOR split is synchronised by an AND join). Activity Patterns support uncovering semantic errors. For instance, an XOR join is used to model a decision but the synchronised process activities do not imply any judgment. Domain Process Patterns are used *on top* of both pattern types. DPPs are applied to analyse the reasonableness of the combination of process activities with respect to descriptions of process activities. For instance, an error occurs in case where an order is released before the order is generated. It does not matter if process activities for order generation are modelled in sequence or choice. Yet, it is crucial that they proceed to release an order.

3. Related work

DPPs simplify the reuse of process model content, support the design of business processes and allow to ~~model validated business processes regarding its content.~~ Therefore, related works tackling these three issues are discussed and compared with DPPs.

Semantics-based analysis of process models. DPPs allow to validate a combination of chunks (model parts) regarding its content. Several related approaches exist tackling this analysis issue. For instance, an initial approach for semantic correctness of process models has been discussed by Fellmann et al. (2010a, 2010b). A weakness of this approach is that it depends on the availability of an ontology and a cumbersome annotation of process models. Some approaches exist that partially analyse the content (semantics) of process model parts. Van Der Aalst et al. (2003) suggests the Cancelation Pattern, that cancel instances in individual activities and Barros, Dumas, and ter Hofstede (2005) proposes the Send Pattern. Hao et al. (2008) defines a question answering-pattern that enables automatically responding to questions. In Thom, Reichert, and Iochpe (2009), a Notification Pattern can be found that defines the status or result of an activity execution that is communicated to one or more process participants. Fern (2000) defines analysis pattern for inventories that result from the abstraction of a real workable inventory. Fern, Yuan, and Brey (2000) also describes analysis patterns for the order and shipment of a product. All these approaches pertain to the semantics-based analysis of particular activities, while the novel concept of DPPs consider a holistics-based analysis of process models regarding its linguistics.

Reuse of Process Models. DPPs suit to facilitate process model reuse (see Section 6.1). A plethora of approaches is available allowing the access to the increasing amount of process models through reuse. Generally, reuse can be performed manually or semi-automatically (Lu, Sadiq, and Governatori 2009; Madhusudan, Zhao, and Marshall 2004; Markovic and Pereira 2007). Modellers can reuse the complete process model, process model parts (Koschmider, Hornung, and Oberweis 2011) or a customised version of the process model (Scheer 1999; Mendling and Simon 2006; Eshuis and Grefen 2008; Hornung, Koschmider, and Lausen 2008), which display different perspectives on a process model in order to improve its comprehensibility. Content-based reuse is not considered in any approach. Therefore, DPPs are a new beneficial concept. Although, DPPs address the linguistics level of communication, they complement techniques allowing reuse on the syntax level such as workflow patterns (see Section 2). Another advantage of DPPs can be seen in the extraction of patterns. To automatically extract process patterns (e.g., see Bögl, Kobler, and Schrefd 2008), a semantic annotation of patterns is presumed. With DPPs, the laborious semantic annotation can be omitted since the algorithm works on a set of process labels found and collected in the domain.

Modelling support for process design. Reuse is closely related to techniques supporting model design since process builders can be assisted in their model design through recommendation of chunks previously created by other users. Currently, only few approaches exist for modelling assistance. These approaches focus on support of small units such as labels, process elements, rules or fragments, or either infer prospective process activities based on process activities at hand in a workspace (Smirnov et al. 2010; Lincoln, Galani, and Gal 2010) or are based on user behaviour (Dorn et al. 2010). The design might also be improved with running systems containing operational data (Niedermann, Redeschütz, and Mitschang 2010). DPPs are suitable to extend these newly suggested modelling support techniques and to allow matured modelling support. DPPs abstract all these approaches since they address the highest level of the communication stack (semantics or pragmatics).

Another stream of research combines case-based reasoning (CBR) and business process modelling in order to leverage past process executions when specifying a new process model. In CBP, executed processes (also referred to as cases or templates) are stored in a knowledge base and are used to create new processes or workflows (Madhusudan, Zhao, and Marshall 2004). CBP and DPP address different ways of modelling support. While CBP is suitable for business process redesign (Mansar and Marir 2003) (CBP builts on a set of reference cases), DPPs act as a template to design validated process models without requiring any reference model.

To summarise, none of these related approaches investigate semantics-based (linguistics) relationships between process model parts for the design of process models.

4. Domain Process Patterns: algorithm and characteristics

In this section, the distilled set of DPPs is described in detail, as well as the procedure that was followed to identify these.

4.1. Classification of DPPs

Generally, process models can be fragmented based on the maturity of process activities to fulfill the model intention. For this, the fragmentation is based upon the tripartition of process models and their maturity, which defines three phases: preparation, processing and completion.[2] Given both domains, order management and manufacturing production, the *preparation* phase is regarded as the time from the initiation of a goods request by a customer (a person, a system, internal or external) until its delivery to processing. This phase covers a series of process activities being responsible for the initiation of a process model; for instance, handling the reception of goods (or inquiries, orders, products and services) by a customer (or a person, a system, internal or external).

The time taken for a checked goods request until the goods are ready to be delivered to fulfillment is called the *processing* phase (synonyms are execution, conducting, managing). The last phase, the completion phase (synonyms are termination, fulfillment),

Table 1. List of Domain Process Patterns classified on occurrence in the process model.

	Preparation	Processing	Completion
Both domains	Checking, Contact	Order	Inventory, Invoicing, Shipment
	Notification		
Order management	Offer		Payment
Manufacturing production		Production	Accounting
		Planning	

covers process activities to design the fulfillment (e.g., shipping, invoicing) of processed goods or requests and completing the processing of the goods request.

This tripartition of a process model appears on an abstract level (e.g., in a Process Map), and thus on a more general level a process model might only incorporate one phase. Section 6 shows how this tripartition allows a novel way of semantics-based analysis of process models.

Based upon the tripartition of process models process models were investigated and were assigned to the appropriate phase. This step was performed by using natural language processing techniques (Matsuo and Ishizuka 2003) and the tagging approach of the recommendation-based modelling support system presented in Koschmider, Hornung, and Oberweis 2011. Both approaches allow to extract keywords of a process model. Synonyms were detected using WordNet. The keywords of all process model parts of the sample were inspected and hypernymys were selected that subsume most of the keywords (e.g., shipment subsumes delivery of goods, transport of goods). For some keywords, the authors could not find an appropriate hypernymy (e.g., goal). Therefore, manual work is also needed to assign keywords or process activities to a DPP. Table 1 summarises the hypernymys, which will be presented in the following as DPPs.

In both investigated domains a series of process activities could be detected that described the checking of requests. These process activities are subsumed by the Checking DPP. A process can be initiated by a customer request. All process activities referring to this action are subsumed by the Contact DPP. For the *processing* phase, the Order DPP was identified that could be complemented by Inventory, Invoicing or Shipment process activities. In all the three phases of *preparation*, *processing* and *completion,* process activities to notify a role were identified. These activities are subsumed by the Notification DPP.

Groups of process activities that were unique for the order management domain are the Offer and the Payment DPP. Groups of process activities to be unique for the manufacturing production domain are Production, Accounting and Planning, where process activities for planning occurred in both the processing and the completion phase.

4.2. Attributes of Domain Process Patterns

Domain Process Patterns presented in this paper result from the investigation of business process models that were found in the repositories of IBM WebSphere, SAP WebFlow and of SAP Business Map. Additionally, the investigation was complemented with a set of process models created by members during several evaluations of a recommendation-based process modelling support system (Koschmider, Hornung, and Oberwies 2011). While most DPPs would seem generic enough to be applied to other application domains (e.g., Banking, Logistics), some appear exclusive to the two investigated domains (e.g., *Inventory Pattern, Production Pattern*).

A DPP is defined by the following seven attributes:

(1) **Pattern Name:** Describes the task being performed by the pattern.
(2) **Description:** Description of the pattern.
(3) **Purpose:** Description of the purpose of the pattern.
(4) **Problem:** Description of the problem that is addressed by the pattern.
(5) **Related Patterns:** List of patterns (in alphabetical order) that are predecessors and successors of the pattern and thus represent the frequent combination pairs with the specific pattern.
(6) **List of Labels:** List of labels (process activity names and arc inscriptions) occurring in a process model part and matching labels of DPPs.^{3}
(7) **Keywords:** List of tags that characterises the pattern. This attribute also includes the DPP name.

4.3. Algorithm to extract DPPs

This section describes the algorithm to extract DPPs. The algorithm works for a noun–verb labelling style of process activities (or event inscriptions). Inhomogeneous labelling styles can be unified using the refactoring approach of (Leopold, Smirnov, and Mendling 2010). Before using the algorithm, three preparation steps are required. In particular, stop words must be filtered, tokenising and stemming need to be applied. *Tokenising* separates process activity labels in nouns and verbs by using line breaks. *Stemming* reduces nouns and verbs to its undeclined form (inquiries → inquiry, checked → check). Note that the list of labels is not exhaustive and should be completed to fully automatically detect DPPs. Due to disambiguity reasons, the term *goods* is restricted with the five hypernyms *article, product, material, supplies* and *service*. For the term *agent*, the following five more concrete terms are subsumed: *customer, person, system, internal, external*. The detection algorithm will be explained based on Figure 2.

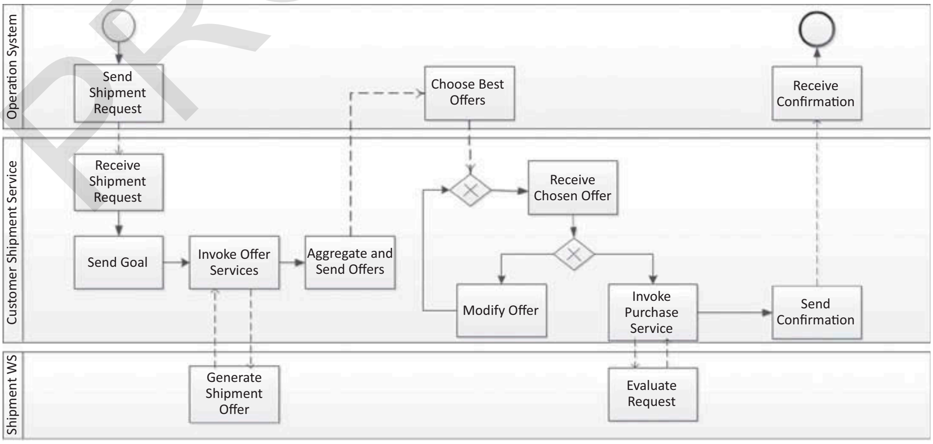

Figure 2. Example process model from the manufacturing domain.

Finally, the algorithm consists of the following three steps.

(1) **Consider activity labels and compare with the attributes List of Labels and Keywords**
In this first step, the labels of each process activity (and arc inscriptions) and the relative position of the process activity in the process model are considered. Note that an activity might also be labelled with more than one noun (e.g., specify shipment request) and each of these nouns might also be used with a verb (e.g., specify shipment, specify request). Then, each label is compared with the attributes *List of Labels* and *Keywords* and a match between them is searched for. In this first step the algorithm distinguishes three cases.

- Case 1: The activity is labelled by one noun and one verb.
- Case 2: The activity label has more than one noun and only one noun occurs in any *List of Labels* of a DPP or matches a keyword.
- Case 3: The activity label has more than one noun and at least two nouns can be found in the *List of Labels* or *Keywords* of different DPPs.

For Case 1 and Case 2, the activity is annotated with the name of the DPP with which a matching exists (matching between the label and the *List of Labels*[4]). For Case 3, each noun being part of a different DPP is initially annotated with the corresponding DPP. The position of process activities plays an important role for Case 3. Patterns can be superseded based on the position of process activities in the process model. Consider the process activity 'send shipment request' in Figure 1, which is annotated with the *Shipment* and the *Contact* DPP. Beginning at process activity four ('invoke offer services'), the algorithm detects the *Offer* Pattern. Since *Shipment* cannot be performed before making an offer; the Shipment DPP is in this case superseded by the Contact DPP.

(2) **Reassignment of process activities to patterns**
A series of process activities that are annotated with an identical DPP can have, in between them, a process activity that belongs to a different pattern as its presets and postsets. In such a case, the singular pattern is superseded by the dominated DPP.

(3) **Compose process activities to superior task**
Following the two preceding preparation steps, process activities are grouped together based on their annotations of DPPs. However, two cases are possible such that process activities cannot be assigned to any group yet.

- Case 1: The process activity is a dangling activity.
- Case 2: The label could be found neither in the *List of Labels* nor in the list of *Keywords*. The reasons for this are:
  - Case 2.1: The process activity (activities) is (are) within a dominated pattern that is applied to the preset and postset of this isolated activity.
  - Case 2.2. The process activity (activities) is (are) in between two different patterns.

For Case 1, the process activity is assigned to the group where the activity has an arc connection and the pattern of the adjacent group is imposed to this process activity (activities). For Case 2, the position of the process activity is important. For Case 2.1, the dominated pattern is imposed. Note the difference between this step and step 2 of the algorithm (reassignment of process activities). Activities assigned to Case 2.1 are process activities with an unknown label and may require manual work. For Case 2.2. the activity (activities) is (are) assigned to its preset. For instance, the activity ‘send goal’ in Figure 2 neither belongs to the Contact DPP nor to the Offer DPP and is assigned to the Contact DPP. In case of several of such in between process activities a manual decision must be made.

4.4. *Pattern description*

This section describes DPPs and a graphical example for each pattern ~~will be~~ shown ~~in BPMN~~. First, DPPs are presented that are found in both the domains investigated (C1-C7). Following are explained~~:~~ the DPPs that are unique for the order management domain (CO1-CO2) and finally the DPPs for the manufacturing production domain (CM1-CM3). Within each part, the patterns are ordered alphabetically.

4.4.1. Generic Patterns

	DPP C1: Checking
Description:	This pattern covers the time from the approval of a request to the fulfillment of the request. A checking process can be initiated after receiving a request by an agent. The request is transmitted using the *Contact Pattern*. The checking phase may include a validation of customer’s data or credibility, validation of the availability of goods requests or request for further information (e.g., price) of goods. The result of the checking phase is an approved, rejected, cancelled request or an update of data.
Purpose:	Represents a series of process activities modelling the approval process.
Problem:	The usage of this pattern implies that requests (or any input) must undergo a validation before they can be processed. This pattern also addresses the situation that a sensitive request is proceeded and security standards must be undertaken (e.g, check creditworthiness).
Related Patterns:	Cancellation, Contact, Inventory, Notification, Offer, Order, Payment, Production, Shipment, Update.
List of Labels:	**Verbs:** (approve|check|change|create|evaluate|inspect|validate), **Nouns:** (request|change|requirements|data|price|permission|availability|creditworthiness|customer|requirements|information|record|inquiry)
Keywords:	Approval, authorisation, checking, evaluation, inquiry, inspection, permission, validation.

Figure 3 shows some examples for the *Checking Pattern*.

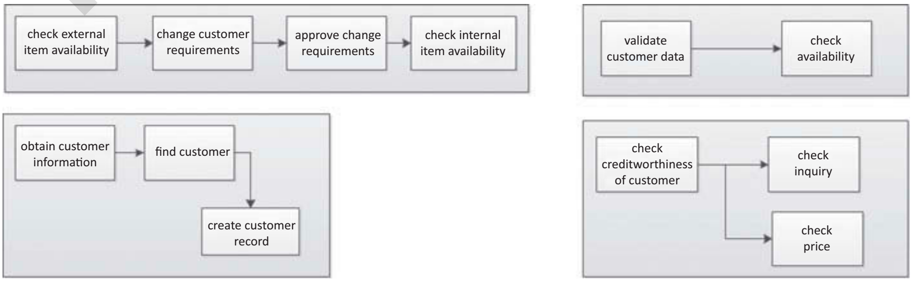
check external item availability
change customer requirements
approve change requirements
check internal item availability
validate customer data
check availability
obtain customer information
find customer
create customer record
check creditworthiness of customer
check inquiry
check price

Figure 3. Checking Pattern.

	DPP C2: Contact
Description:	This pattern is used to model activities for asking for further information or sending a request to an approver. The contact process may be initiated with an expressed need for information (e.g., quote, price) or an offer. The *Contact Pattern* represents a one-way interaction from the side of an agent with a supplier. The contact usually initiates the checking of the request. The answer to the request can be performed by the Notification Pattern.
Purpose:	Represents a series of process activities modelling the contact process.
Problem:	The usage of the Contact Pattern implies an external trigger (e.g., a customer request) of a process model. Additionally, the Contact Pattern indicates the need for clarification before processing of the input can be started.
Related Patterns:	Checking, Notification.
List of Labels:	**Verbs:** (contact|request|calculate|send|wiat|ask|receie), **Nouns:** (customer|price|receive|response|quote|request)
Keywords:	request, sending, asking, response.

Figure 4 shows some examples of the *Contact Pattern*.

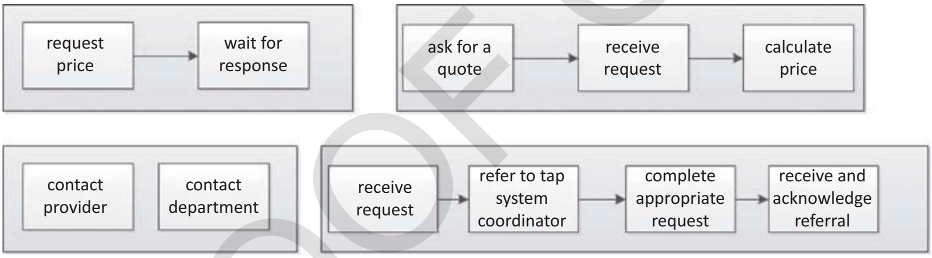

Figure 4. Contact Pattern.

	DPP C3: Inventory
Description:	The *Inventory pattern* relies on a collection of information and handles the inventory process. Inventories keep track of the stock of goods (material, product, article, service and supplies). The inventory process is initiated if invoices or reports about the inventory status of goods need to be generated.
Purpose:	Represents a series of process activities modelling the inventory process.
Problem:	The usage of this pattern addresses the involvement of activities concerning the stock.
Related Patterns:	Checking, Notification, Order, Production, Shipment, Update.
List of Labels:	**Verbs:** (check|update|evaluate|receive|allocate|change|delete|modify|search|request|inform), **Nouns:** (stock|inventory|(&& (information|status|level|goods|record)))
Keywords:	catalog, stock.

Figure 5 shows some examples of the *Inventory Pattern*.

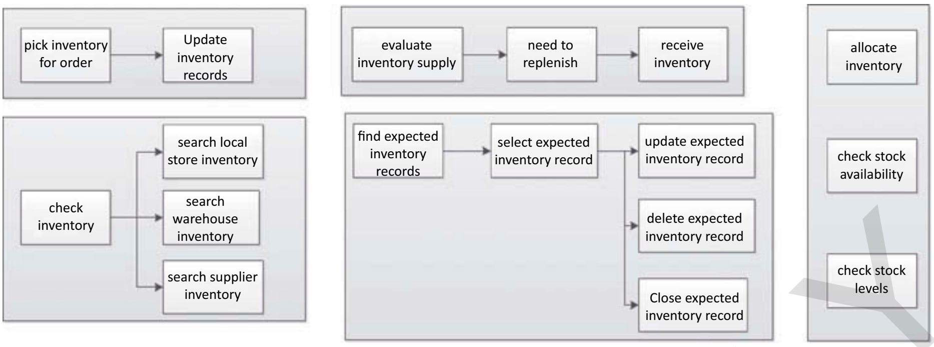
pick inventory for order
Update inventory records
check inventory
search local store inventory
search warehouse inventory
search supplier inventory
evaluate inventory supply
need to replenish
receive inventory
find expected inventory records
select expected inventory record
update expected inventory record
delete expected inventory record
Close expected inventory record
allocate inventory
check stock availability
check stock levels

Figure 5. Inventory Pattern.

	DPP C4: Invoicing
Description:	This pattern is used for an invoicing process in an end-to-end process. A seller prepares an invoice for the buyer according to the purchasing. The invoicing process is initiated if new invoices need to be generated (because goods have been shipped) or if payment needs to be handled. After creation of an invoice, the invoice will be sent to buyer via network. On the buyer side, the invoice will be received and then authorised.
Purpose:	Represents a series of process activities modelling the invoicing process.
Problem:	The usage of this pattern addresses an end-to-end process handling invoicing. Invoicing also implies the processing of goods
Related Patterns:	Accounting, Notification, Order, Production, Shipment.
List of Labels:	**Verbs:** (generate|create|send|debit|invoice|receive|authorise|charge), **Nouns:** (invoice|credit card|amount|goods|article)
Keywords:	billing, debit.

Figure 6 shows some examples of the *Invoicing Pattern*.

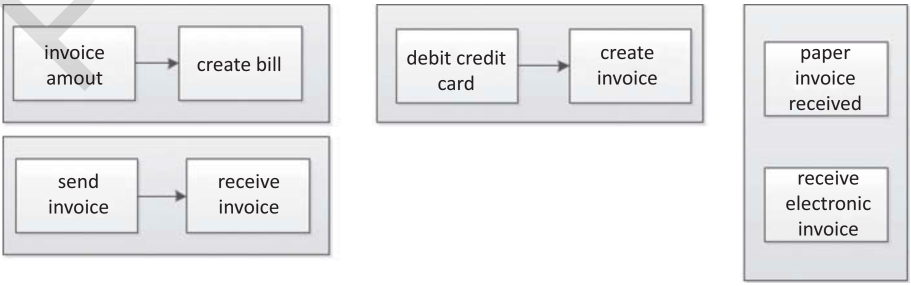

Figure 6. Invoicing Pattern.

	DPP C5: Order
Description:	The *Order Pattern* includes activities to order goods. Orders can be placed after a positive checking of requests or after an acceptance of an offer by the offeree. The control flow of the order process is from the time of taking of the order until the order has reached its final status e.g., cancelling or fulfilled to be shipped.
Purpose:	Represents a series of process activities modelling the order process.
Problem:	This pattern addresses the handling of all kind of activities required to fulfill an order
Related Patterns:	Checking, Inventory, Invoicing, Notification, Offer, Production, Shipment, Update.
List of Labels:	**Verbs:** (buy|purchase|deliver|send|receive|check|process|define|accept|reject|generate|release|wait), **Nouns:** (order|(&&customer))
Keywords:	Customer order, purchase order, buyer purchase order.

Figure 7 shows some examples of the Order Pattern.

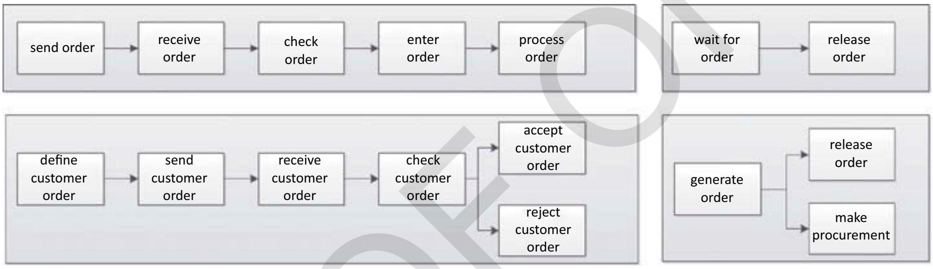

Figure 7. Order Pattern.

	DPP C6: Shipment
Description:	A shipment process generally takes place in two different contexts. A shipment process can be initiated after a request is received against an order. When a shipment process is in ‘drop shipment’, ordered goods are directly shipped to the customer from the supplier. In both cases, the seller (or direct supplier) is expected to deliver the ordered goods to the corresponding address indicated by a buyer. This pattern ends with the ordered goods that reach the address. This pattern may also involve all activities necessary to conform to the agreed performance, such as packing, transport or service delivery. A shipment may take place before or after payment, while an invoice can be shipped together or separately with a product.
Purpose:	Represents a series of process activities modelling the shipment process.
Problem:	The usage of the Shipment Pattern implies a successful result (e.g., order or production of goods) that needs further treatment.
Related Patterns:	Checking, Inventory, Invoicing, Notification, Order, Payment, Planning, Production.
List of Labels:	**Verbs:** (request|receive|decide), **Nouns:** (shipping|shipper|transport|(&&(information|schedule|price))); **Verbs:** (transport|deliver), **Nouns:** (order|goods)
Keywords:	Delivery, release, transport, packing, shipment scheduling.

Figure 8 shows some examples of the Shipment Pattern.

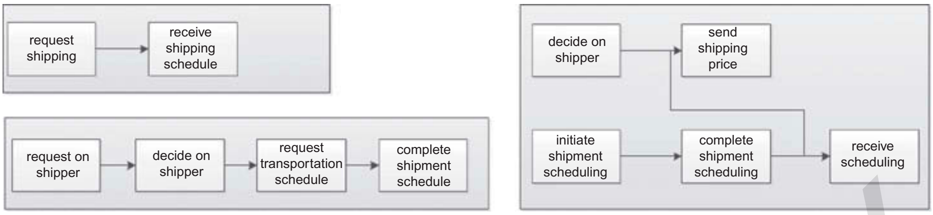
request shipping
receive shipping schedule
request on shipper
decide on shipper
request transportation schedule
complete shipment schedule
decide on shipper
send shipping price
initiate shipment scheduling
complete shipment scheduling
receive scheduling

Figure 8. Shipment Pattern.

DPP C7: Notification

Description: The *Notification pattern* represents a bi-directional interaction between two agents from the perspective of the notification sender. The pattern is initiated if business roles need to be informed about a process status (e.g., acceptance, rejection, cancellation, update, waiting). The goods to be handled can be accepted, rejected or has a pending status by a system or by a semi-automated process such as an approval.

Purpose: Represents a series of process activities modelling the notification process.

Problem: The usage of the Notification Pattern implies an information about the process status and a bi-directional interaction.

Related Patterns: Cancellation, Checking, Contact, Inventory, Offer, Order, Payment, Planning, Shipment.

List of Labels: **Verbs:** (inform|notify|send|), **Nouns:** (customer|person|pending|notification|confirmation|response)

Keywords: Acceptance, rejection, confirmation, cancellation, update.

Figure 9 shows some examples of the Notification Pattern.

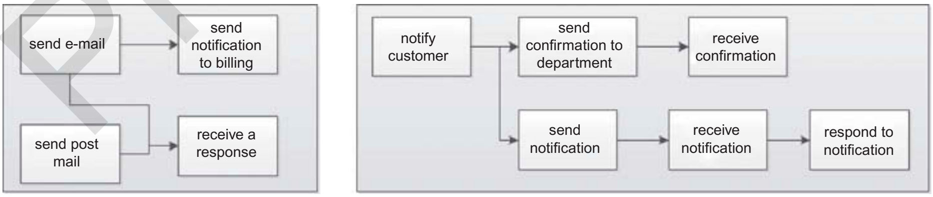

Figure 9. Notification Pattern.

4.4.2. Order management patterns

The following two patterns were found exclusively in the order management domain.

	DPP CO1: Offer
Description:	This pattern covers the time from a checked request to answering to the request and includes activities for offer proposition, offer acceptance, offer counseling and quote calculation. An offer describes activities for communicating to the supplier about an intention to fulfill a (goods) request. Types of offers vary from free offers, discounts or free quotes.
Purpose:	Represents a series of process activities modelling the offer process.
Problem:	The usage of the Offer Pattern implies an interaction between two agents about an offer.
Related Patterns:	Checking, Notification, Order.
List of Labels:	**Verbs:** (deceide|make|choose|accept|reject|offer), **Nouns:** (offer|selection|quotation)
Keywords:	Quote.

Figure 10 shows some examples of the Offer Pattern.

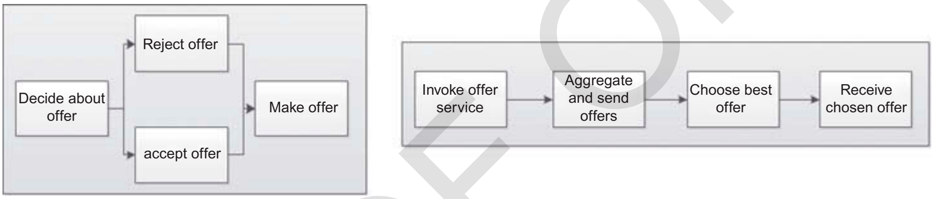

Figure 10. Offer Pattern.

	DPP CO2: Payment
Description:	The *Payment Pattern* is used to execute the payment process that will be launched to respond to received invoices. In the payment process, a buyer pays the agreed monetary value to a seller through a respective payment service providers or a bank. This pattern also includes activities to handle overdue payments and to remind customers about an outstanding payment.
Purpose:	Represents a series of process activities modelling the payment process.
Problem:	The Payment Pattern addresses the execution of activities for payment.
Related Patterns:	Checking, Notification, Shipment.
List of Labels:	**Verbs:** (remind|purchase|identify|request|receive|charge), **Nouns:** (payment|material|(&& (customer|method)))
Keywords:	Charge, Cash, Credit Card

Figure 11 shows some examples of the Payment Pattern.

4.4.3. Manufacturing production patterns

The next three patterns occurred only in the manufacturing production domain.

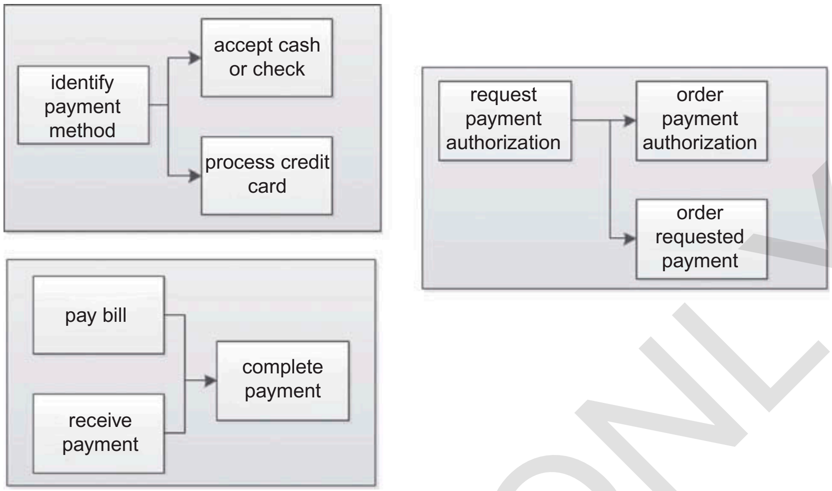

Figure 11. Payment Pattern.

	DPP CM1: Accounting
Description:	The *Accounting Pattern* describes the (financial) accounting of (production) goods. The *Accounting Pattern* keeps track of receivables, costs and expenses of goods and covers activities for accounting, financial services and bookkeeping.
Purpose:	Represents a series of process activities modelling the accounting process.
Problem:	The Accounting Pattern addresses the need for accounting goods.
Related Patterns:	Invoicing, Production.
List of Labels:	**Verbs:** (analyse|identify|determine|record|trail|adjust), **Nouns:** (transaction|account|balance)
Keywords:	Banking, bookkeeping.

Figure 12 shows some examples of the Accounting Pattern.

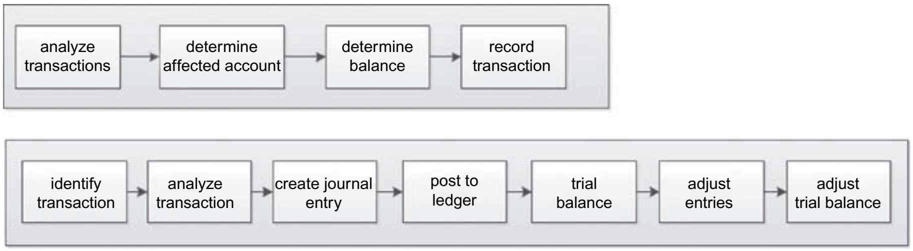

Figure 12. Accounting Pattern.

	DPP CM2: Planning
Description:	The *Planning Pattern* keeps track of activities ensuring that all components (raw materials) that are necessary to produce or ship goods are ready and available before starting the next phase. Planning of activities may involve interaction of participants that is handled with the Notification Pattern. This pattern also includes activities for making changes to goods that should be produced or shipped.
Purpose:	Represents a series of process activities modelling the planning process.
Problem:	The usage of the Planning Pattern implies that planning activities are required before starting or ending the production.
Related Patterns:	Notification, Production, Shipment.
List of Labels:	**Verbs:** (initiate|plan|prioritise|analyse|manage), **Nouns:**(forecasting|resource|material|operation|utilisation|capacity)
Keywords:	Design, production scheduling.

Figure 13 shows some examples of the Planning Pattern.

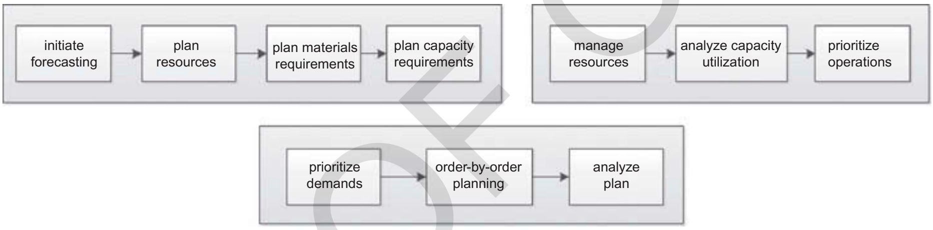

Figure 13. Planning Pattern.

	DPP CM3: Production
Description:	This pattern keeps track of activities for ensuring the production of goods starting from goods that are planned till the fulfillment of the production. The pattern also covers activities making sure that the quality is assured during and after production of goods. Activities used in the Production Pattern are production execution and production fulfillment. This pattern also covers activities for monitoring variations in components.
Purpose:	Represents a series of process activities modelling the production process.
Problem:	The usage of the Production Pattern implies the production of goods and addresses quality issues.
Related Patterns:	Checking, Order, Planning, Shipment.
List of Labels:	**Verbs:** (check|initiate|rollback|reject|assure|manage|monitor|execute), **Nouns:** (production|quality|(&&capacity))
Keywords:	Manufacture, assembly.

Figure 14 shows some examples of the Production Pattern.

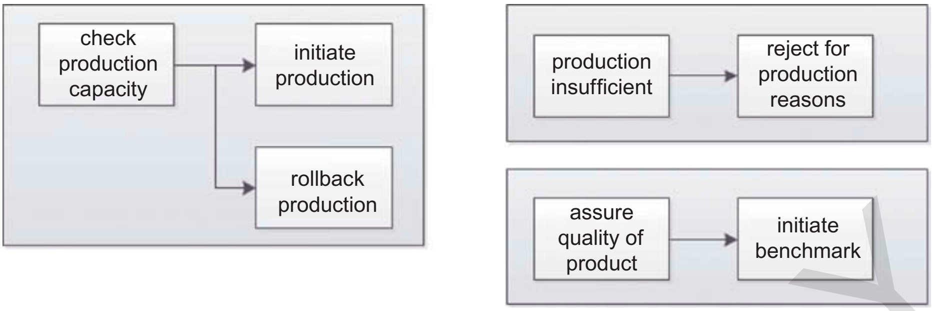
check production capacity
initiate production
rollback production
production insufficient
reject for production reasons
assure quality of product
initiate benchmark

Figure 14. Production Pattern.

5. Empirical findings for domain modelling

To investigate the ways process model parts are logically and ~~emantically~~ used during process modelling, 27 process models from the manufacturing production domain and 32 process models from the order management domain were investigated. The process models were given in four different modelling languages: UML Activity Diagrams, Petri nets, EPCs and BPMN. Ten process models from the order management domain have been created during an evaluation of the usefulness of the recommendation-based modelling support system. For these 10 process models, the process builders have defined the corresponding process model parts using the modelling support tool. For the remaining process models, the process model parts were extracted manually. Finally, the investigation presented in this paper is also based on 89 process model parts from the manufacturing production domain and on 91 process model parts from the order management domain.

The appendix shows an example business process model that is built with six DPPs. The benefits of this evaluation with respect to process reuse are:

- *Semantics-based model validation*. For this, the interest is on the relatedness (combination possibilities) of DPPs. Before suggesting a DPP for reuse, the system checks combinations of DPPs with the process model part (DPP) being edited in the active workspace. In case that no combination possibilities were detected between a DPP being edited and remaining DPPs, the ineligible DPPs are not further considered.
- *Decision guidance for selecting appropriate process model parts*. The popularity of a DPP might attribute how frequent a DPPs has been reused and refers to an implicit user feedback and patterns observed in other users' preferences. The process builder might favour a DPPs due to its popularity score.

5.1. Analysis approach

The authors calculated a relationship value R for three analysis settings:

- To analyse the occurrence of DPPs, R_{DPP_1} represents the relative frequency of a DPP within the set of 180 process model parts,

- To analyse the number of process model part combinations, R_{DPP_2} is the combination frequency of DPPs within the set of 59 business process models,
- To analyse the relatedness of DPPs, R_{DPP_3} expresses the relative frequency of a DPP within the set of considered business process models.

5.2. Analysing setting 1: general results

All DPPs could be found in the investigated process models. Patterns that occurred twice in a process model and belonged to different phases (e.g., the *Notification Pattern* and the *Planning Pattern*) were counted only once. Figure 15 shows the R_{DPP_1} value classified based upon the tripartition of process models. The amount of patterns belonging to the preparation phase is higher compared to the the processing and completion phase.

The occurrence of each single DPP is shown in Figure 16. The most frequently used DPP is the *Checking Pattern*, followed by the *Notification Pattern* and the *Order Pattern*.

5.3. Analysing setting 2: number of process model part combinations

Unfortunately, no comment can be given about the size of the companies using the business process models. The process models were investigated separately from the

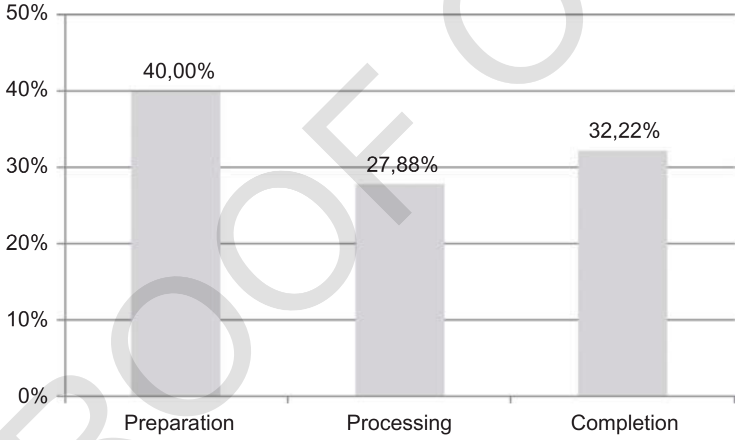

Figure 15. Occurrence of domain process patterns based on modelling phases.

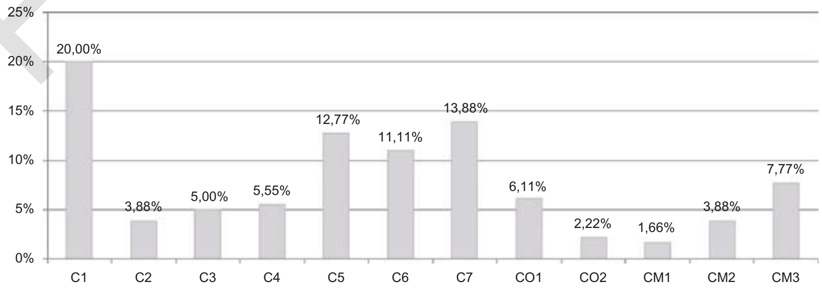

Figure 16. Occurrence of Domain Process Patterns. C1: Checking, C2: Contact, C3: Inventory, C4: Invoicing, C5: Order, C6: Shipment, C7: Notification, CO1: Offer, CO2: Payment, CM1: Accounting, CM2: Planning, CM3: Production.

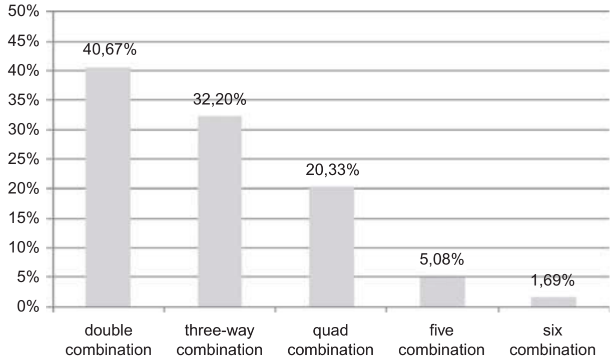

Figure 17. Occurrence of process model part combinations.

model size. Figure 17 shows the number of DPPs used to build a process model. In this setting, R_{DPP_2} was calculated. Most processes are built using two DPPs. One process 410 model was counted that has been constructed with six process model parts. However, the number of DPPs in a process model depends on the modelling language. While processes modelled with BPMN indicate a higher number of DPPs, Petri-net-based models use a low number of DPPs within a process model.

5.4. Analysing setting 3: combination of DPPs

Figures 18, 19 and 20 show in detail the combinations of the three most frequently used 415 patterns. Figure 18 shows that the *Checking Pattern* is frequently combined with the *Contact Pattern* (DPP C2), the *Notification Pattern* (DPP C8), the *Offer Pattern* (DPP CO1) and the *Order Pattern* (DPP C5). Other combinations pairs are only found occasionally. A combination between the *Checking Pattern* and the *Contact Pattern* indicates that an approval process is usually initiated due to a contact. The combination between the 420 *Checking Pattern* and the *Notification Pattern* indicates that the requester frequently receives a response after the checking phase has been completed. The *Checking Pattern* is completed with a release of an offer or an order (which is based on the frequent

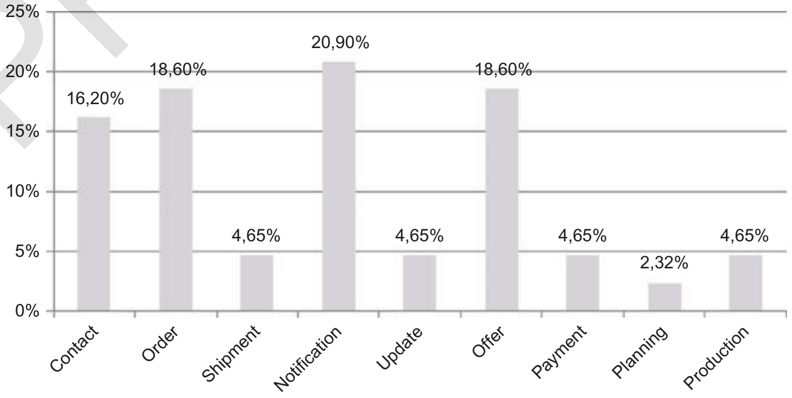

Figure 18. Combinations with Checking Pattern.

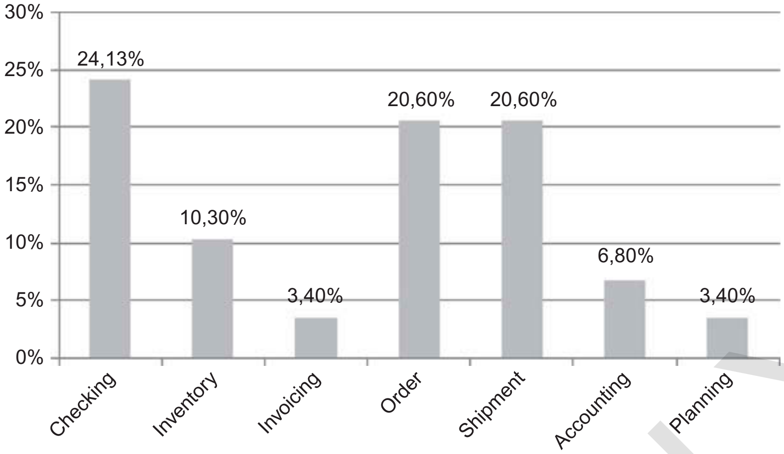

Figure 19. Combinations with Notification Pattern.

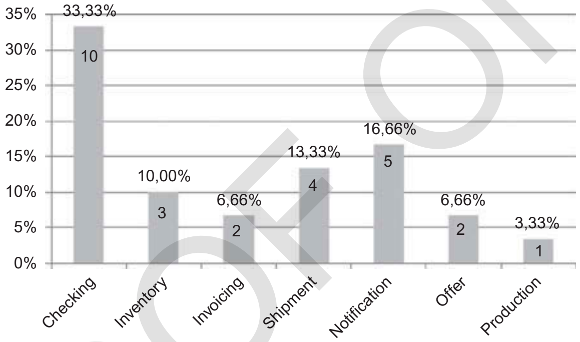

Figure 20. Combinations with Order Pattern

combination of the *Checking Pattern* with the *Offer Pattern* and the *Order Pattern*). When analysing the combinations of the *Notification Pattern* (DPP C8) (see Figure 19), it becomes evident that the *Checking Pattern* (DPP C1), the *Order Pattern* (DPP C5) and the *Shipment Pattern* (DPP C6) are often assembled with the *Notification Pattern*. One reason for that combination may be that data is processed in the three model parts requires communication for the initiating of further steps (e.g., a confirmation message that goods can be shipped).

The combinations with the *Order Pattern* is given in Figure 20. The most frequent combinations are found with the *Checking Pattern* (DPP C1), the *Notification Pattern* (DPP C8) and the *Shipment Pattern* (DPP C6). These combinations might indicate that an order is initiated after a request has been checked (due to frequent combination of order and checking). As soon as the order is fulfilled, the goods are shipped (combination of order and shipment).

The remaining DPPs are combined in the following ways:

- Accounting: Invoicing (40%), Production (60%),
- Contact: Checking (71.5%), Notification (28.5%),

- Inventory: Checking (9%), Notification (27.2%), Order (18.18%), Production (9%), Shipment (27.2%), Update (9%),
- Invoicing: Accounting (13.3%), Notification (6.6%), Order (20%), Production (26.6%), Shipment (33.3%),
- Offer: Checking (57.14%), Notification (21.42%), Order (21.42%),
- Payment: Checking (33%), Notification (33%), Shipment (33%),
- Planning: Notification (22.2%), Production (55.5%), Shipment (22.2%),
- Production: Checking (16.6%), Order (8.3%), Planning (41.6%), Shipment (33.3%),
- Shipment: Checking (6.6%), Inventory (10%), Invoicing (16.6%), Notification (20%), Order (13.3%), Payment (6.6%), Planning (6.6%), Production (19.6%),

Finally, the *Accounting Pattern* (DPP CM1) is frequently used with the Invoicing Pattern (DPP C4) and the Production Pattern (DPP CM3). The *Inventory Pattern* frequently occurs with the *Notification Pattern* (DPP C7), the *Shipment Pattern* (DPP C6) and the *Order Pattern* (DPP C5). The *Invoicing Pattern* (DPP C4) is performed in combination with the *Order Pattern* (DPP C5) and the *Shipment Pattern* (DPP C6). The *Offer Pattern* (DPP CO1) frequently appears with checking activities. The mostly used complement for the *Planning Pattern* (DPP CM2) is the Production Pattern (DPP CM3), while the latter is also often used in combination with the *Shipment Pattern* (DPP C6). The Shipment Pattern itself is often combined with several other patterns.

5.5. Summary of the evaluation

The evaluation shows the popularity and the relatedness of each single DPP within the investigated sample. To guarantee that the investigated models were designed for similar modelling intentions, the authors paid attention only to two application domains.

In the investigation set of process models, the authors could establish evidence for the existence of all DPPs. However, Figure 16 shows that some patterns occur only rarely, as measured relatively to the number of investigated process model parts.

Depending on the type of goods, different charging methods are initiated. Accounting is activated for production goods while non-production goods are charged with invoicing. In this evaluation, the combinations of DPPs based upon the type of goods were not considered. To cope with such analysis, Workflow Data Patterns (Russell et al. 2005) are an appropriate complementary candidate for DPPs.

The combinations of process model parts also depend on organisational policies. While in some organisations goods can only be shipped if they were payed for, in other organisations the shipment activity is performed without a pre-payment.

The authors also found out that some DPPs seem to be limited to a specific application domain. Therefore, a set of generic, core DPPs might exist that is applicable to far more modelling areas than presented in this paper. To support the wider dissemination and development of DPPs within and beyond the studied domains, an accompanying website has been set up as was mentioned earlier.

The next section shows how to use the evaluation results for a semantics-based model validation.

6. Domain process patterns: application and implementation

In this section, two application scenarios for DPPs are described. Domain Process Patterns facilitate and simplify the reuse of process model content that should be supported by means of a modelling tool. The first scenario provides the outline of precisely such a tool which would automatically suggest the appropriate process model parts for reuse. The second scenario discusses the use of DPP in the context of process model analysis. The benefit of DPPs within this scenario is a coherent description of process models.

Domain Process Patterns are also appropriate to improve the understanding of a process model by process model readers. A small number of approaches investigate the ways in which process models are built with respect to content and, therefore, become understandable to a user's perception. For instance, a business process describing activities of *production*, *shipment* and *payment* can be subsumed by a model reader as an in-house production process, while activities describing a *request*, an *offer*, a *notifation*, an *order* and a *shipment* characterise an order process, which is initiated by a request for which an offer is submitted and that is based on a notification the offer has been ordered. Showing to model readers only the keywords that characterise a process might already reduce the effort to understand the model. In such a situation, the benefit of DPPs for business process readers is that they can ignore the modelling notation and even must not be aware of it.

6.1. Application scenario: reuse of process model parts

To explain model reuse in the context of a recommendation-oriented tool, let the following scenario be given, as shown in Figure 21. A process builder intends to create a process model that processes an item. In case of item availability, the item should be shipped by an applicable shipper. The process builder has already modelled six process activities. Subsequently, she is not sure how to complete the process model. For this purpose, she requests the functionality 'Make Recommendations'. The modelling tool addresses these requests by querying a repository. Based on the data in the repository, the following

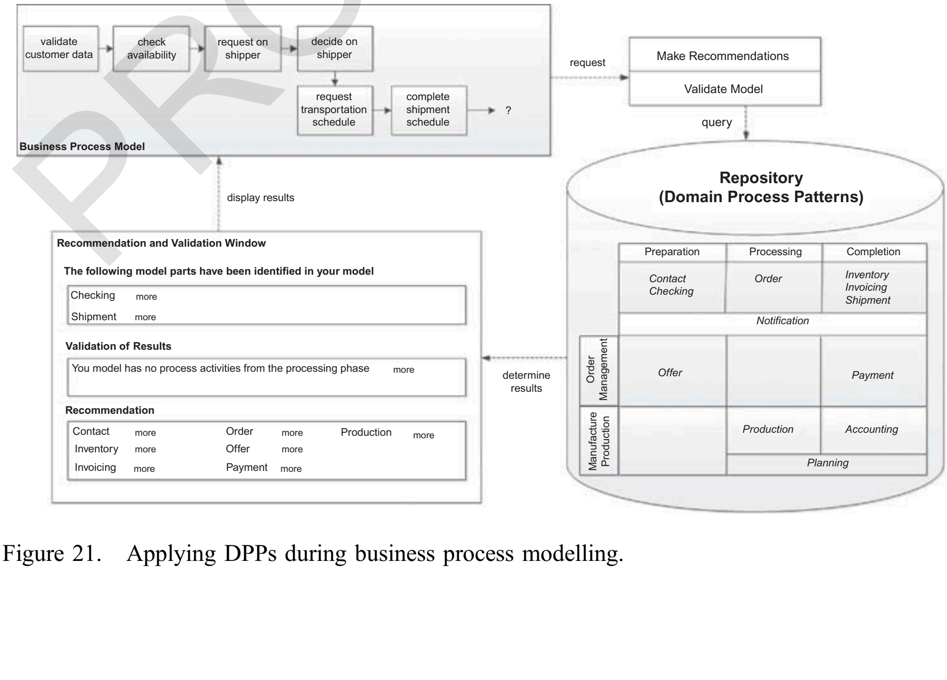

Figure 21. Applying DPPs during business process modelling.

results are determined and displayed to the process builder. The tool has identified both the Checking and Shipment DPPs (in the active workspace of the process builder). Since both patterns address the preparation and the completion phase, the tool recommends the process builder to consider process activities for processing the input (item) of the process model. Additionally, the tool recommends a list of DPPs, which it has identified as appropriate in this context. The Accounting DPP is not suggested because, based upon the evaluation results, it is neither combined with the Checking DPP nor with the Shipment DPP. If the process builder is interested in the popularity of a DPP and requires more detailed information about a specific DPP, she can continue with the ‘more’ information link.

6.2. Application scenario: process model analysis

Beside process reuse, DPPs can also be used for analysis or verification purposes – as was shortly mentioned in the introduction. DPPs allow for analysing the flow of process activities with respect to their reasonableness from a content perspective. Such a semantic-based verification of process models is only investigated in a limited fashion.[5]

Given Figure 21, assume the process builder also wants to validate her model with respect to ‘content’ (semantics) consistency. For this purpose she invokes the functionality ‘Validate Model’. Then, the system initially identifies the DPPs used in the active workspace of the process builder. In this example, the Checking and the Shipment DPPs were identified. Checking addresses the introductory (preparation) stage of order management or production management processes; While Shipment concludes these process models. No link between both phases exists. The system has detected that process elements from the processing phase are missing and suggests e.g., to use the Order DPP. The process model is then logically coherent when process elements from all three phase are used.

The recommendation of DPPs is based on the labels of process activities and does not undergo any syntactical verification (which can be performed based on control-flow patterns (Van Der Aalst et al. 2003)). Syntactical relationships have no impact on the recommendation. As soon as the modeller has completed the model, he or she can apply further verification techniques that help to uncover deadlocks.

6.3. Implementation

The recommendation of DPPs and its validation is implemented as part of a modelling support tool for process models (Koschmider, Hornung, and Oberweis 2011). This modelling support tool works on process models (parts) that are stored in the Petri Net Markup Language (PNML (Weber and Kindler 2003)). The PNML specification defines mandatory and optional elements. One optional element is `<toolspecific/>`. For the scope of application, this XML element has been extended with the attribute (`ref`, referring to a DPP) as shown in listing 1. All elements of a process model part are stored within the `<group/>` element. Further, this element has two attributes. The `name` attribute specifies the title and `ref` refers to a DPP that is described in detail within the `<pattern/>` element (see listing 2) in another XML file. The three elements `<objective/>`, `<description/>` and `<changeoption/>` are not newly inserted. However, they are explained for reasons of clarity. The element -`<objective/>` specifies the objective fulfilled by this process model (part) (e.g., modelling the approval of an order request) and the element `<description/>` provides a textual description of the process model (part). The element `<changeoption/>` indicates the number of reuses and shows the average number of deletions or insertions made when selecting a process model part.

Listing 1 Listing for pattern reference in PNML file

```
<toolspecific tool=“SemPeT” version = “v1.3”>
  <group name=“Item Checking” ref=“Checking”>
    <transition ref=“validate customer data”/>
    <transition ref=“check availability”/>
    <objective/>
    <description/>
    <changeoption/>
  </group>
</toolspecific>
```

The child elements of each `<pattern/>` element refer to attributes of DPPs.

Listing 2 Listing for the Checking DPP

```
<pattern>
  <name>Checking</name>
  <description>The pattern describes... </description>
  <relatedPatterns>
    <relatedPattern>Cancellation</relatedPattern>
    ....
  </relatedPatterns>
  <listLabels>
    <verbs>
     <verb>approve</verb>
     ....
    </verbs>
  </listLabels>
  <keywords>
    <keyword>Approval</keyword>
    ...
  <keywords>
</pattern>
```

Domain Process Patterns can be applied for every kind of business processes and they are independent of a modelling notation. The efficiency of DPPs correlates with the labelling quality of process activities. In case that activities are labelled imprecisely (e.g., do order) or different abstraction notions are used for labels (e.g., *order certificate is initiated* vs. *order is checked* where an order certificate is more concrete as an order), the quality of a DPP also decreases. To overcome this limitation, users can preview a popularity degree of a process model part (that is also shown with its release date). A low reuse number (low popularity) might indicate a low quality of a process model part.

7. Conclusion and outlook

Knowledge of modelling language syntax alone is not sufficient for building 'good' process models. Profound modelling experience is required to successfully apply a modelling language in practice. To ease the design of process models, several patterns have been proposed earlier that investigate the ways data, control flows, resources or quality constraints are used in workflows.

In this paper, the authors investigated the ways process model parts are combined during process modelling. As a result from that analysis DPPs are suggested, which are classified based on the three modelling phases of preparation, processing and completion. Domain Process Patterns capture particular business functions that frequently occur within processes of a particular modelling domain. These represent valuable, reusable knowledge on how process models are structured.

To confirm the occurrence of DPPs in actual models, the combinations of 190 process model parts from the order management and the manufacturing production domains were investigated. All DPPs could be identified in the investigated set of model parts. While some model parts frequently occur in almost all the investigated process models, other patterns depend on the type of considered goods (productive vs. non-productive) or organisational policies. An outlook was given on how to exploit the DPPs for supporting the actual modelling process by sketching the features of a modelling tool that would provide recommendations on their usage.

It remains to investigate the occurrence of DPPs in further modelling domains such as sales, distribution and marketing, by collecting large sets of process models from these domains. As more organisations start storing their process models in structured way, this appears a feasible endeavour. It is the aim of the authors to extend the collection of DPPs on the website that was set up at http://domain-process-patterns.jimdo.com/ and the authors warmly invite other researchers to contribute to this endeavour. The analysis of this evolving collection will confirm the contention that specific DPPs are broadly used across many modelling domains, while others will be confined to one single or only a few domains. Insights like these, along with an understanding of how such patterns are combined in practice, will contribute to the further development of advanced modelling tools, similar to the one that is suggested in this paper. The concrete support provided by such tools serve the ultimate aim to make the modelling process more efficient and effective.

Notes

1. See http://domain-process-patterns.jimdo.com/.
2. In Section 6 shows how process model reuse might benefit from such a tripartition.
3. In this paper the authors give only some examples for labels. A more comprehensive list of labels can be found on the DPP web page.
4. Unambiguities in activity labels such as synonyms and hypernyms can be detected using the approaches presented in Ehrig, Koschmider, and Oberweis (2007).
5. A good overview of this kind of verification is given in Fellmann et al. (2010a).

References

Barros, A., M. Dumas, and A. ter Hofstede. 2005. "Service Interaction Patterns." In *Proceeding of the 3rd International Conference on Business Process Management*. Vol. 3649 of *Lecture Notes in Computer Science*, 302–318. Berlin: Springer Verlag.

Bögl, A., M. Kobler, and M. Schre. 2008. “Knowledge Acquisition from EPC Models for Extraction of Process Patterns in Engineering Domains.” In *Multikonferenz Wirtschaftsinformatik*. Berlin: GITO-Verlag.

Davenport, T. 1993. *Process Innovation: Reengineering Work through Information Technology.* Boston, MA: Harvard Business School Press.

Dorn, [illegible] et al. 2010. “Self-adjusting Recommendations for People-driven Ad-hoc Processes.” In *Proceedings of 8th International Conference on Business Process Management*, September. Vol. 6336 of *Lecture Notes in Computer Science*. Berlin: Springer. AQ2

Ehrig, M., A. Koschmider, and A. Oberweis. 2007. “Measuring Similarity Between Semantic Business Process Models.” In *Proceedings of the Fourth Asia-Pacific Conference on Comceptual Modelling, Ballarat, Australia,* 71–80. Darlinghurst: Australian Computer Society.

Eshuis, R. and P. Grefen, 2008. “Constructing Customized Process Views.” *Data Knowledge Engineering* 64 (2): 419–438.

Fellm[illegible] M., et al. 2010a. “Whats Inside the Box? Prospects and Limitations of Semantic Verification in Process Modeling.” In *EMISA 2010 - Entwicklungsmethoden fr Informationssysteme und deren Anwendung*. Vol. 172 of *LNI.* AQ3

AQ4 Fellma[illegible] M., et al., 2010b. “An Ontology-driven Approach to Support Semantic Verification in Business Process Modeling.” In *MobIS 2010: Modellierung betrieblicher Informationssysteme - Modellgestütztes Management*, Vol. 171 of *LNI GI*, 99–110.

AQ5 Fern[illegible], 2000. “Stock Manager: An Analysis Pattern for Inventories.” In *Proceedings of PLoP 2000.*

Fern, E. B., X. Yuan, and R. Brey. 2000. “Analysis Patterns for the Order and Shipment of a Product.” In *Proceedings of PLoP 2000.* AQ6

Hao, T., et al. 2008. “Semantic Patterns for User-Interactive Question Answering.” *Concurrency and Computation: Practice and Experience* 20 (7): 783–799.

Hornung, T., A. Koschmider, and G. Lausen. 2008. “Recommendation Based Process Modeling Support: Method and User Experience.” In *Proceedings of the 27th International Conference on Conceptual Modeling*, Vol. 5231 of *Lecture Notes in Computer Science*, *OCT,* 265–278. Barcelona: Springer.

Koschmider, A., M. Song, and H. A. Reijers. 2010. “Social Software for Business Process Modeling.” *Journal of Information Technology* 25: 308–322.

Koschmider, A., T. Hornung, and A. Oberweis. 2011. “Recommen[illegible]on-based Editor for Business Process Modeling.” *Data and Knowledge Engineering* (to ap[illegible]). AQ7

Leopold, H., S. Smirnov, and J. Mendling. 2010. “Refactoring of Process Model Activity Labels.” In *Natural Language Processing and Information Systems*, Vol. 6177 of *Lecture Notes in Computer Science,* 268–276. Berlin: Springer.

Lincoln, M., M. Golani, and A. Gal. 2010. “Machine-Assisted Design of Business Process Models Using Descriptor Space Analysis.” In *Proceedings of 8th International Conference on Business Process Management*, Vol. 6336 of *Lecture Notes in Computer Science*. Berlin: Springer.

Lu, R., S. Sadiq, and G. Governatori. 2009. “On Managing Business Processes Variants.” *Data and Knowledge Engineering* 68 (7): 642–664.

Madhusudan, T., J. L. Zhao, and B. Marshall, 2004. “A Case-based Reasoning Framework for Workflow Model Management.” *Data Knowledge Engineering* 50 (1): 87–115.

Malone, T., K. Crowston, and G. Herman. 2003. *Organizing Business Knowledge: The MIT Process Handbook*. Cambridge: MIT Press.

Mansar, S. L. and F. R. H. A. Marir. 2003. “Case-based Reasoning as a Technique for Knowledge Management in Business Process Redesign.” *Electronic Journal on Knowledge Management* 1 (2): 113–124.

Markovic, I. and A. C. Pereira. 2007. “Towards a Formal Framework for Reuse in Business Process Modeling.” In *Business Process Management Workshops*, Vol. 4928 of *Lecture Notes in Computer Science*, 484–495. Berlin: Springer.

Matsuo, Y. and M. Ishizuka. 2003. “Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information.” In *FLAIRS Conference AAAI Press*, edited by I. Russell and S.M. Haller, 392–396. Menlo Park, CA: AAAI Press.

Mendling, J. and C. Simon. 2006. “Business Process Design by View Integration.” In *Business Process Management Workshops,* 55–64. Berlin: Springer.

Niedermann, [illegible] Radeschütz, and B. Mitschang, 2010. “Deep Business Optimization: A Platform for Automated Process Optimization.” In *INFORMATIK 2010 – Business Process and Service Science – Proceedings of ISSS and BPSC*, Vol. 177 of *LNI*, 168–180. AQ8

Norman, R. 1996. *Object-Oriented System Analysis and Design*. Englewood Cliffs, NJ: Prentice-Hall.

Reijers, H. A., R. S. Mans, and R. A. van der Toorn. 2009. “Improved Model Management with Aggregated Business Process Models.” *Data and Knowledge Engineering* 68 (2): 221–243.

Rosemann, M. 2006a. “Potential Pitfalls of Process Modeling: Part A.” *Business Process Management Journal* 12 (2): 249–3254.

Rosemann, M. 2006b. “Potential Pitfalls of Process Modeling: Part B.” *Business Process Management Journal* 12 (3): 377–384.

Russell, N., et al. 2005. “Workflow Data Patterns: Identification, Representation and Tool Support.” In *Proceeding of the 24th International Conference on Conceptual Modeling*, Vol. 3716 of *Lecture Notes in Computer Science,* 353–368. Berlin: Springer Verlag

Scheer, A. W. 1999. *ARIS – Business Process Modeling*. Berlin: Springer.

Sharp, A. and P. McDermott. 2008. *Workflow Modeling: Tools for Process Improvement and Applications Development*. Boston: Artech House Publishers.

Smirnov, S., et al. 2010. “Meronymy-Based Aggregation of Activities in Business Process Models.” In *29th International Conference on Conceptual Modeling*, Vol. 6412 of *Lecture Notes in Computer Science,* 1–14. Berlin: Springer.

Thom, L., M. Reichert, and C. Iochpe. 2009. “Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence.” *International Journal of Business Process Integration and Management (IJBPIM)* 4 (2): 93–110.

Van Der Aalst, W. M. P., et al. 2003. “Workow Patterns.” *Distributed and Parallel Databases* 14 (1): 5–51.

Weber, B. and M. Reichert, 2008. “Refactoring Process Models in Large Process Repositories.” In *Proceedings of the 2 International Conference on Advanced Information Systems Engineering*, 124–139. AQ9

Weber, M. and E. Kindler. 2003. “The Petri Net Markup Language.” In *Petri Net Technology for Communication-Based Systems, Advances in Petri Nets*, edited by H. Ehrig, 1–21. Berlin: Springer.

Appendix

Figure 22 shows a randomly selected BPMN business process model. You can find the business process of the role ‘myCompany’ when searching for ‘order management process BPMN’ in Google

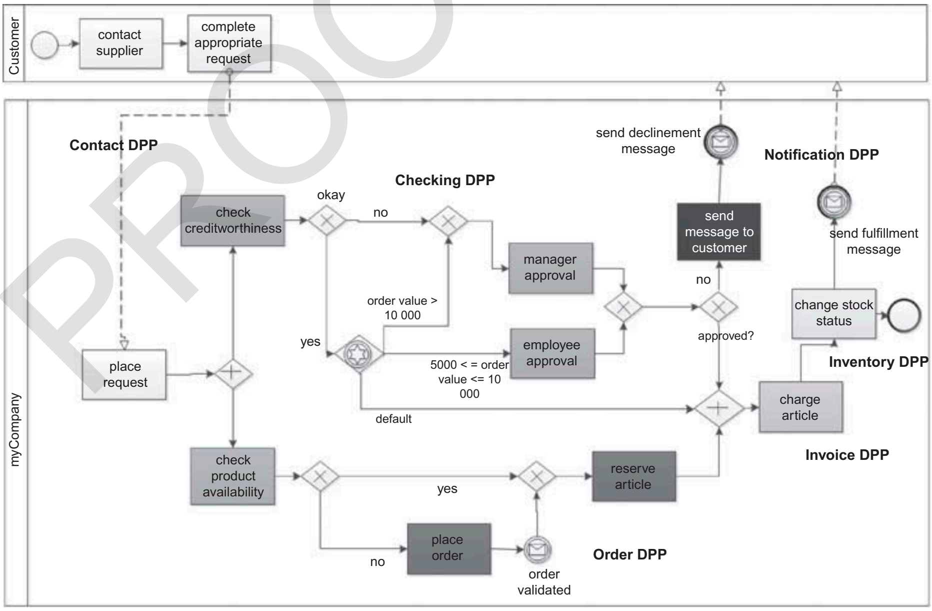

Figure 22. Example Business Processes with six DPPs.

Images. This process contains the six DPPs Contact, Checking, Order, Invoice, Inventory, Notification. These DPPs imply that the business process has been triggered by a role (e.g., customer) by posing a request. The request requires an approval where the result of the approval is an order that needs an invoice. The order also implies an inventory change. Finally, the customer is notified about the procedure.