Decomposition of Tasks in Business Process Outsourcing

Kurt Sandkuhl[1,3], Alexander Smirnov[2,3], Nikolay Shilov[2,3]

[1] The University of Rostock, Rostock, Germany
[2] SPIIRAS, St.Petersburg, Russia
[3] ITMO University, St.Petersburg, Russia
kurt.sandkuhl@uni-rostock.de, smir@iias.spb.su, nick@iias.spb.su

Abstract. In industrial areas with a highly competitive environment many enterprises consider outsourcing of IT-services as an option to reduce IT-related costs. In this context, cloud computing architectures and outsourcing of business processes into the cloud are potential candidates to improve resource utilization and to reduce operative IT-costs. In this paper, we focus on a specific aspect of cloud computing and outsourcing: the use of concepts from crowd-sourcing or crowd computing in business process outsourcing (BPO). The approach used in this paper is to bring together techniques from enterprise modeling and from crowd-computing for the purpose of business process decomposition. The contributions of the paper are an analysis of requirements to process decomposition from a business process outsourcing perspective, three different strategies for performing the decomposition and an initial validation of these strategies using an industrial case.

Keywords: Business Process Outsourcing, Crowdsourcing, Enterprise Modeling, Task Pattern, Process Decomposition

1 Introduction

In industrial areas with a highly competitive environment all enterprise departments and functions are expected to contribute to efficient operations and an economic cost structure. In particular the IT-budgets of organizations have been under pressure during the last decade with a clearly expressed expectation towards IT-departments to provide solutions and services of high quality tailored to business demands. Furthermore, many enterprises consider outsourcing of those IT-services as an option to reduce IT-related costs, which can be classified as commodities [1]. In this context, cloud computing architectures and outsourcing of business processes into the cloud are potential candidates to improve resource utilization and to reduce operative IT-costs [2]. Outsourcing of resources, products and competences could also be necessary due to continuous fluctuations and changes in business processes caused by the increased dynamicity of the modern global markets. This often happens for two reasons: the enterprise does not have enough capacities to fulfill the current demand of a certain resource / product / competence to solve all pertinent tasks in time, or the enterprise doesn't have required resource/products/competences corresponding to the current task.

But outsourcing approaches often are criticized for being not sufficiently flexible when automatable and manual tasks have to be combined [3]. In this paper, we focus on a specific aspect of cloud computing and outsourcing: the use of concepts from crowd-sourcing or crowd computing in business process outsourcing (BPO). Crowd computing is informally defined in [4] as “an umbrella term to define a myriad of tools that allow human interaction to exchange ideas, non-hierarchical decision making and full use of mental space of the globe”.

More concrete, the aspect addressed in this paper is decomposition of a process into what crowd-sourcing defines as “micro-tasks” (see section 2.2), i.e. smaller portions of the process which could be outsourced to a crowd-member. At first glance, a straightforward answer to this question seems to be that a business process anyhow consists of different activities, which can be considered as micro-tasks. Since business processes often have been modeled with a process modeling language (BPMN [6], EPC [7], flow charts [5] or similar), these process models could also serve as source for the micro-tasks. However, a closer analysis of real-world requirements reveals that constraints with respect to competences of the crowd-member, resources required for the task and the subject of the work have to be taken into account. Techniques for capturing and expressing such constraints are known from enterprise modeling. Thus, the approach used in this paper is to bring together techniques from enterprise modeling and from crowd-computing for the purpose of business process decomposition.

The contributions of the paper are an analysis of requirements to process decomposition from a business process outsourcing perspective, three different strategies for performing the decomposition and an initial validation of these strategies using an industrial case. The remainder of the paper is structured as follows: Section 2 briefly introduces the required background from enterprise modeling and crowdsourcing. Section 3 presents the industrial case from utility industries motivating the research and providing examples for business process outsourcing which can be studied to elicit requirements. Section 4 presents our approach for decomposing business processes into micro-tasks which includes three different strategies for different industrial demands. Section 5 evaluates the three strategies. Section 6 summarizes the work and discusses future research.

2 Background

This section briefly summarizes the background for our work from enterprise knowledge modeling and crowdsourcing. In enterprise modeling, our work is based on the task pattern approach (section 2.1) and in crowdsourcing on task decomposition into micro-tasks (section 2.2).

2.1 Enterprise Knowledge Modeling with Task Patterns

In general terms, enterprise modeling is addressing the systematic analysis and modeling of processes, organization structures, products structures, IT-systems or any other perspective relevant for the modeling purpose [8]. Sandkuhl et al. [9] provide a detailed account of enterprise modeling approaches. Enterprise models can be applied for various purposes, such as visualization of current processes and structures in an enterprise, process improvement and optimization, introduction of new IT solutions or analysis purposes. Enterprise knowledge modeling combines and extends approaches and techniques from enterprise modeling. The knowledge needed for performing a certain task in an enterprise or for acting in a certain role has to include the context of the individual, which requires including all relevant perspectives in the same model

[10]. A best practice for identifying these perspectives is the so-called “POPS*”-approach proposed by [11]. POPS* is an abbreviation for the perspective of an enterprise to be included in an enterprise model: process (P), organization structure (O), product (P), systems & resources (S) and other aspects required for the modelling purpose (*). The best practice basically recommends to always include the four POPS perspectives in a model because they are mutually reflective: process are performed by the roles captured in the organisation structure, the roles are using systems and resources which at the same time capture information about products; manufacturing and design of products is done in processes by roles using systems, etc. [10].

Patterns are a proven way to capture experts’ knowledge in fields where there are no simple “one size fits all” answers [12], such as enterprise modelling. Each pattern poses a specific design problem, discusses the considerations surrounding the problem, and presents an elegant solution that balances the various forces or drivers. The POPS* best practice was applied in the EU-FP6 project MAPPER to capture reusable portions of enterprise knowledge in so called task patterns [13], which proved feasible and economically rewarding [14]. Task patterns always include all four POPS perspectives and are represented in a visual modelling language.

2.2 Crowd Sourcing

Crowdsourcing is an emerging research area and it is usually understood (e.g., [15, 16]) as a form of outsourcing, in which tasks traditionally performed by organizational employees or other companies are sent through the internet to the members of an undefined large of group people (called “crowd”). The research area of massively parallel solution of problems with the help of “crowds” is still actively developing and there are several highly connected research areas: crowdsourcing, crowd computing, human computations, social computing, peer production. Boundaries between them are often blurred. Even more recent is the concept of hybrid crowd, where human solvers are accompanied by hardware and software services. This concept is more general, as it in some sense extends both classical service-oriented approach and crowd computing, but pertains all special effects inherent to crowd computing as a result of the inclusion of an outside human to an organization process in a transient, per-task basis.

In the context of this paper, the decomposition of organizational tasks into smaller portions, also called micro-tasks, is in focus. Micro-tasks are informally described as clearly defined sub-tasks of an organizational activity which are useful for the organization and clearly defined, can be performed independently by a crowd member, are economically affordable and involve no risk. Previous work on decomposition focused on a reference model [17], the use of the crowd for deciding on the decomposition [18] or construction of complex workflows out of micro-tasks [19].

3 Case Study

Research in this paper is motivated by an industrial case from business process outsourcing (BPO). The business service provider (BSP) studied in the case study is a medium-sized enterprise from Germany which offers more than 20 different BPO services to their clients. The target group for these services is medium-sized utility providers and other market roles of the energy sector in Germany, Bulgaria, Macedonia and several other European countries. Many energy distribution companies are outsourcing some business functions and business processes connected to these functions. Examples for typical business functions are meter readings, meter data evaluation, automatic billing, processing and examination of invoices, customer relationship management and order management. The BSP offers the performance of a complete business process for a business function or only of selected tasks of a business process. The IT-basis for these services in our case study is a software product which was developed and is maintained by the BSP. Integrated with a workflow engine and business activity monitoring, this software product provides the business logic for the energy sector, which is implemented using a database-centric approach. In addition to this software product, other cloud-based services for information exchange, document management and security are integrated. Different deployment models are used including a provider-centric model (the software product and the business processes are run at the BSP's computing center), a client-centric model (the software product is installed at the client site and the manual work of the business process is performed at the BSP) and mixed models (e.g. the software product is offered in the cloud, work and process are performed partly at the client and partly at the BSP).

When providing the outsourcing services for their clients, the BSP needs to offer the technical facilities for providing the service (see above) and the workforce for performing activities, which cannot be done in an automated way. Such activities often concern exceptions in the automated part of the processing, which will be illustrated in an example below. This example is depicted in Figure 1 and includes the POPS perspectives (see section 2.1), which all are required to completely describe what resources, roles, competences and products are needed to perform a given activity. The example was developed using the modelling tool Troux Architect[1] and the GEM[2] modelling language. The relationships in this model are typed, which usually is indicated by text showing the relationship type. For readability reasons, we had to switch off the visibility of the relationship texts.

The selected example is a process concerned with the communication between market roles in energy sector about meter readings and energy consumption of households. The core process is depicted in the middle of Figure 1 and includes the activities "receive and pre-process EDIFACT file", "check syntax of messages", "check model correctness", "validate message consistency", "perform transactions" and "acknowledge messages". This process basically shows all steps from receiving an EDIFACT[3] file with often thousands of messages, identifying the different messages by pre-processing the file, checking all messages for syntactical and semantic correctness, validating the soundness of the message content, recording the transaction and acknowledging it. This kind of EDIFACT message exchange is common practice between energy producers, distributors and grid operators, and is subject to regulation.

As long as no exceptions occur, the process can be performed along this "happy path" and is fully automated, i.e. no human actors are involved. But as soon as

[1] See www.troux.com

[2] GEM = Generic Enterprise Modeling language

[3] EDIFACT = Electronic Data Interchange for Administration, Commerce and Transport (cf. http://www.unece.org/cefact/edifact/welcome.html)

exceptions occur, the “knowledge workers” need to be involved, which is described by the alternative paths. One possible exception type is syntax errors in the EDIFACT file. In this case the process is aborted and an error message is sent to the sender. Another exception type is caused by a model error. In this case “correct model errors” is performed, which contains several activities: model errors could by caused by wrong/unknown message types or missing data depending on the message type.

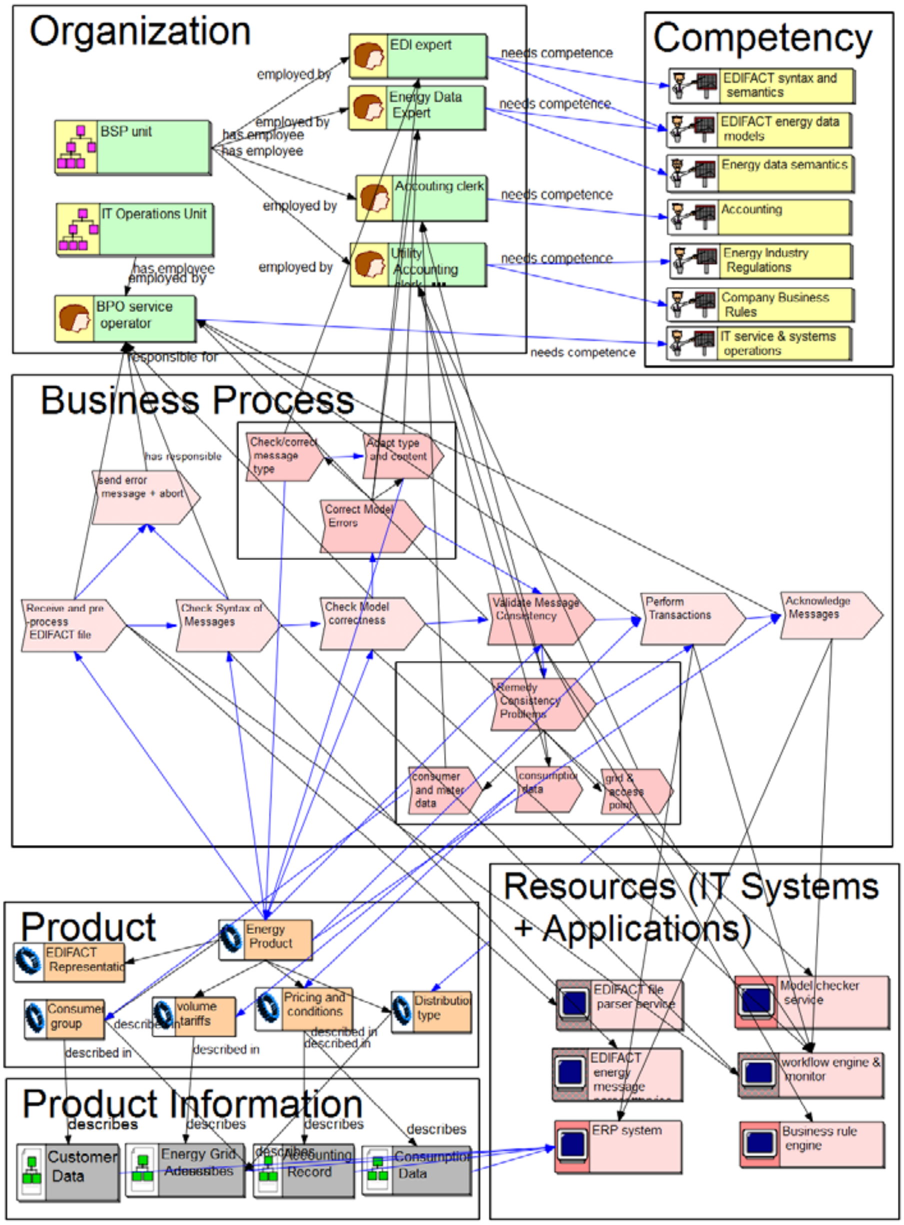


Figure 1: Visual model for Meter Data Processing following the task pattern concept

Such exceptions usually can only be remedied by human actors. This is done by an “energy data expert” and requires expert knowledge regarding the information model used in EDIFACT messages of the energy sector. In Figure 1, “correct model error” and its sub-activities are framed by a rectangle indicating that this is a task with refinements.

During validation of message consistency, several exceptions can happen: the address data of the consumer can be incorrect, e.g. if the consumer has several addresses and meters which were mixed up; the consumption data can be implausible; the grid data can be faulty. Some inconsistencies (e.g. of consumer data) can be corrected by an accountant without specific domain knowledge from energy sector. Others require competencies regarding the energy sector and business rules of the company and are performed by a utility accounting clerk. If the errors and inconsistencies can be remedied successfully, the messages can be processed by the ERP system and the transactions can be recorded. Products in the utility sector basically are different tariffs which depend on the customer group, the volume of consumption, the way of distribution (own energy grid or third party grid) and the pricing / payment conditions. The type of product and related product information are relevant for some steps in checking model correctness and crucial for validation of consistency and performing the actual transaction.

Resources required for performing the business process primarily are different information systems and services. A workflow engine controls and monitors the overall process flow. An ERP system for the energy sector manages all product information and performs the transactions. EDIFACT file parser, message parser and model checker are implemented as web-services. A rule engine is used to model and execute company business rules. In time periods with high workload or large numbers of exceptions, the BSP is interested in including external workforce in the performance of the BPO service. This could be an application case for using crowdsourcing, although the BSP so far does not have experience in this field. What activity could be outsourced to which outsourcing partner depends on different criteria: for most activities, the competence requirements are decisive, i.e. only a sub-supplier (or crowd-member) with the right competence could possibly be considered as outsourcing partner. Furthermore, the availability of resources and the access to product details or product information can be important. In case of confidential product details, this aspect might be the decisive one when considering outsourcing and in case of use of internal resources provided in a private cloud only, the resource aspect is dominant for decision making.

4 Strategies for Decomposition of Business Process Models

This section proposes and specifies different strategies for the decomposition of business process models into smaller portions, which in crowdsourcing are called micro-tasks (see section 2.2). The purpose of this decomposition is to identify tasks potentially suitable for crowdsourcing including the competences, resources and product information needed for these tasks. All strategies follow the same general approach:

- starting point for each strategy is a business process model which includes the POPS perspectives according to the task pattern concept, e.g. a model like the example illustrated in Figure 1,
- the aim is to isolate micro-tasks from the process model which also must follow the task pattern concept (see section 2.1), i.e. which have to include the POPS

perspectives in order to fully specify activity, competences, resources and product information needed.

- the strategies are based on structural patterns in the process model. In this paper we do not take into account runtime information and we do not analyze the textual parts of the model elements, but only their types.
- the strategies reflect the different priorities from the industrial case, i.e. to consider competence, resource or product information as highest priority
- the structural patterns of each strategy at the same time form the structure of the micro-tasks, i.e. if the model has been fully decomposed into patterns, the required micro-tasks are defined.
- we assume that the resources are not consumed by the activities. This assumption is possible as we focus on business process outsourcing using digital resources (IT systems, services) and not on conventional machinery or consumable resources in manufacturing industry.

The competence-first strategy follows the principle that a portion of a process only can be outsourced, if a supplier or crowd-member can be identified who has all competences required by the role(s) assigned to the process-portion under consideration. Thus, the strategy starts from roles and competences in the process model and attempts to segment the process into portions with as few as possible process steps. These process steps together with the resources, product details, roles and competences assigned to them from the micro-tasks. The resource-first strategy assumes that the availability of all resources required for a process-portion is decisive when outsourcing it. The strategy starts from the resources in the process model and attempts to segment the process into portions with as few as possible process steps.

The product-first assumes that the access to product information is decisive, starts from here and continues like the resource-first strategy. When discussing the strategies in 4.1 to 4.3, we use a symbol set which is illustrated by the generic model shown in Figure 2. In this particular example there is a process with two tasks (which both are potential micro-tasks), each of them is associated with one IT resource and one role. Both of the tasks are associated with one common product. Each role is associated with one competence. For simplicity, only tasks and corresponding resources (product or IT resource or role together with competence) are shown in the figures in section 4.1 to 4.3 illustrating the patterns.

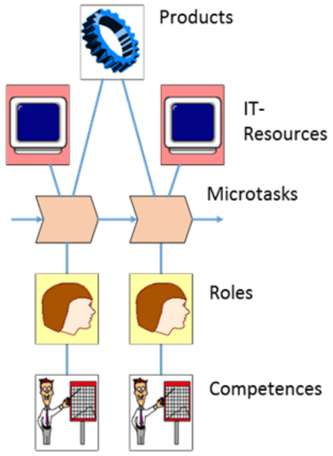
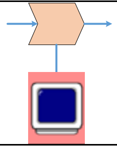
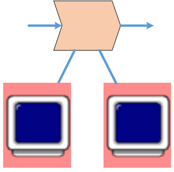
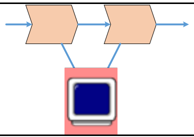
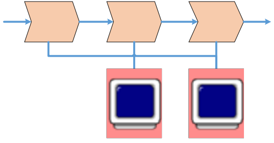


Figure 2: Symbols used when describing the strategies illustrated in a generic model

4.1 Resource-first strategy

The motivation for this strategy is that missing or insufficient capacity of IT resources for business process implementation could require outsourcing of those tasks requiring these resources. Thus, the strategy assumes that the availability of resources is the primary precondition for crowdsourcing, i.e. decomposing a process consisting of tasks into micro-tasks requires that for each micro-task all required resources are available at the outsourcing partner. By examining all possible task - resource multiplicities, the patterns depicted in table 1 were identified. The patterns below illustrate the possible situations and suggest corresponding outsourcing strategies.

Table 1. Resource-first strategy patterns

#	Diagram	Description
1.		This is the first and the most simple and straightforward pattern when a task of the business process requires one resource. The solution for this pattern is to consider this task as a microtask for outsourcing.
2.		In this pattern, a task requires at least two resources and there is a lack for at least one of them which is the reason for outsourcing. There could be two possible solutions here. The first one is to try to split the task into two microtasks so that the one, which requires a missing resource, could be outsourced and the other could remain within the company. This is especially recommended if this microtask deals with some sensitive information, which the company would prefer not to share with third parties. If splitting the task is not possible, the whole tasks is a microtask to be outsourced.
3.		In this pattern, two or more sequential tasks require the same resource. The solution is straightforward: the microtask to be outsourced consists of all sequential tasks requiring the resource.
4.		In this pattern, two tasks require a resource, which has to be outsourced. However, between these tasks, there is one or more other tasks, which can be performed internally. There could be two possible solutions here: outsource all tasks as one microtask or only those tasks, which require the missing resource. The solution might depend on the sensitivity of the information used in the intermediate tasks: the more sensible the information, the more reasons to keep these tasks inside the company.

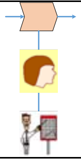
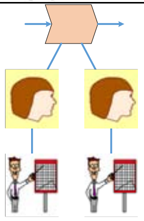
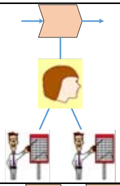
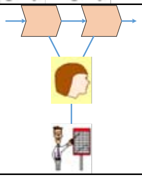
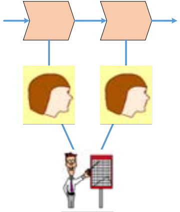
4.2 Product-first strategy

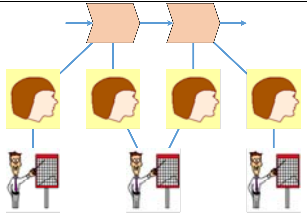
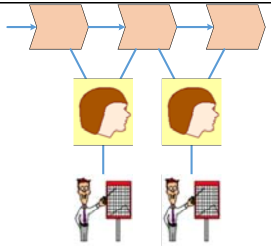
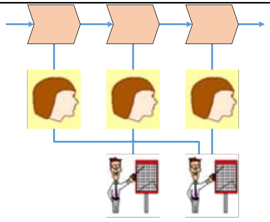
The product-first strategy is motivated by the fact that different tasks in a business process require different parts of the product or the product information. “Product” in this context can also be a service and “product part” a portion of information required for this service. Outsourcing “product-first” could be motivated by the fact that only certain parts of the product details shall be disclosed to someone outside the enterprise. However, if a task requires certain product details, all these product details have to be made available to the corresponding micro-task. Similar to the resource-first strategy, we examined all possible task - product multiplicities. The patterns identified were those already shown in table 1 for the resource-first strategy, if we substitute the “resource” symbols in these patterns with a “product” symbol. Thus, there are also four patterns (#5 to 8) for the product-first strategy.

4.3 Competence-first strategy

The competence-first strategy is different from the previous ones because usually in enterprise modelling there is no direct link between competences and tasks, but tasks are related through user roles. The reason for using this strategy is obviously a lack of required competences. Due to the increased complexity caused by the intermediate layer of user roles it has eight patterns.

Table 2. Competence-first strategy patterns

#	*Diagram*	*Description*
9.		Simple and straightforward pattern when a microtask requires one role and one competence, which has to be outsourced. The solution is the same as for the previous strategies: outsource the appropriate microtask.
10.		In this pattern, a microtask requires at least two roles, with only one of the roles requiring a missing competence, which has to be outsourced. There could be two possible solutions here. The first one is to try to split the microtask into two so that the one, which requires a role with missing competence, could be outsourced and the other could remain within the company. If splitting the microtask is not possible, than the whole microtask has to be outsourced.
11.		In this pattern, a microtask requires a role that requires several competences, with one of the competences requiring outsourcing. Since the role cannot be split, the only solution here is to outsource the whole microtask.
12.		This pattern is also very similar to the previously described strategies: two or more microtasks require a role with a competence, which has to be outsourced. The solution is to outsource the appropriate microtasks.
13.		In this pattern, two sequential microtasks require different roles; however, both of the roles need a missing competence. The solution is to outsource both of the microtasks.

#	*Diagram*	*Description*
14.		This pattern is similar to the previous one; however, each of the two microtasks is also associated with roles, which do not require outsourcing. This condition produces an idea to split the microtasks so that some of the microtasks could be outsourced and others could stay within the company. If splitting of at least one of the microtasks is not possible, than the need to split the other one should also be re-considered.
15.		In this pattern, a microtask (in the middle) requires two roles, with one of the roles (e.g. the one on the right) requiring a missing competence, which has to be outsourced. There could be two possible solutions here. The first one is to try to split the microtask into two so that the one which requires the role with missing competence could be outsourced together with the following microtask, and the other could remain within the company with the previous microtask. If splitting the microtask is not possible, than the two microtasks (the one in the middle and the following one) have to be outsourced.
16.		In this pattern, two microtasks require roles with a competence that has to be outsourced. However, between these microtasks, there are one or more other microtasks, which can be performed internally. There could be two possible solutions here: outsource all microtasks or only those, which require the missing competence. The reasons to choose one of the solutions could be different and the decision has to be based on the particular circumstances.

5 Case Study-based Evaluation of the Strategies

This section discusses the evaluation of the strategies proposed in section 4. This evaluation focuses on three different questions, which basically address feasibility, usefulness and differences of the strategies:

1. Can the strategies be applied for decomposing real-world business process models into micro-tasks? This includes whether it is possible to identify micro-tasks and whether the business process can be fully decomposed.
2. What differences exist between the different strategies? This question includes (a) differences between individual micro-tasks and (b) usefulness of the sets of micro-tasks for practical use.
3. Are the micro-tasks identified by the strategies applicable for outsourcing tasks in practice? This includes whether the complete business process can be (re-) composed from the micro-tasks and whether isolated performance of micro-tasks would be possible.

In order to answer questions 1 and 2(a), the process model presented in section 3 was decomposed into micro-tasks with all three strategies and the resulting sets of micro-tasks were compared. For questions 2(b) and 3, an expert evaluation of the micro-tasks was performed. Table 3 shows the result of applying the different strategies for decomposing the business process introduced in section 3. The column “Str.” indicates, which strategy or strategies produced the micro-task (C = competence-first; P = product-first; R = resource-first). In the column “organization” role and competence are separated with a “/”; in all columns, whenever two or more processes, roles, product details or resources are required, they are separated with a “;”.

Table 3 also shows that the three strategies produce different sets of micro-tasks which have some overlap. Resource-first produces the smallest number of micro-tasks with only 7 elements, but with the largest tasks of all strategies (#15). The reason for this is that the resource “workflow engine” is needed for all activities in #15 while each of the individual tasks in #15 also requires at least one additional resource. Production-first and competence-first show only a difference regarding the micro-tasks identified for the tasks included in “remedy consistency problems” (see Figure 1).

Table 3: Micro-tasks produced by the different strategies

#	*Str.*	*Process*	*Organization (role/competence)*	*Product detail*	*System (Resource)*
1	C P	Receive and pre-process EDIFACT file	BPO service operator / IT service & system operations	EDIFACT representation	EDIFACT file parser service; workflow engine (WE)
2	C R P	Send error message + abort			EDIFACT file parser service; EDIFACT energy message parser
3	C P	Check syntax of message			EDIFACT energy message parser; WE
4	C P	Check model correctness			Model checker service; WE
5	C P	Validate message consistency		Consumer group; distribution types; pricing and conditions	Business Rule engine; WE
6	C P	Perform transactions		Consumer group; volume tariffs; distribution types; pricing and conditions	ERP system; WE
7	C P	Acknowledge message		EDIFACT representation	
8	C R P	Check/correct message type	EDIFACT expert / EDIFACT syntax and semantics	EDIFACT representation	Model checker service; WE
9	C R P	Align data & content	Energy data expert / EDIFACT energy data representation		
10	C	Validate consumer and	Accounting clerk /	Consumer	ERP system;

	R	meter data	accounting	group; distribution types	WE
11	C	Validate consumption data; validate grid and access data	Utility accounting clerk / energy industry regulations; company business rules	Consumer group; distribution types; pricing and conditions	
12	R	Validate consumption data			
13	R	validate grid and access data			
14	P	Validate consumer and meter data ; validate consumption data; validate grid and access data	Accounting clerk / accounting; Utility accounting clerk / energy industry regulations; company business rules		
15	R	Receive and pre-process EDIFACT file; Check syntax of message; Check model correctness; Validate message consistency; Perform transactions; Acknowledge message	BPO service operator / IT service & system operations	EDIFACT representation ; consumer group; volume tariffs; distribution types; pricing and conditions	EDIFACT file parser service; EDIFACT energy message parser; Model checker service; Business Rule engine; ERP system; (WE)

For the expert evaluation, we needed an experienced professional from the BPO domain. One of the BPO product managers of the BSP introduced in section 3 was selected. This expert had more than three years of experience in this position. The expert evaluation included three steps. The first step was to go through all micro-tasks identified by the three strategies, i.e. the set of all sets. The expert had to answer the question for each micro-task whether the micro-task was completely defined, i.e. all resources, competences and product details are included. This check related to the task pattern feature of being “self-contained”. As a result, for one micro-task a missing resource was discovered. It showed that this was due to a mistake in the model where the resource was not connected to the activity with the required relationship type. All other micro-tasks were found to be self-contained. The second step was to judge the different sets of micro-tasks produced by the three strategies whether they fully define the business process and which of the sets was considered by the expert as the best one and why. According to the expect judgment, the set produced by the resource-first strategy is the best one because it keeps the “happy flow” of the business process (i.e. micro-tasks #1 and #3 to #7) together. Further discussion with the expert shows that high throughput is important for this fully automatic flow which is another argument for keeping the happy flow integrated. The only problematic result of the resource-first strategy is according to the expert the separation of “validate consumption data” and “validate grid and access data” (i.1. #12 and #13) since these two steps are tightly interwoven. The comparison of competence-first and product-first shows advantages for competence-first, mainly because competence-first separates the “validation of consumer and meter data" (#10) from the other validation tasks (#11) while product-first keeps them together (#14). "Validation of consumer and meter data" requires less knowledge and could be performed by a less-experienced knowledge worker.

The third step was to ask the expert to select the micro-tasks out of the set of sets which could be outsourced to the crowd, i.e. where the required resources could be provided in a public cloud or via secure access paths to the private cloud, where the product detail of information was not affected by a high level of confidentiality and where it can be expected that a substantial number of people have the required competences. The expert identified only one task suitable for crowdsourcing from his perspective: "validation of consumer and meter data" (#10).

6 Summary and Future Work

Motivated by an industrial case, the paper investigated strategies for decomposing business processes into micro-tasks in the context of crowdsourcing. The three strategies put an emphasis on different priorities of enterprises by starting from resources, competences or product details when deciding on decomposition. The three strategies proved to be feasible and – at least in the industrial case investigated – also produced sound proposals for micro-tasks.

The use of the task pattern approach with the POPS perspectives proved to be both, feasible and useful. In the industrial case it helped to clearly identify the prerequisites for performing micro-tasks in terms of product information, required competences and resources. However, task patterns originally were meant to capture reusable organizational knowledge, while micro-tasks cannot be considered as portions of organizational knowledge since the required context of use, i.e. the overall business process and organization structures, is not fully defined which does not make it an "asset" for the organization.

Although the decomposition of the business process example from section 3 easily could be performed manually, the algorithmic nature of the strategies calls for the implementation of a software tool performing the decomposition. This will be part of the future work and can form the basis for a quantitative evaluation of the strategies, e.g. by using the task pattern collection developed in the MAPPER project.

Future work also needs to investigate potential refinements of the strategies

- runtime information should be taken into account, if available. Many business process models include information about expected execution times of tasks which could be a first hint for refining the strategies,
- the textual parts of the model elements could be analyzed and not only their types. The semantics of the text could indicate how tightly different tasks are related to each other which could recommend to keep them together,
- potentially required changes in strategy have to be investigated for the case that resources are consumed by the activities, like conventional machinery or consumable resources in manufacturing industry.

Our work so far has a number of limitations: we used just one case and one expert for evaluation and motivation purposes. Other cases could show different requirements which may affect the utility of the strategies. The evaluation of the strategies is far complete. We did not include perspectives, like implementability, performance or acceptance, which again might lead to refined strategies or different approaches.

Acknowledgements

This work was partially financially supported by the Project 213 within the research program I.5P of the Russian Academy of Sciences, by Government of Russian Federation, Grant 074-U01. Furthermore, it was partly financed by the German Ministry of Research and Education, research project KOSMOS-2.

References

1. N. G. Carr, “IT doesn't matter,” *Engineering Management Review, IEEE*, vol. 32, no. 1, 2004.
2. Krogstie J. (2012). *Model-Based Development and Evolution of Information Systems - A Quality Approach*. Springer. London.
3. Thuan N.H., Antunes P. and Johnstone D. (2013). Factors Influencing the Decision to Crowdsource. *Proceedings of the 19th International Conference CRIWG 2013*, Wellington, New Zeland, Oct. 30 – Nov. 1, 2013. Springer, LNCS, vol. 8224, pp. 110-125.
4. Schneider D, de Souza J and Moraes K. (2011). Multidões: a nova onda do CSCW? In: *8th Brazilian Symposium on Collaborative Systems* SBSC
5. Knuth, D.E. (1963). Computer-drawn flowcharts. Communications of the ACM, 6(9), 555-563.
6. White, S. A. (2004). Introduction to BPMN. IBM Cooperation, 2(0), 0.
7. Scheer, A.-W. and M. Nüttgens (2000). ARIS architecture and reference models for business process management. Berlin Heidelberg: Springer.
8. Vernadat, F. B. (1996). Enterprise Modelling and Integration. Chapman & Hall, 1996.
9. Sandkuhl, K.; Stirna, J.; Persson, A. and M. Wißotzki (2014) Enterprise Modeling: Tackling Business Challenges with the 4EM Method (The Enterprise Engineering Series). Springer Verlag, Berlin Heidelberg. ISBN 978-3662437247.
10. Lillehagen, F.; Krogstie, J. (2009) Active Knowledge Modelling of Enterprises. Springer, 2009. ISBN: 978-3-540-79415-8.
11. Lillehagen, F. (2003), The Foundations of AKM Technology, *10th International Conference on Concurrent Engineering (CE) Conference,* Madeira, Portugal.
12. Enterprise Integration Patterns, http://www.eaipatterns.com/ (Accessed 26 March 2012).
13. Sandkuhl, K., Smirnov, A., Shilov, N. (2007) Configuration of Automotive Collaborative Engineering and Flexible Supply Networks. In Cunningham, P. and Cunningham, M. (Eds.): Expanding the Knowledge Economy – Issues, Applications, Case Studies. ISBN 978-1-58603-801-4.
14. Sandkuhl, K. (2010) Capturing Product Development Knowledge with Task Patterns: Evaluation of Economic Effects. Quarterly Journal of Control & Cybernetics, Issue 1, 2010. Systems Research Institute, Polish Academy of Sciences
15. Schenk, E. and Guittard C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, 2011(1), 93-107.
16. Howe, J. (2006a). The rise of crowdsourcing, in Wired magazine, 2006, Dorsey Press. p. 1-4.
17. Smirnov, A.; Ponomarev, A.; Shilov, N.. Hybrid crowd-based decision support in business processes. CENTERIS 2014 (15-17 10.2014, Lisbon, Portugal), vol. 16, 2014, pp. 376-384.
18. Kulkarni, A., Can, M., & Hartmann, B. (2012). Collaboratively crowdsourcing workflows with turkomatic. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (pp. 1003-1012). ACM.
19. Little, G., Chilton, L. B., Goldman, M., & Miller, R. C. (2010). Turkit: human computation algorithms on mechanical turk. In Proceedings of the 23nd annual ACM symposium on User interface software and technology (pp. 57-66). ACM.