



HAL
open science

Integrating Local and Global Optimization in Capability Delivery

Kurt Sandkuhl

► **To cite this version:**

Kurt Sandkuhl. Integrating Local and Global Optimization in Capability Delivery. 10th IFIP Working Conference on The Practice of Enterprise Modeling (PoEM), Nov 2017, Leuven, Belgium. pp.341-351, 10.1007/978-3-319-70241-4_23 . hal-01765251

HAL Id: hal-01765251

<https://inria.hal.science/hal-01765251v1>

Submitted on 12 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Integrating Local and Global Optimization in Capability Delivery

Kurt Sandkuhl

University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, Germany
kurt.sandkuhl@uni-rostock.de

Abstract. Efficient value creation and service delivery processes are considered as key factor to competitiveness in a globalized market. The systematic management of the capabilities of an enterprise is emerging into a key activity for achieving efficiency. In capability management, the capability design and delivery approach (CDD) has been proposed and its utility was demonstrated in previous work. An essential activity in CDD is the analysis of the context of capability delivery in order to understand the cause of variations in delivery of capabilities. This paper builds on CDD and elaborates on the feature of CDD to combine local context information from individual clients with global context information from the capability provider when optimizing capability delivery. An additional method component for the CDD methodology is proposed and demonstrated using a case from business process outsourcing.

Keywords: capability, BPO, local context, global context

1 Introduction

In many industrial sectors efficient value creation and service delivery processes are considered as the key factor to competitiveness in a globalized market environment with information technology as enabler and strategic instrument for service management. The systematic management of the capabilities of an enterprise, which often are reflected in the business services offered to customers and the technical services associated to them, is emerging into a key activity for achieving efficiency. The term capability is used in various industrial and academic contexts with often different meanings (see section 4.1). Most conceptualizations agree that capability includes the ability to do something (know-how, organizational preparedness, competences) and the capacity for actual delivery in an application context.

In order to facilitate capability management, we propose to not only consider business services but also to explicitly analyze their delivery contexts and to model both, the service and the context. This approach, which is called capability-driven design and development (CDD), aims at capturing the factors that are decisive for flexibility, dynamics and variability. CDD is grounded in industrial requirements and includes method components, a common meta-model and a tool environment (see section 2). The method provides mechanisms for adjustment of capability delivery.

For business process outsourcing (BPO), which is in focus of the industrial case selected for this paper (see section 2.1), the adjustments during capability delivery are

based on the service level agreements between client and business process service provider (BSP) as well as on BSP-internal policies, such as the responsible units for a capability and their schedule. Currently, the CDD approach includes primarily adaptation mechanisms, which focus on this “local view” for separate capabilities and/or clients. However, in industrial practice the BSP also would like to have an additional “global” view encompassing several capabilities or clients. Examples are the business transactions of a certain client in all capabilities, all business transactions of a certain category, the cases currently processed in a department, etc. In order to provide the global view and to offer integrated local and global optimization, an aggregation and filtering of local view data on a higher level is required. CDD has to be prepared for this purpose, e.g. by reflecting the indicators required for optimization in context elements and different optimization strategies in context sets (see section 2.2). The focus of this paper is on methodical aspects of this CDD extension.

The main contributions of this paper are (a) the analysis of requirements for integrating local and global views, (b) the concept of local-as-global and global-as-local and (c) an example for using the method extension. Section 2 presents an industrial case form BPO motivating the work and summarizes previous work in capability management. Section 3 presents an approach for integrating local and global optimization manifested in a method component for CDD. Section 4 discusses related work and section 5 summarizes initial experiences and gives an outlook.

2 Context-aware Business Services: the CaaS Approach

Work in this paper was performed as a part of the EU-FP7 project “Capability-as-a-Service in Digital Enterprises (CaaS)”. This section introduces an industrial case in BPO (2.1) motivating the work, and the CDD approach developed in CaaS (2.2).

2.1 Business Process Outsourcing of Energy Distribution Companies

Main partner in the industrial case is the company SIV.AG from Rostock (Germany). SIV offers BPO services to a variety of medium-sized utility providers and other market roles of the energy sector in Germany and several European countries. BPO, i.e. the performance of a complete business process for a business function by a BSP outside an organization, has to offer and implement solutions with variations for different cases. One variation is inherent in the business process as such. Even though core processes can be defined and implemented in standard software systems, configurations and adjustments for the organization in question are needed. The second cause of variation is the configuration for the country of use, i.e. the implementation of the actual regulations and bylaws. The third variation is related to the resource use for implementing the actual business process for the customer, i.e. the provision of technical and organizational capabilities.

Basis for these services is SIV’s software product kVASy4. Integrated with the business process environment, kVASy4 provides business logic for the energy sector. Different deployment models are used including a provider-centric model (kVASy4 and the business processes are run at SIV), a client-centric model (kVASy4 is

installed at the client site and the manual work of the business process is performed at SIV) and mixed models (e.g. kVASy4 in the cloud, process performed partly at the client and partly at SIV). SIV envisioned a more dynamic way of providing BPO services to their customers for ad-hoc up-scaling of services for existing customers.

As a basis for the research work presented in this paper, we performed a case study in SIV's BSP business unit. Qualitative case study is an approach to research that facilitates exploration of a phenomenon within its context using a variety of data sources. This ensures that the subject under consideration is not explored from only one perspective, but rather from a variety of perspectives which allows for multiple facets of the phenomenon to be revealed and understood. Within the case study, we used three different perspectives, which at the same time represent sources of data: We observed the activities during business service provision, we examined the business process models used for execution of the outsourcing and we interviewed different roles involved in BPO services. The case study work resulted in a better understanding of the characteristics of local and global context in the case and in requirements to the method extension presented in section 3. After the case study, SIV also formed the basis for evaluating the new method extension (see section 3.4).

2.2 Capability-driven Design and Delivery (CDD)

Business services are IT-based services which digital enterprises provide for their customers. Business services usually serve specified business goals, they are specified in a model-based way and include service level definitions. The CDD approach explicitly defines (a) the potential delivery contexts of a business service, (b) the potential variants of the business service for these delivery contexts and (c) what aspect of the delivery context would require what kind of adaptation. The potential delivery context basically consists of a set of parameters (context elements) which characterize the differences in delivery. The combination of all context elements and their possible ranges defines the context set. The potential variants of the business service are represented by process variants. Since in many delivery contexts it will be impractical to capture all possible variants, we propose to define patterns for the most frequent variants and to combine and instantiate these patterns to create actual solutions. The connection between context elements, patterns and business services has to be captured as transformation or mapping rules. These rules are defined during design time and interpreted during runtime. CDD has been further elaborated by defining a meta-model (see, e.g. [9]), a methodology and a tool environment.

The *CDD methodology* consists of method components. To structure the methodology the method components have been divided into upper-level method components, describing a certain application area with method components relevant for that area. The upper-level method components are the following:

- Getting Started with CDD. Supports decision making whether or not CDD is suitable for an organization, and how to get started with CDD.
- Capability Design Process. Contains an overview on how to design, evaluate and develop capabilities using process models, goal models and other models.
- Enterprise Modelling. The component contains method components that guide the creation of enterprise models that are used as input for capability design.

- Context modelling. Describes the method components needed for analysing the capability context, and the variations needed to deal with variations.
- Reuse of Capability Design. This component contains guidelines for the elicitation and documentation of patterns for capability design.
- Run-time Delivery Adjustment. Describes the components needed to adjust a capability at runtime.

The *CDD tool environment* includes the following functional components:

- capability design tool CDT - provides modeling elements defined in the capability meta-model and models the required capability including business service, business goals, context and relations to delivery patterns.
- Pattern composition module and repository - identifies patterns for capability delivery, composes them and offers storage and maintenance of patterns.
- Context platform - captures and provides data from external data sources including sensing hardware and Internet based services such as social networks.
- Capability delivery navigation application (CNA) - provides means for monitoring and adjustment of capability delivery. It includes a module for monitoring context and goal KPI, predictive evaluation of capability delivery performance and delivery adjustment algorithms.
- Capability delivery application (CDA) is developed following the process and technologies used by a particular company. The CDD methodology only determines interfaces required for the CDA to be able to receive capability delivery adjustment commands from the capability delivery navigation.

3 Local and Global Optimization in Capability Delivery

3.1 Purpose of the Method Extension

This section proposes a method extension for the methodology part of CDD which has the focus on monitoring and optimizing a capability in BPO. The method extension was motivated by and developed and applied in the industrial case presented in section 2.1. In BPO monitoring of capability delivery primarily focuses on two different aspects which might cause capability adjustments:

- the business service provider has to make sure that the capability is provided in accordance to the contractual service-level agreement with the client. Typically, such agreements include maximum processing time for a business transaction, maximum waiting time before a business transaction is processed, capacity to provide for the client in terms of knowledge workers, etc.
- internal policies of the business service provider have to be met. Examples are the way to decide what business unit is responsible for a business transaction and in what order the cases have to be processed (dispatching), the quality assurance measures to be performed for each case, the targeted maximum processing time, average processing time or waiting time until processing.

Both aspects usually are reflected in performance indicators which ideally can be captured automatically in the runtime environment. Furthermore, the context model for a capability often reflects both aspects by including context elements mirroring the

decision strategies. An advantage of the CDD approach is the possibility to differentiate between different levels of monitoring during capability delivery. From a technical perspective, BPO frequently uses different installations of the capability delivery application (e.g. separate installations for clients with high volume of business transactions) which potentially are deployed on different technical platforms (e.g. in the computing center of the client, at the service provider's premises or on cloud resources). Thus, the "local view" of each CDA installation and the "global view" of all local installations will have to be subject of monitoring and optimization. Furthermore, the BPO also is interested in additional views while monitoring, like all business transactions of a certain client, all business transactions of a certain category, the cases currently processed in a department, all cases delayed independently of the installation and service type, etc.

The local view of each CDA installation usually is equipped with an own monitoring application (e.g. a business process activity monitor), but the global view cannot easily be provided as they require an aggregation and filtering of local view data on a higher level. Furthermore, an optimization strategy on global level might require modifications in the decision logic on local level and vice versa. With many local installations a manual intervention on either local or global level is too time consuming, i.e. dynamic adjustments require a technical integration of local and global level. In the CDD approach, CNA offers the possibility to establish a global monitoring and optimization level on top of local CDA installations with the additional option to not only use indicators captured in the runtime environments but also from other context sources, even outside the enterprise IT. However, if the subject of integrated local and global optimization is addressed, capability model and adjustment algorithms have to be prepared for this purpose, e.g. by reflecting the indicators required for optimization in context elements. The method extension proposed is addressing the subject of capability modelling for integrated local and global monitoring and optimization. One of the aims in this context is to minimize the interventions in local systems.

The issue of optimizing a set of systems with potentially different „local" strategies from a global perspective, i.e., for a "global" strategy, has been subject of study in operations research and production systems before. Often, the optimization problem can be expressed in a formal way, e.g., as optimization model, and established optimization approaches can be applied, e.g. linear optimization, mixed integer programming, network-based optimization or non-linear optimization. In this context, the method component aims at supporting problem analysis and development of the optimization model by identifying the required decision variables and restrictions. However, the actual development of the optimization model is not part of this method component, as this topic is already covered in existing literature.

3.2 Global-as-Local or Local-as-Global

Independently of the actual optimization approach, there are different ways to implement the actual optimization in enterprise IT on local level (i.e. in CDA) and global level (i.e. in CNA). In the context of this method component, we consider two different approaches and support decision making on what approach to use. Global-

as-local assumes that the overall optimization requires changes in behavior of the local system which requires information about the global situation. Local-as-global assumes that local systems adapt their behavior using only information about their local situation and the overall optimization is done by coordinating local systems on a global level. Global-as-local typical would be applicable in the following situations:

- Implementation of specific strategies for overall resource usage which depend on actual situation of these resources
- Strategies which require human intervention at central point
- Strategies which require information not available at local systems

Selected situations when local-as-global is adequate are

- Low update frequency of global parameters, often with high latency
- High performance requirements for transactions, i.e. there is no possibility to suspend a process for receiving information from the central unit
- Local status affects global optimization (e.g. rare kind of exception)

Implementation of global-as-local typically would either include that the local system asks the global system for instructions how to continue before proceeding with a business case, which technically is similar to a blocking remote procedure call. If non-blocking behavior is required, the global system would have to update decision variables of the local system whenever there is a change in scheduling strategy. However, this is only adequate if the scheduling is not dependent on the individual case but the overall resource availability. In an implementation of local-as-global the local system typically decides independently but notifies the global system. The global system will have to re-assign resources based on the notification information.

3.3 Procedure of the Method Extension

The method extension consists of several steps, which can be summarized as its procedural part. The steps and their purposes are as follows:

- Step 1 - Scoping: to decide for which capability or capabilities local or global optimization shall be prepared
- Step 2 - Identify and gather relevant information. This includes earlier analysis results, in particular from capability modelling, contractual agreements with the client, internal policy documents and ideas about potential optimization strategies
- Step 3 - Extract desired decision logic: To draft an idea how the desired optimization could be reached and to prepare the identification of decision variables needed for implementing the desired decision logic
- Step 4 - Identify decision variables: To identify the decision variables needed for implementing the desired decision logic
- Step 5 - Decide on local-as-global or global-as-local: To decide whether local-as-global or global-as-local should be implemented
- Step 6 - Implement either local-as-global or global-as-local, i.e. to design the way of implementing the capturing of decision variables and the communication between local system and global level.

3.4 Example

For brevity reasons, the example will only include the first 4 steps which are essential to demonstrate the intention of the method component. In step 4, we will only show the details of one of the example capability variations, variation V2 (see below).

Step 1- Scoping: As an example we will use a capability of the business service provider introduced in section 2.1. This capability includes two major variations V1 and V2. The required services for the two variations are performed by two organization units A and B which have a need to achieve better load balancing between them. The use of the method component aims at (a) identifying the decision variables and restrictions for implementing workload balancing between organization units A and B, and (b) at deciding whether local-as-global or global-as-local has to be implemented for V1 and V2.

Step 2- Information gathering: If the BSP is handling exceptions in the first variation (V1) the knowledge workers from organization unit A are responsible. For variation 2 (V2), organization unit B is taking care of the exceptions. As the exceptions from V2 are related to those of V1 but V2 exceptions are far more complex, the knowledge workers from organization unit B would also be qualified to work on exceptions from V1, but not vice versa. Balancing of work load between the two units currently is not optimal as no overall view on global level is available. If the capability owner has the impression that organization unit A has too much work and the organization unit B has free capacity, work is redistributed from A to B until B starts to complain. For V2, the service-level agreements with the clients are structured similar to V1. For V2 it states that up to 10 exceptions per working day will be handled by the client; each exceptions on top until max. 50 has to be managed by the BSP. In case of more than 50, the exceptions will be handled by the client unless the client changes the upper limit. All exceptions to be handled by the BSP are sent to the task management system of organization unit B. For V2, the BSP receives a compensation based on the number of exceptions handled. The service-level agreement with two of the clients includes a penalty for each exception whose handling time exceeds a certain time limit.

Step 3 - Extract desired decision logic: The desired optimization for V1 is as follows: IF the sum of jobs in the task management queue is higher than what is the capacity of organization unit A for a given time period AND organization unit B still has capacity available THAN the dispatcher on duty asks organization unit B to also work on exceptions of V1 OTHERWISE additional resources have to be called in by the dispatcher for the next shift. For V2 the capability owner puts the priority on avoiding penalties with at the same time optimizing resource use in unit B. The wish of the capability owner is to assign priorities to exceptions, i.e. each exception should get a priority number. The priority number is calculated using the average waiting time and contracted maximum processing time agreed on with the client.

Table 1: Final list of decision variables for V2 on global level

Decision variable	Measurable Property	Captured in the local system?
BSP human resources	Schedule (planned capacity per date and org. unit)	NO

SUM of the backlog sizes of all local systems for the exception leading to V1	Backlog size	YES
Exception type	BAM notification	YES
Maximum processing time	SLA information (for each exception type and client: max. hours)	NO
Average processing time	operations average (for each exception type: average processing time)	NO
Current workload in BSP	Task management data (current number of tasks and average processing time per org. unit)	NO

Table 2: Final list of decision variables in the local system for V2

Decision variable	Measurable Property	Captured in the local system?
Backlog threshold for the exception leading to V2	Backlog size	YES
Exception type	BAM notification	YES
Processing priority	Calculated based on maximum processing time, average processing time and current workload at BSP	NO

Step 4 - Identify decision variables: To identify decision variables includes different activities including a separate view on local and global level and context modeling. The final lists of decision variables for V2 are shown in table 1 (global level) and table 2 (local level).

Step 5 - Decide on local-as-global or global-as-local: As shown in tables 1 and 2, for V2 not all decision variables in the local systems are captured locally. Thus, “global-as-local” has to be implemented. For V1, all decision variables required in the local systems are captured locally. Here, “local-as-global” has to be performed.

4 Related Work

4.1 Capability Management

The term capability is used in different areas of business information systems. In the literature there seems to be an agreement about the characteristics of the capability, still there is no generally acceptance of the term. The definitions mainly put the focus on “combination of resources” [2], “capacity to execute an activity” [1], “perform better than competitors” [4] and “possessed ability [12]”. The capabilities must be enablers of competitive advantage; they should help companies to continuously deliver a certain business value in dynamically changing circumstances [5]. They can

be perceived from different organizational levels and thus utilized for different purposes. According to [6] the firm performance is the greatest, when the enterprises map their capabilities to IT applications. In this perspective the capabilities are provided as business services, i.e. designed and delivered in a process-oriented fashion. Capabilities are directly related to business processes that are affected from the changes in context, such as, regulations, customer preferences and system performance. As companies in rapidly changing environments need to anticipate variations and respond to them [3], the affected processes need to be adjusted quickly.

In the CaaS project capability is defined as *the ability and capacity that enable an enterprise to achieve a business goal in a certain context* [7, 9]. Ability refers to the level of available competence, where competence is understood as talent intelligence and disposition, of a subject or enterprise to accomplish a goal; capacity means availability of resources, e.g. money, time, personnel, tools. This definition focuses on the components of enterprise modelling such as goal modelling and utilizes the notion of context, thus stresses the variations of the standard processes.

4.2 Adaptive Case Management

During the last years, adaptive case management (ACM) has attracted a lot of attention in the field of business process modelling. Whenever business processes are only weakly structured or include a lot of exceptional situations, they cannot be fully specified and automated. Such business processes typically show many situations where a human agent, the so called knowledge worker, has to be involved in completing a business process or has to fully take over this process. In this context, the term ACM-system was defined as follows: "Systems that are able to support decision making and data capture while providing the freedom for knowledge workers to apply their own understanding and subject matter expertise to respond to unique or changing circumstances within the business environment" [10].

Service providers in BPO are interested to perform as many activities in a business process in an automatable way, i.e. to only base BPO on IT resources only. If case handling has to be done by knowledge workers, BPO service providers are interested to reduce the costs for this step because of economic reasons: case work requires personnel with the right knowledge, coordination of this personnel and definition of service levels and quality attributes in relationship to the client. The main focus of ACM is the "case", which has three main features [11]:

- Organization: conventional business process models always include a control flow, which often is missing in ACM. ACM often is data-centred in contrast to the process-centred approach of a process model,
- Case handling: case handling often is performed not only by one individual knowledge worker, but includes the interaction of several workers. This requires a high level of cooperation from each individual participant.
- Adaption: in comparison to business process management which attempts to treat all instances of a process in an identical way, ACM requires adaption of the case handling to each individual case.

5 Summary

New situations in business environments arise due to changes in regulations, bylaws and customer preferences. Capabilities help companies to continuously deliver a certain business value in these dynamically changing circumstances by adjusting the service delivery to different contexts. This paper focuses on a specific aspect of capability management: integration of local and global optimization. A method extension is proposed which focuses on analyzing the optimization context and identifying the required decision variables. The method extension complements the CDD methodology. An example of using the methodology in an industrial case of BPO illustrates the method extension use. Future work has to include application of the method extension in more real-world application in order to facilitate a systematic evaluation and improvement of the method extension.

Acknowledgments. This work has been performed as part of the EU-FP7 funded project no: 611351 CaaS – Capability as a Service in Digital Enterprises.

References

1. Jiang Y, Zhao J An empirical research of the forming process of Firm inter-organizational e-business capability: Based on the supply chain processes. In: 2010 2nd International Conference on Information Science and Engineering (ICISE), pp 2603–2606
2. Antunes G, Barateiro J, Becker C et al. Modeling Contextual Concerns in Enterprise Architecture. In: 2011 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp 3–10
3. Eriksson T (2013) Processes, antecedents and outcomes of dynamic capabilities. *Scandinavian Journal of Management*: -. doi: 10.1016/j.scaman.2013.05.001
4. Boonpattarakon A (2012) Model of Thai Small and Medium Sized Enterprises' Organizational Capabilities. *jmr* 4(3). doi: 10.5296/jmr.v4i3.1557
5. Stirna J, Grabis J, Henkel M., Zdravkovic J. (2012) Capability Driven Development – An Approach to Support Evolving Organizations. *PoEM 2012, LNBIP vol 134*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 117–131
6. Chen J, Tsou H (2012) Performance effects of 5IT6 capability, service process innovation, and the mediating role of customer service. *Journal of Engineering and Technology Management* 29(1): 71–94. doi: 10.1016/j.jengtecman.2011.09.007
7. Zdravkovic, J., Stirna, J., Henkel, M., and Grabis, J. (2013) Modelling Business Capabilities and Context Dependent Delivery by Cloud Service. *CAiSE 2013, LNCS 7908*, pp. 369--383. Springer, Heidelberg
8. Mena T, Bellamine-Ben Saoud N, Ahmed M et al. (2007) Towards a Methodology for Context Sensitive Systems Development. *Modeling and Using Context, LNCS, Vol 4635*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 56–68
9. Bērziša, S.; Bravos, G.; Gonzalez Cardona, T.; Czubayko, U.; España, S.; Grabis, J.; Henkel, M.; Jokste, L.; Kampars, J.; Koç, H.; Kuhr, J.-C.; Llorca, C.; Loucopoulos, P.; Juanes Pascual, R.; Pastor, O.; Sandkuhl, K.; Simic, H.; Stirna, J.; Zdravkovic, J. (2015) Capability Driven Development: An Approach to Designing Digital Enterprises. *Business and Information Systems Engineering (BISE)*. February 2015.
10. Swenson K.D. (2010). *Mastering the unpredictable: How adaptive case management will revolutionize the way that knowledge workers get things do*. Meghan-Kiffer Press, Tampa
11. Herrmann C. and Kurz M. (2011). Adaptive case management. In *S-BPM ONE-Learning by Doing-Doing by Learning*, Springer, 80-97.