Research School for Operations Management and Logistics

Business Process Technology and the Cloud: defining a Business Process Cloud Platform

Vassil Stoitsev, Paul Grefen

Beta Working Paper series 393

BETA publicatie	WP 393 (working paper)
ISBN	
ISSN	
NUR	982
Eindhoven	September 2012

Business Process Technology and the Cloud: defining a Business Process Cloud Platform

Vassil Stoitsev, Paul Grefen

School of Industrial Engineering, Eindhoven University of Technology

Eindhoven, The Netherlands
September, 2012

Abstract

The present state of the integration between business process technology and the Cloud is vague and not well defined. Industry research organizations predict that enterprises will be moving in both these directions in the next few years. This will increase the need for a clear integration between these two areas. Apart from this, many current problems with automated business processes stem from the poor connection between business application systems and the needed business process support, as processes are still hardcoded in application logic. In this paper, we first give an overview of the technologies that deal with business processes and the different flavors of related architectures in the IT world. Furthermore, we present a summary of what the Cloud is and what models revolve around it. We compare the different reference architectures for Cloud Computing offered by industry and academia according to their levels of aggregation and abstraction. We extend the concept of Business-Process-as-a-Service (BPaaS) into the concept of Business Process Cloud Platform (ProCPlat) and propose four scenarios that cover the different types of process definition and process enactment ownership. Finally, we discuss how industry makes choices concerning projects that involve both process technology and Cloud Computing technologies. We believe that our proposal for ProCPlat is a first step towards an easy and manageable platform for business process support in the Cloud.

Contents

1. Introduction

During the past several years, Cloud Computing has been regarded as one of the hottest trends in the IT industry, even though the topic of large scale computing has emerged some fifty years ago. Together with it, business process technology has been going through different stages of development, which are sometimes called waves by the industry research companies. Despite the fact that a lot of money is invested into process technology innovation, there are still many applications running inside corporate datacenters, which encapsulate support for business processes inside their software architectures. This makes the software less agile and difficult to modify. As more software application are ported to the Cloud, the problem just migrates, stays unresolved, and is even magnified as eventual connectivity to on-premise Business Process Management Systems (BPMS) is complicated. We consider this a challenge that needs to be resolved in the near future. With this paper, we present our first steps towards a solution to this problem.

Tackling such a problem requires looking into three main fields on knowledge: process technology, Cloud Computing, and architecture of information systems. We can categorize the first two as the main building blocks and the last one as the connecting element. We consider that taking a single-sided perspective will bring biased results, and for this reason we interconnect pieces of knowledge from the three fields. Doing that, we extend the concept of Business-Process-as-a-Service and propose a logical division based on process technology theory, which is mapped later in this paper to the existing Cloud service stack. The division presents the ownership of a business process from two angles: process definition and process enactment which form four different scenarios for Business Process Management. Some of them are already existent in the Cloud ecosystem, as we summarize shortly the market for process support in the Cloud.

The rest of the paper is organized as follows. The next section discusses the differences between business processes and workflows, and the supporting software systems for the two concepts. In Chapter 3 we provide details about Cloud Computing, trace back its historical development, and explain the different Cloud services which currently exist in the industry. Chapter 4 details the terminology around different software architectures and compares different Cloud Computing reference architectures. In Chapter 5 we provide our definition of Business Process Cloud Platform and explain what scenarios can emerge after categorizing business processes according to the owner of process definition and process enactment. Chapter 6 discusses how realization decisions are taken for Cloud-based projects that require process support. Finally, we provide some general conclusions and shed some light on our future work on the topic.

2. What is a workflow and what are workflow management systems?

During the past decades, process technology has been constantly growing as it was going through different stages of development. It is currently one of the big hypes in the Information Technology industry in general. The purpose of this chapter is to explain the terminology around process technology since it is common for the literature to mix it up. We also make some relations to the architectures of business applications. We do not trace the evolution of process technology, nor redefine the terminology in the field since this is out of the scope of this study.

Business Processes

The most abstract entity in process technology is a business process. It was well-defined over the years in different publications. Many authors base their studies on the definition of Davenport and Short [1], which dates back to 1990 and is a cornerstone for process technology in general. According to them, a business process is "a set of logically related tasks performed to achieve a defined business outcome." The term is defined in similar fashion by the Workflow Management Coalition (WfMC) in 1999 [2]. One comprehensive study puts more stress on obtaining a tangible or intangible product as the business process is executed [3]. For the purpose of this study we are satisfied with the definition of Davenport and Short as it is lightweight and easy to grasp.

Workflows

Next, the term 'workflow' needs to be well-defined. It is common that authors interchange this term with business process. The WfMC defines it in a very good way as "the automation of a business process" [2]. It is rather clear that the term workflow is a lot more technology-oriented compared to the term business process. One study even adds upon the technology factor stating that "a workflow comprises cases, resources, and triggers that relate to a particular process" [3]. Cases, resources and triggers stem from the organization and the functioning of an enterprise, and relate to the implementation of process technology. Another logical proof for the difference between the two terms under discussion is the fact that a business process can be complicated enough so that it involves multiple workflows, which might even be of heterogeneous nature, i.e. being processed by different systems.

Since the focus of this study is on workflows and their support, we first explore the technology supporting workflows. We visualize the order of exploration of the terminology concerning process technology in *Figure 1*. We differentiate between two conceptual and two technical topics as each of the conceptual topics has a counterpart on the other side of the dashed line.

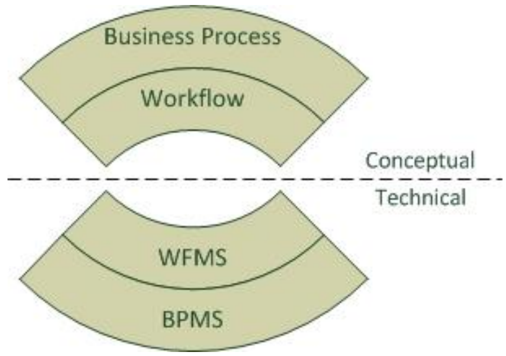

Figure 1: Terminology mapping

Workflow Management Systems

The amount of publications regarding the technology supporting workflows or Workflow Management Systems (WFMS) is quite vast, so this document is not focusing on all of its aspects as this is out of our scope. First, it is wise to mention that workflow management does not require a system in all cases. For years, people have executed workflows using their hands and without relying on a smart technology to help them [4]. Actually a lot of enterprises still rely only on their know-how to manage the execution of workflows manually and WFMSs come to help larger enterprises where the correct management of workflows generates business value. It can be observed that there is a bit of discrepancy on the definition of WFMS. One study defines the terms as a general software package for managing business processes [3]. In this case, the terms workflow and business process are used interchangeably. The WfMC defines WFMS in a more complicated way [2], but it does not mention that WFMS is a general purpose technology. As such it should be able to function properly regardless of the business domain, the type of information flowing and the people involved in the execution of a workflow. The actual WFMS is not performing any work by itself but it is just orchestrating who or which application should process the data and then deliver the result to the next party in the workflow.

In order to classify the types of solutions for workflow management we combine two classifications presented by in two different studies [5], [6]. The components on the left side of *Figure 2* conceptually represent the evolution of applications over the years. The components on the right side represent the categories of workflow management support.

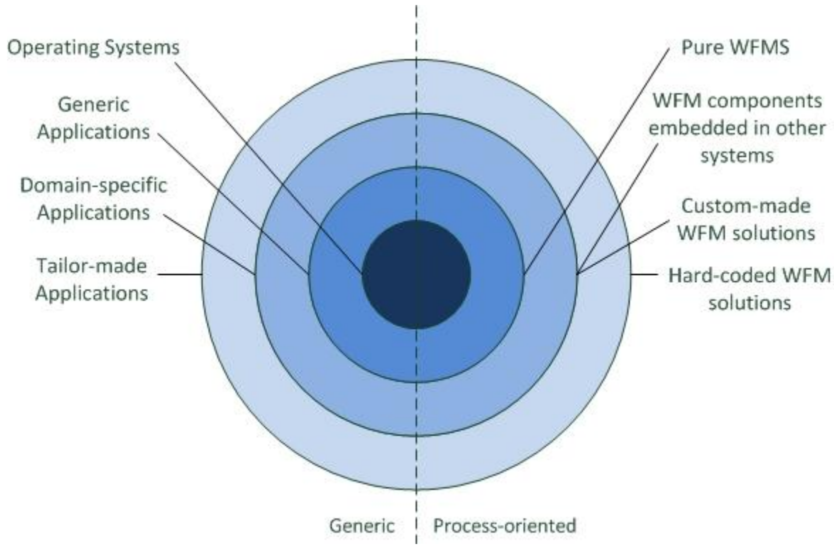

Figure 2: Process technology categorization

In the beginning of the IT industry, only operating systems and tailor-made applications existed, but over the years the applications became more and more generic, so the second and third inner circles emerged. As shown on the right side of the diagram, applications can be very tightly coupled with the workflow management or completely decoupled. For example *Pure WFMS* are *Generic Applications* as they do not have a defined domain to operate in. In the early years of development of process technology, the industry was in the outermost circle, but over the years it is not that simple to determine whether the development was continuously going inward or it was more following a random path. It is logical to think that *WFM embedded components* emerged after *Pure WFMS* were proven to be a success. The conclusion that we can draw is that WFMS emerged as a result of abstracting things out, or in other words using a concrete solution that goes through several phases of generalization in order to create a universal solution. Nevertheless, a good question to investigate at a later stage is if the total generalization of workflow management is the optimal solution to the needs of a business. It is still quite common that there are many hard-coded solutions and the challenge is to find whether there is a balance point between hard-coding and abstraction.

Creating an industry-wide abstraction for WFMS is a task that was handled by the WfMC already in 1995. The coalition came with a reference model which defines what basic features a WFMS should incorporate [7]. In total five additional components connected with interfaces to the workflow engine are proposed as we can see in *Figure 3*.

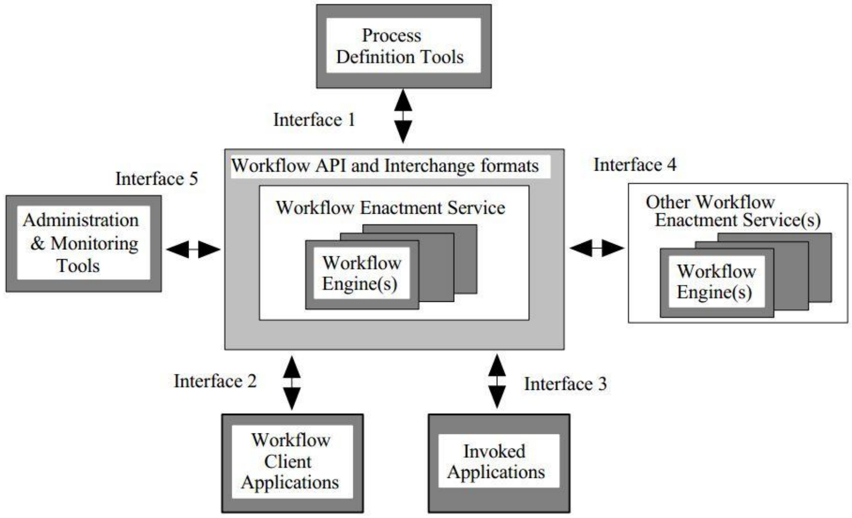

Figure 3. WfMC Reference Model

The WfMC defines a Workflow API and Interchange formats which are important as they define the way the workflow engine is accessed and fed with data. Interesting for our further research is to find out how this reference model best integrates with the architecture of an application. In addition to the WfMC reference model, which is very generic, there are other models as the Mercurius architecture [8], which are more detailed. The Mercurius architecture describes a WFMS in several aggregation levels, i.e., uses functional refinement to provide a more detailed model of a WFMS.

Business Process Management Systems

Exploring the last sector of *Figure 1* brings us to Business Process Management Systems (BPMS). As a higher abstraction layer, BPMS encompass WFMS. Nowadays businesses are talking on a daily basis about BPMS, which are also named Business Process Management Suites. This is a natural evolution as the development of technology is enabling the support of more complicated systems. One study that looks into BPMS in 1995 [9] at which point it was hard to make a distinction between BPMS and WFMS. The author claims that WFMS at this time were “the first Generation of BPMS” and he was right as nowadays the term WFMS is not popular. Another study claims that at a certain point of the development of technology many vendor companies just rebranded their WFMS into BPMS [10]. What BPMS are actually adding on top of the existing technology is more sophisticated monitoring and diagnostic activities. Another goal of BPMS is to enable the improvement of the business activities by integrating heterogeneous application environments. Nowadays, enterprises would like to directly connect to their business partners (those can be part of the same company) and to do this in a seamless fashion. This integration is interesting to be investigated further together with finding the best scenario to achieve it.

3. What is Cloud Computing?

Over the last five years the excitement around "The Cloud" is constantly growing and the concept is currently competing with Social media for the first place in the IT industry hypes. As many business managers see Cloud Computing as something very innovative, the idea about large scale computing is not new at all. During the 1980's mainframe-based computer centers, which provided big amount of computing power at one physical location, were quite trendy too. About half a century ago John McCarthy has suggested during his centennial speech at MIT that one day computation power would be consumed as a basic service like water and electricity. In order to handle "The Cloud" correctly, the providers and consumers of Cloud Computing services must have a good architecture that is underlying their platforms or solutions. Scientists, industry professionals, and standardization organization have all come up with different propositions for references architectures. In this chapter we will discuss some of the already mentioned concepts while refraining from trying to redefine any of them.

Cloud Computing Definition

Many scholars have tried to provide a definition for Cloud Computing and to describe its scope. All [11], [12] and [13], which were all published about three years ago, provide good definitions for Cloud Computing. Nevertheless, in this study we use the definition recently provided by NIST [14] which is also acknowledged and used by the Open Group [15]:

> *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

This definition is generic and does not go either in too many technical details, or in explanations about business relevance. In addition to the definition NIST provides five important characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service), and four deployment models (Private cloud, Community cloud, Public cloud, and Hybrid cloud). Of those, the service models are most interesting to be investigated, thus we explore them in detail in one of the following sections. The two most important characteristics that must be mentioned, as they are widely recognized by all cited studies until now, are elasticity and multi-tenancy. Elasticity refers to the ability of the system to scale out horizontally and provision more systems which are mirrored images of the one already running. This concept is often confused with scaling up vertically – an action that involves the increase of computational power of a single system. The second important characteristic multi-tenancy deals with the pooling of resources in a way that those resources can serve multiple consumers of the Cloud Computing service.

Historical development

The idea about computational power as a common is not new. In 1961 John McCarthy predicts that computation would become a public utility during his centennial speech at the Massachusetts Institute of Technology. According to some newer predictions [12] computing would ultimately become the fifth utility together with water, electricity, gas and telephony, even though telephony will be absorbed by computing, according to us. If we take a look at the present state of computing, the claim does seem realistic.

Looking some years back, there were different trends in the field of Cloud Computing but they all fall under the hood of distributed computing [13]. We get the following definition for it from [16]: “a distributed system consists of multiple processes that must cooperate; hence one must be concerned not just with the behavior of individual components, but also with their joint behavior in the context of the overall application.” Cloud Computing fits well into this definition. It even provides mechanisms that secure the safe operation of multiple processes in one big system. Another type of computing matching the aforementioned definition is Grid computing, which scholars classify as the closest concept to Cloud Computing [12], [13]. There are discussions in the IT industry if Cloud is just a new fancy word for Grid. We agree up to a point with this statement as there are some vital differences between the two types of computing. In its nature Grid computing was trying to solve single tasks that requite vast computational power. The machines participating in a Grid were most of the time powerful servers, managed by separate parties and this is the main difference. Cloud Computing is all about cheap commodity hardware and, currently, a single owner of the computational resources. In addition to this, the concept of services computing is also the source of some ideas in the Cloud. Most of the time the different components of the Cloud Computing platforms can be accesses as services and additionally there is a service orientation between the components themselves inside the platforms. Finally, Cloud Computing can be regarded as an enabler of ubiquitous computing [17], as well as Grid computing is. With its immense power, the Cloud can enable real-time communication of different devices anywhere, anytime. The Cloud Computing services are almost 100% of the time up and ready to scale out depending on what the smart devices all around need.

The emergence of Cloud Computing was not random in any case and has its pure business reasons. The big vendors like Amazon, Microsoft, and Google invested into the Cloud because they saw the business value. In the industry there is always a comparison made between capital expenditures and operational expenditures. This demonstrates how traditional environments require stepwise provisioning of computational power that takes time and financial resources. On the other hand the Cloud can always follow the load of the deployed system by the customer and provide exactly what is needed. This makes the environment efficient and does not waste energy to run unused computational power. Another important business reason for the emergence of the Cloud is that companies do not have any upfront costs in order to start a new business or a new service. This inevitably creates potential users of the Cloud Computing offerings as enterprises are willing to cut on their constantly rising costs for Information Technology. This also means that businesses can focus on their core competences instead of dealing with menial tasks like provisioning servers, and investing money in upgrades of hardware and software.

Everything-as-a-Service (XaaS)

As we already mentioned, some stress should be put on the service models that come together with the concept of Cloud Computing. NIST recognizes three service models [14]: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Those three are also the most cited ones at different technical conferences, but in additional to them there are numerous other models defined by different scholars. The fourth most important one that we are going to investigate in depth later in this study is Business-Process-as-a-Service (BPaaS). Recently it is generating more hype as it is slowly emerging on the market for services. Moreover there are numerous others offering which can be marketed as a service. Commonly they are referred to as Everything-as-a-Service (XaaS) [18]. Other examples of XaaS are Communication-as-a-Service, Monitoring-as-a-Service, Desktop-as-a-Service and even Human-as-a-Service [18].

The most common type of service is definitely Software-as-a-Service. It has been around for about 20 years already with the emergence of the Web in the early 1990's. This is the easiest model for a company to provide its customers with some productivity such as e-mail service or file management, and it is quite common concept. For this reason, it would be more interesting to discuss Platform-as-a-Service. According to a research made by Gartner [19] PaaS will be gaining more power over the years until 2015. Not only big companies will be entering the PaaS market but also small ones that provide developers with innovative platforms to build applications for. Most of the big players in the market already have some kind of offering around PaaS, as Microsoft appears to be the leader in it with almost a complete development platform providing elaborate mechanisms for integration and security. PaaS is also interesting because it enables the companies to focus their resources on building smartly-architected solutions at the level of conceptual software architecture instead of planning the provisioning of servers or the configuration of out-of-the-box online software. People like to be unique and building custom software, which is integrated with different platforms, will always stay in the software business. In the following two chapters we would, first, further investigate the common architectures underlying the Cloud Computing platforms, and then explore Business-Process-as-a-Service in detail.

4. What are the different flavors of architecture in the IT industry and what are the implications for the Cloud?

The concept of architecture has occurred centuries ago when people started imposing some rules in the physical process of building. The reason for this was the need for durability and safety of the buildings. Years later when Information Technology emerged the first software applications were simple, with limited amount of requirements and thus limited amount of functions [20]. As the technology developed systems were getting more complex and some structuring was needed in order to prevent chaos, so the term *architecture* started to be used in the field of software development too. Over time it has been proven in industry that badly-architected software solutions tend to fail. One of the most widely-know examples is the failure of the Ariane 5 space rocket in 1996. Recently the Cloud platform of Microsoft experienced an outage that can be related to a tiny bit of bad architecture too [21]. In this chapter we discuss different flavors of architectures, different reference architectures for Cloud Computing, the differences between Cloud Computing architectures and application architectures, and some common application architectures considerations for the Cloud.

Architecture types

We identify several types of architectures that provide different levels of abstraction (concreteness) and aggregation (granularity). The level of abstraction defines how concrete an architecture model is with respect to the software building blocks. At its highest level of abstraction an architecture model provides no information about types of systems and software vendors, while at the lower level of abstraction even the product versions should be explicitly mentioned. The level of aggregation defines how detailed an architecture model is with respect to the number of components. At the highest level of aggregation a system is only a single black box, while at the lowest level all small subsystems are identified. The first flavor of architecture we call nowadays *concrete architecture* [20] or *solution architecture* [22]. Both terms refer to the underlying structure of a software system that offers a defined set of functionalities [22] which meet the business requirements in a specific situation. Over the years there were other terms for the same type of architecture which are still valid terms as software architecture[1] [23] or capability architecture [24]. When a type of application is often encountered, it is wise to create high-level designs that can be reused in multiple situations concerning this type of application. Such designs are classified as *reference architectures* [20]. In most cases reference architectures are based on accumulated best practices in an industry domain [22]. Still, some reference architectures are designed as a result of research in a new area of knowledge, so some differentiation can be made [25]. Those two different types of reference architectures are classified as Practice Reference Architectures and Futuristic Reference Architectures [25]. Most reference architectures are either generic like CORA [26], domain-specific like NORA [27] or technology-specific like Microsoft .NET [28]. A specialization of reference architecture is *standard architecture* which is specific for a single organization [20]. This type of architecture could be found in large enterprises or governmental institutions [29]. *Figure 4* provides the relationships between the different types of architectures, which can be also seen as traversing the abstraction level of the different architecture types.

[1] In reality, software architecture can be regarded as umbrella term for most of the terminology that we discuss in this chapter.

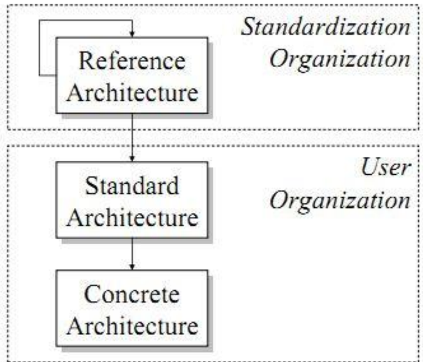

Figure 4: Reference and Standard architectures [20]

An important topic for industrial engineers nowadays is the concept of enterprise architecture. It can be simply defined as the architecture of an enterprise which provides some principles sufficient to meet the requirements of a business [22]. This definition is quite generic and does not show any relation to the technological and business process aspect of an enterprise. For this reason, we consider the following definition [30]:

> *Enterprise architecture: a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure.*

It is important to notice that enterprise architecture provides not only blueprints (or models) but also principles and methods for the development of the architecture which is important when designing for a large enterprise. There are various frameworks which can be used to develop the enterprise architecture of a company. Some of the frameworks like Zachman Framework are more general [31], others like TOGAF [32] and IAF [33] touch only a single aspect of enterprise architecture like the methods or the models. It is common that a method framework like TOGAF is combined with different reference architectures in order to achieve the overall enterprise architecture. In this respect we can categorize enterprise architectures and reference architectures according to their level of aggregation. Enterprise architectures are slowing less details about a business and provide a more generic picture (high aggregation level), while reference architecture focus on a specific aspect of the business (low aggregation level).

Cloud Computing reference architectures

During the last three years both academia and industry have been active in defining reference architectures for Cloud Computing. In order to discuss the different propositions in an easy to understand way, we categorize them again according to their levels of abstraction and aggregation [20]. We plot their relative values and discuss them along the aggregation axis in *Figure 5*:

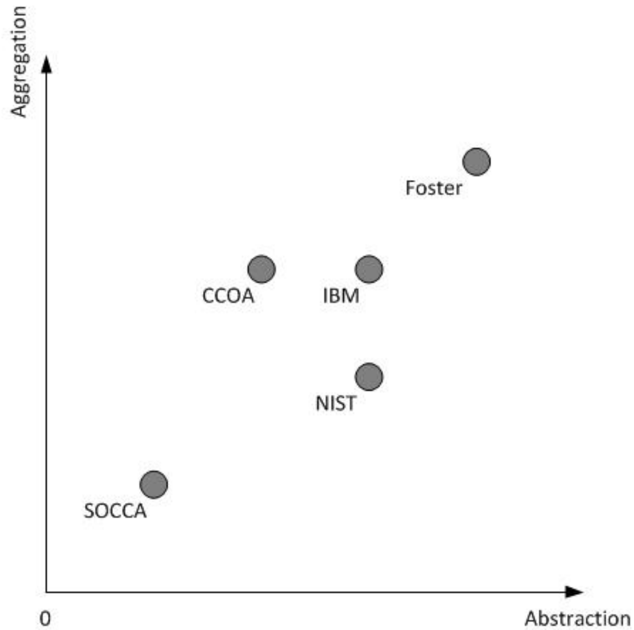

Figure 5: Plotting Cloud Computing Architectures

The simplest model, labeled as *Foster* in *Figure 5*, has high levels of abstraction and aggregation [13]. The authors provide a four-layer architecture without showing any details about the layers. The top three layers of the model can be correlated with the three basic service models that were discusses in the previous chapter - IaaS, PaaS, and SaaS. The fourth layer presented is the actual hardware but there is no service model which directly relates to using hardware. A bit more granular model is provided by IBM [34]. The model is defined in terms of modules which are connected to three main roles: Cloud Service Consumer, Cloud Service Provider, and Cloud Service Creator. The reference architecture model of IBM provides also drilled-down models for the modules in the main model. Thus it is possible to slide the reference architecture along the aggregation axis in *Figure 5* but for our comparison we take into consideration only the base models presented. Another reference architecture developed by IBM is the Cloud Computing Open Architecture (CCOA) [35]. It was developed 2 years earlier than the IBM reference architecture. It has the same level of aggregation but the level of abstraction is lower as the model uses more technology-specific terminology. The architecture is based on modules which are encapsulated in a holistic ecosystem. A downside of the model is the omission of a module that handles security. The authors of the architecture provide a drilled-down model which is even more concrete from technological point of view. Similar to the first reference architecture of IBM is the one defined by NIST [36]. It is a vendor-neutral architecture that is consistent with the NIST definition for Cloud Computing [11]. According to NIST, the architecture is designed in a way that it does not impair the innovation in Cloud Computing at the technology level [36]. This architecture is also role-based but this time there are five different roles identified: consumer, provider, auditor, broker, and carrier. The background colors used for the components of the model make it ambiguous as they do not fully correspond to what is described in the specification document. The most low-level architecture in terms of aggregation and abstraction is the Service-Oriented Cloud Computing Architecture (SOCCA) [37]. It is a layered architecture which is used to aggregate services from different Cloud Computing providers using ontologies. The authors of the architecture have the opinion that at the current state of Cloud Computing services there are no Service Levels Agreements or support of multi-tenancy. Even at the time of publication some of the vendors were providing such kind of features. In addition to those generic reference architectures for Cloud Computing there are publications concerning architectures that have a more specific purpose like federation of Cloud Computing services [38], building applications for the .NET framework [39], or achieving market equilibrium through regulation supply and demand of Cloud Computing services [12].

Cloud Computing architectures versus Cloud Computing application architectures

Before starting our discussion of the application architectures that are used in Cloud environments we should state the difference between “Cloud Computing architectures” and “Cloud Computing application architectures.” The first term refers to the conceptual, high-level organization of a Cloud Computing platform without taking into consideration the underlying technologies. Some significant research has been conducted in this area and different reference architectures have been defined as described in the previous section. The second term under discussion refers to the structure of an application that is running in a PaaS environment. The limitation to PaaS comes from the fact that the IaaS model provides empty virtual machines which can be configured in a way that mimics a traditional on-premise computing environment, so there is nothing new and specific in architecting those applications. Furthermore, if we look into SaaS, it is possible to design something similar to an application architecture but there are limitations. There exist cases when the SaaS solutions are mash-up applications that integrate services from different providers, so some kind of architecting can be done. Nevertheless it is limited to the SaaS solution and the level of architectural creativity that it allows.

Application architectures considerations for Cloud solutions

Even though academia has defined a number of reference architectures for Cloud Computing, almost all the information around application architectures for the Cloud comes from industry [40]. This is logical as the actual vendors define application architectures and best practices in the field. When an application architecture for a Cloud solution is designed, there are three specific aspects that we find important after conducting our research: scalability, minimization of mutable state, and reliability. All of the three aspects are related to the non-functional system qualities. This is due to the fact that the PaaS offerings do not impose restrictions on the functional qualities of a deployed solution different from the non-Cloud environments. Traditional applications that run inside corporate datacenters are not always suitable to be migrated into the Cloud. As PaaS services are based on a fixed programming model, custom-built applications would be more suited for an IaaS environment unless they were initially built for the same programming model. For example applications using the Microsoft .NET framework are easily migratable to the Cloud, as the PaaS platform of Microsoft is based on the same framework.

The most often mentioned feature of the Cloud is undoubtedly its property to easily scale out. In order for this to happen in the most rational way, the architecture must be designed in a way so that the coupling is as loose as possible [41]. This means that each single piece of functionality can be scaled out individually. Having separate components in the different layers of an application architecture implies that those components must communicate asynchronously between each other in order to achieve independence from the rest of the components [42]. The asynchronous communication must be guaranteed by a durable information exchange mechanism such as message queues. The queues are fed with information from one component which is later consumed by another component depending on its availability. In this way the components do not have to wait for the completion of a task by another component. At the current moment most of the PaaS vendors offer queuing functionality out of the box. Decoupling of traditional applications also enables scenarios where hybrid Cloud solution can be utilized, where some of the functionality is still kept in the on-premise datacenter [41]. In those scenarios a Cloud service bus is helpful for a secure message interexchange. Another important factor for the proper scalability is the implementation of elasticity mechanisms which can be classified in three groups: cycling scaling, event-based scaling, and auto-scaling [41]. The last method is the most interesting one and it can be achieved by implementing business rules that are provided with diagnostics data from the application in order to make a scalability decision. Finally, scalability is also an important property for the Cloud data stores. Relational databases are proven to scale out well. Nevertheless, technologies like key-value stores and document stores also scale out quite well horizontally. At the same time those technologies are a lot cheaper compared to traditional relational databases. Key-value stores do not have fixed schemas as relational databases and do not guarantee full ACID (atomicity, consistency, isolation, durability). Instead they provide BASE (basically available, soft state, eventually consistent) [43]. In a lot of scenarios relational databases are not really needed, so the optimal solution is to find the balance between relational databases and key-value stores.

Minimization of mutable state can be regarded as an implementation guideline that reflects on an application architecture. It deals with the state of variables at runtime and the possible changes that might occur with them [42]. In a practical scenario it is hard to make all variables immutable but minimizing their number is beneficial for a Cloud application. It is common that applications are hosted on a number of computing instances, each of which will fail at some point in time [41]. Keeping mutable information within such an instance leads to loss of information upon failure. In order to prevent this from happening distributed caching mechanisms are used to store the mutable information. In a Cloud Web scenario, as shown in *Figure 6*, the distributed cache is positioned between the web servers and the different data stores. This allows the user session data to be stored outside of the dynamic memory of a *Web Instance*, in a distributed cache that is failure-safe. Again, the vendors of Cloud services provide those caching mechanisms out of the box, so that they can be directly utilized by the users.

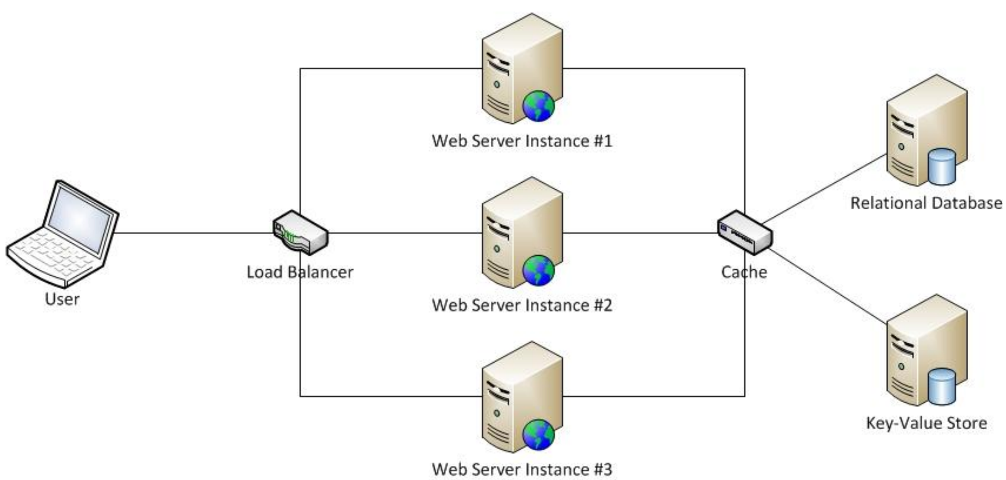

Figure 6: Distributed Caching mechanism

As already mentioned computing instances running Cloud applications will fail at certain point in time [41]. It is the responsibility of the application architect to create a design that is pessimistic enough, i.e. there must be an automatic recovery mechanism incorporated in the application. Even though Cloud vendors provide SLA's, an outage is always possible since the Cloud is relatively new technology. Some of the major vendors have had problems over time that resulted in breach of the SLA's. In the first section of this chapter we mentioned that some authors propose market-based architectures for the Cloud where different vendors can compensate each other when there is massive failure with one of them. At the current moment this is not possible because the Cloud platforms are very different in their programming models but in the future more standards should be expected and the Cloud will become a real everyday utility. Thus, we can conclude that at the current stage Cloud applications must be designed in a way that there is a backup scenario if the datacenter of the Cloud provider fails. This is the reason the reason why mission-critical systems are still not migrated to the Cloud.

5. What are the definition and the characteristics of a Business Process Cloud Platform (ProCPlat)?

As we already mentioned the Cloud has become one of big hypes in the IT industry recently. If we complement this hype with more buzz words as process outsourcing and process orientation, we will reach a more specific aspect of the Cloud which is often named Business-Process-as-a-Service (BPaaS). BPaaS is widely regarded [34], [44] as the fourth service model on top of the three models proposed by NIST [14]. It is still underdeveloped and there is a lack of clear definition what exactly it encompasses. In this chapter we extend on the current concept of BPaaS presented by the industry and academia, and formulate a new definition for a Business Process Cloud Platform (ProCPlat).

Defining ProCPlat

Even though a lot of research is currently focused on the Cloud, the business process area inside it is still vaguely defined. One of the reasons for this is its industrial immaturity. We must make it clear from the beginning that BPaaS, which is marketed as the service model dealing with business processes, is not just another fancy word for SaaS that deals with processes, even though BPaaS tends to be considered as a special form of SaaS in some cases [45]. For example, typical CRM solutions in the Cloud, which involve some business processes, would continue to be part of SaaS, while BPaaS would give more flexibility and agility to the end user in configuring predefined processes that are executed in the Cloud. Later in this chapter we describe how business processes relate also to SaaS, and even to PaaS, as our platform covers a wider spectrum of scenarios involving business processes.   Currently the best definition of BPaaS is provided by IBM as they merge information regarding the Cloud market presented by two of the industrial research companies – Gartner and IDC [34]:

> *Business process services are any business process (horizontal or vertical) delivered through the Cloud service model (Multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via Web-centric interfaces and exploiting Web-oriented cloud architecture.*

This definition is not further elaborated upon by IBM and it does not explicitly specify whether the processes can be custom-defined, even though that can be implicitly concluded from placing BPaaS on the top of the Cloud services stack. In addition, the definition does not include some characteristics that we consider important for running business processes in the Cloud. Those characteristics can be found in different publications and at the end of this section we provide a new definition for a Business Process Cloud Platform which includes some of the characteristics. The Open Group provides an explanation of BPaaS [15] which extends the NIST definition of Cloud Computing [14] and gives an important addition to the definition of IBM – the processes running in the Cloud can be partiality supported by people. In addition, a lot of processes which are a subject of Business Process Outsourcing (BPO) like payroll, HR, CRM, etc. are mentioned. They form the vertical implementation orientation and one half of the processes that can be executed in the Cloud according to us. The functionality in this half would be categorized as out-of-the-box functionality as enterprises prefer to get the whole stack of processes related to non-core business activities. The other half, which is a subject of the Business Operations Platform concept [44], is connected to the creation of custom processes defined by the application developers [40]. This could be regarded as the horizontal implementation orientation which in other words is BPMS functionality in the Cloud. The functionality of this half would be then both out-of-the-box and custom as this implementation orientation provides greater flexibility. *Figure 7* provides a graphical representation of the distinction between vertical and horizontal implementation orientation:

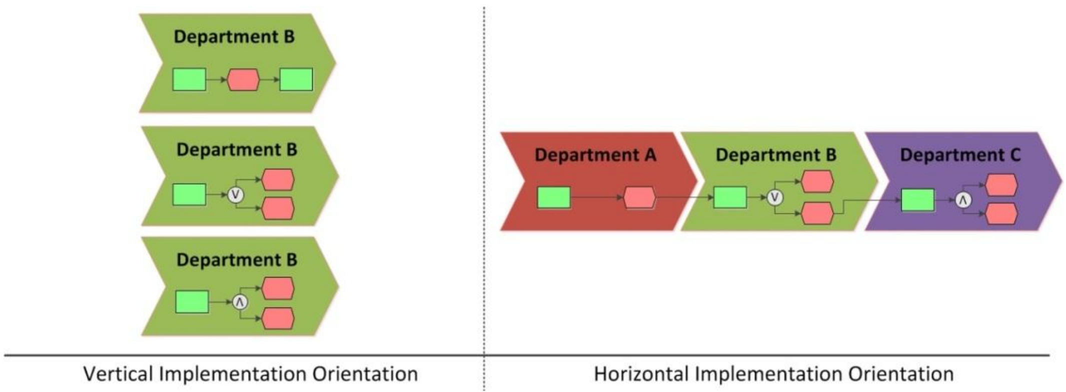

Figure 7: Vertical vs. Horizontal Orientation

Different authors also propose a form of syndication2 of processes and their brokerage inside a BPaaS platform [40], [46], [47]. It is also claimed that the current Cloud service stack is rather restrictive which makes the development of process-centric applications complicated [40]. Thus, the syndication could be even elevated to the level of dynamic creation of business processes, similar to the concept of instant virtual enterprises (IVE) [48] and involving the concept of Universal Description, Discovery and Integration (UDDI), which is part of the Web Services protocol stack [49]. Based on those additional characteristics we form our definition of ProCPlat:

> *Business Processes Cloud Platform (ProCPlat) is a platform composed of Cloud services that enables the definition, enactment, and syndication of business processes (fully automated or partially supported by people) in both horizontal and vertical fashion via Web-centric or programming interfaces.*

This definition uses as a basis the IBM interpretation of BPaaS [34] but it adds some properties that we regard as important. We consider the syndication (and at a more generic level the integration) of custom processes a factor that will drive the emergence of ProCPlat in the future together with the partial support of processes by people.

Reasons for emergence

After extending on the BPaaS concept we provide some reasons for the forecasted emergence of this additional Cloud Computing platform as we see them at the current moment. First, the most obvious reason is financial and it coincides with the reasons for the emergence of the Cloud in general – economy of scale. Additionally, implementing an in-house BPMS solution is expensive not only because of the initial charges for consultancy, but also because of the licensing fees. In this case a reasonable question to

[2] Syndication can be regarded as a more specialized form of integration

be asked would be if the BPMS companies would agree to lose part of their revenue as the Cloud uses the pay-per-use principle. The answer is that ProCPlat would allow more companies to use process-oriented technology which might actually increase the profit of the vendors at a later stage. The second reason for emergence that we identify is connected to agility. Currently, most of the SaaS solutions are canned, so only cosmetic configuration is possible [44]. ProCPlat would enable the customization of the process-aware systems and syndication of processes with other business partners which would give ProCPlat an advantage in Business-to-Business scenarios.

Ownership models in ProCPlat

Another important characteristic that requires discussion is the governance model around ProCPlat – who is the owner of the process definitions, who is allowed to enact them, and who is actually executing them. Currently, most of the business processes run inside corporate datacenters and the interaction with external parties is achieved through complicated integration schemas. Bringing the processes or parts of them into the Cloud enables the emergence of new ownership scenarios and we identify four of them, adding on top of some already identified [45]. The four scenarios are mapped in *Figure 8* according to the multiplicity of the process definition owner and the multiplicity of the process enactment owner. We consider that in all scenarios the party responsible for the execution of the process is the ProCPlat vendor.

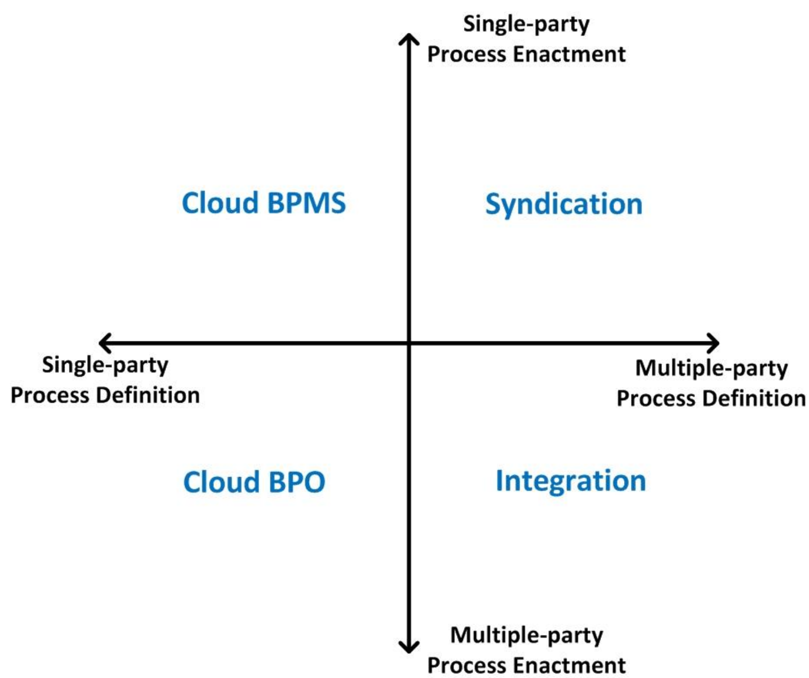

Figure 8: Mapping ownership scenarios

In the *Cloud BPO* scenario the owner of the process definition is the actual provider of the ProCPlat or a third party which provides a process definition as a product. In this case the customer is adding only configuration changes to the process. This scenario resembles a traditional BPO. In the *Cloud BPMS* scenario the sole owner of the process is the consumer of the ProCPlat. This is similar to porting a process from a traditional BPMS solution into the Cloud. In the third scenario the customer is creating a *syndication* of two or more processes coming from different providers. There exist an intermediate process definition defined by the customer of the platform, but still the actual sub-processes have a different owner. In the fourth and most interesting scenario we have a joint ownership by two or more consumers of the ProCPlat. This case provides a seamless *integration* of processes of two enterprises which is important for Business-to-Business (B2B) relationships. Additionally, the case relates to the concept of business network process (BNP) [50] which help for the realization of collaborative processes in B2B environments. The *integration* quadrant of *Figure 8* should be sufficient to cover an architecture supporting BNP's [50]. We consider the third, and the fourth scenarios as two of the drivers for the future development of ProCPlat.

Additional considerations

In order to conclude with the specifics of ProCPlat we need to mention several additional factors that might bring some concerns to the market. In general the business process area in the Cloud should bring more innovation and be one of the significant areas for development in the future. The proposed ProCPlat is expected to bring wider reach for BPMS technology, especially in SMB's. Also on the other side of the Cloud boundary, the ProCPlat might create a more fragmented market for processes as some new specialized companies start selling their know-how for specific processes. The ProCPlat should help resolving the vendor lock-in [40] to which Cloud customers are bound, since moving solutions between different platforms is hard to be achieved. This is due to the differences in the programming models of the Cloud vendors. ProCPlat would enable a dynamic collaboration between parties with smooth integration. Nevertheless all this might come with its price – companies being unwilling to put their processes in the Cloud, which in most cases represent the businesses' core competences. This would create a good amount of work for the legal departments as sensitive information would go beyond the corporate firewall. An additional concern would be the ownership of the business process data at any given moment in time. Still, the ProCPlat providers are expected to be flexible enough in order to fulfill the expectations of the market, so we consider that this issue would find its reasonable solution.

Industry state

At the current moment no one of the vendors labels its service offerings officially as BPaaS, nor is a platform similar to ProCPlat proposed. Still there are services that match up to a point part of our definition for ProCPlat. The closest match comes in the BPaaS field from IBM with IBM Global Expense Reporting Solution [51] which provides out-of-the-box configurable processes that deal with handling expenses. Additionally, IBM offers BlueWorks Live for Cloud process modeling. Three of the other big vendors that have some offerings are Cordys, Pega, and SAP. Cordys provides a service called Process Factory which touches on the topic of syndication which we identified already as important feature of ProCPlat. Pega and SAP market their solutions as Cloud-based but it is harder to specifically relate them to BPaaS as they seem more similar to BPMS in the Cloud, which is common for the SaaS service model. There are also two consultancy companies which are mentioning BPaaS in their service offerings – BeyondCore and Genpact. As a whole the market for business processes in the Cloud is not yet matured, so making a deep analysis at this stage is difficult.

6. When do we make technology choices when designing process-aware Cloud systems?

As we already mentioned in one of the previous chapters, we can categorize architectural descriptions according to their levels of abstraction and aggregation. To also cover the requirements to solution spectrum, we add a third dimension called the realization dimension [20]. The architectural descriptions along this dimension can be categorized in the range of very business-orientated to very technology-oriented. When combined with the other two dimensions – abstraction and aggregation, we end up with the three-dimensional design space presented in *Figure 9:*

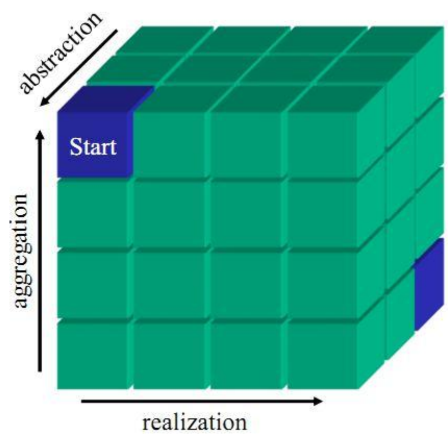

Figure 9: Three-dimensional design space [20]

Each IT project involving the design of information systems is expected to somehow traverse the cube of *Figure 9* from the start to the end cell (highlighted in the diagram). During this traversal, different architectural blueprints are created in the form of models. At the start of the process we have a highly business-oriented model which is abstract and has high level of aggregation, while at the end we have a well-defined IT-oriented model with all of its specifics around software versions and sub-systems.

In order to reach the end, we need to make at some point in time an architecture embodiment decision which is concerned with the choice of the specific technology for the information system that is designed [20]. This is typically done in the second half of the realization dimension as usually business requirements are not technology-dependent. If we take into considerations projects that are process-aware and are to be deployed in a Cloud environment, we can observe some additions to the normal embodiment scenario. As we expect that BPaaS platforms would offer support for standardized process modeling languages as BPMN [52], we can postpone the embodiment decision that concerns the *Abstraction* axis until the end as the Cloud process-support technology would be able to deal with our models. The same line of reasoning could be used for the *Aggregation* dimension as the granularity of the process definitions does not impair the choice of technology. For those two reasons, we can claim that it is possible to postpone the embodiment decision concerning processes in the Cloud until a late point on the timeline of a project. We represent this graphically in *Figure 10*, where the place of the embodiment decision is colored in red.

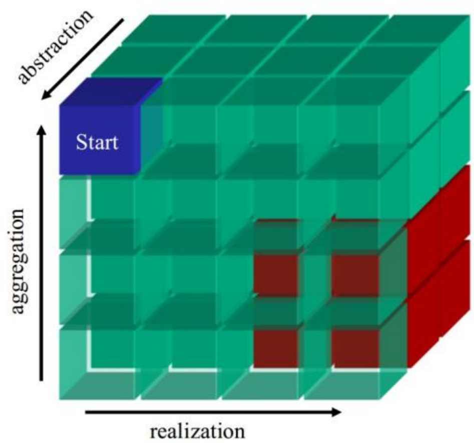

Figure 10: Place of embodiment decision in the design space [20]

7. Conclusions and Future Work

As we have discussed in the introduction, Cloud and process technology are two of the “hot topics" in the IT industry. However, a strong connection between the two concepts is still not present, even though this would enable new scenarios for collaboration between enterprises. Thus, our goal is to find a solution for this problem.

We explore the concepts of process technology, Cloud Computing, and architecture of information systems as a starting point of our research. We make different classifications, which, we believe, help for understanding the fields in an easy manner. Unfortunately, the amount of literature that discusses process support in the Cloud is limited. Additionally, the vendors still do not label their service offerings as BPaaS as the concept is not yet standardized. This leads us to extending the concept of Business-Process-as-a-Service and proposing the Business Process Cloud Platform (ProCPlat). We propose four scenarios for ownership in process technology, and elaborate on them. We also make a short overview of what products are already existent on the market that can be connected with process support in the Cloud in some way. Finally, we analyze how decisions are taken when it comes to introducing process support to Cloud systems. This is important from managerial point of view, as it is typically that non-IT people decide how money will be invested in IT.

This working paper is a first step towards the definition of a Cloud platform handling different flavors of business processes and its relation to Cloud Computing application architectures. In our future research, we intend to clearly position ProCPlat in the Cloud ecosystem and propose different models at Business, Organizational and Architectural levels.

References

[1] T. Davenport and J. Short, "The New Industrial Engineering: Information Technology And Business Process Redesign," *Sloan Management Review*, vol. 31, no. 4, pp. 11-27, Summer 1990.

[2] Workflow Management Coalition, "Terminology & Glossary," Specification WFMC-TC-1011, 1999.

[3] W.M.P. van der Aalst and K.M. van Hee, *Workflow management : models, methods, and systems*.: MIT Press, 2002.

[4] H.A. Reijers, *Design and control of workflow processes : business process management for the service industry*.: Springer, 2003.

[5] W.M.P. van der Aalst, "Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management," *Lecture Notes in Computer Science*, vol. 3098, pp. 1-65, 2004.

[6] W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, "Business Process Management: A Survey," *Lecture Notes in Computer Science*, vol. 2678, pp. 1-12, 2003.

[7] Workflow Management Coalition, "The Workflow Reference Model," Specification TC00-1003, 1995.

[8] P. Grefen and R. Remmerts de Vries, "A Reference Architecture for Workflow Management Systems," *Data & Knowledge Engineering*, vol. 27, no. 1, pp. 31-57, 1998.

[9] D. Karagiannis, "BPMS: Business Process Management Systems," *ACM SIGOIS Bulletin*, vol. 16, no. 1, pp. 10-13, 1995.

[10] H.A. Reijers, "Implementing BPM systems: the role of process orientation," *Business Process Management Journal*, vol. 12, no. 4, pp. 389-409, 2006.

[11] University of California at Berkeley, "Above the Clouds: A Berkeley View of Cloud Computing," Department of Electrical Engineering and Computer Sciences, UCB/EECS-2009-28, 2009.

[12] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for," *Future Generation Computer*

Systems, vol. 25, pp. 599-616, 2009.

[13] I. Foster, "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop*, 2008, pp. 1-10.

[14] National Institute of Standards and Technology, "The NIST definition of Cloud Computing," National Institute of Standards and Technology, Special Publication 2011.

[15] The Open Group, *Cloud Computing for Business*.: Van Haren Publishing, 2011.

[16] K.P. Birman, "The Process Group Approach to Reliable Distributed Computing," Cornell University, Special Technical report 1991.

[17] M. Weiser, "Some computer science issues in ubiquitous computing," *Communications of the ACM*, vol. 36, no. 7, pp. 75-84, 1993.

[18] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape," in *CLOUD '09 Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 2009, pp. 23-31.

[19] Gartner, "PaaS Road Map: A Continent Emerging," Gartner, 2011.

[20] P. Grefen. (2012, Summer) Business Information System Architecture. Document.

[21] Microsoft Corporation. (2012, March) Summary of Windows Azure Service Disruption on Feb 29th, 2012. [Online]. http://blogs.msdn.com/b/windowsazure/archive/2012/03/09/summary-of-windows-azure-service-disruption-on-feb-29th-2012.aspx

[22] D. Greefhorst and E. Proper, *Architecture Principles: The Cornerstones of Enterprise Architecture*.: Springer, 2011.

[23] P. Kruchten, "Architectural Blueprints - The "4+1" View Model of Software Architecture," *IEEE Software*, vol. 12, no. 6, pp. 42-50, 1995.

[24] The Open Group. (2009) TOGAF 9.1: Definitions. [Online]. http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_27

[25] S. Angelov, J. Trienekens, and P. Grefen, "Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case," in *2nd European Conference on Software Architecture*, 2008, pp. 225-240.

[26] T. Elzinga, J. van der Vlies, and L. Smiers, *The CORA Model*.: Sdu Uitgevers, 2010.

[27] ICT Uitvoeringsorganisatie. (2012) E-Overheid. [Online]. http://www.e-overheid.nl/onderwerpen/e-overheid/architectuur/nora-familie/nora

[28] Microsoft Corporation. (2009) Microsoft Application Architecture Guide. Document.

[29] Het Waterschapshuis. (2012) E-Overheid. [Online]. http://www.e-overheid.nl/onderwerpen/e-overheid/architectuur/nora-familie/wilma

[30] M. Lankhorst, *Enterprise Architecture at Work*, 2nd ed.: Springer, 2009.

[31] Zachman International. (2008) The Zachman Framework. [Online]. http://www.zachman.com/about-the-zachman-framework

[32] The Open Group. (2012) TOGAF. [Online]. http://www.opengroup.org/togaf/

[33] J. van't Wout, M. Waage, H. Hartman, M. Stahlecker, and A. Hofman, *The Integrated Architecture Framework Explained*.: Springer, 2010.

[34] IBM. (2011, February) IBM Cloud Computing Reference Architecture 2.0. Document.

[35] L. Zhang and Q. Zhou, "CCOA: Cloud Computing Open Architecture," in *2009 IEEE International Conference on Web Services*, 2009, pp. 607-616.

[36] National Institute of Standards and Technology, "NIST Cloud Computing Reference Architecture," Special Publication 2011.

[37] W. Tsai, X. Sun, and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," in *2010 Seventh International Conference on Information Technology*, 2010, pp. 684 - 689.

[38] B. Rochwerger and D. Breitgand, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 535-545, July 2009.

[39] C Vecchiola, X. Chu, and R. Buyya, "Aneka: A Software Platform for.NET-based Cloud Computing," *CoRR*, July 2009.

[40] M. Papazoglou and W. van den Heuvel, "Blueprinting the Cloud," *IEEE Internet Computing*, vol. 15, no. 6, pp. 74-79, Nov.-Dec. 2011.

[41] Amazon Web Services, "Architecting for the Cloud: Best Practices," Whitepaper 2011.

[42] P. Bell, "Architecting applications for the cloud," IBM Corporation, Whitepaper 2011.

[43] D. Pritchett, "BASE - An ACID Alternative," *ACQ Queue*, pp. 48-55, May/June 2008.

[44] P. Fingar. (2010, April) Enterprise as a Service (EaaS) – That's where BPM. Document.

[45] R. Accorsi, "Business Process as a Service: Chances for Remote Auditing," in *Workshop Proceedings of the 35th Annual IEEE International Computer Software and Applications Conference*, 2011, pp. 398-403.

[46] A. Norta, "A Hub Architecture for Service Ecosystems: Towards Business-to-Business Automation with an Ontology-Enabled Collaboration Platform," in *Proceedings of the 6. International Conference on Web Information Systems and Technology (WEBIST) 2010*, 2010, pp. 240-243.

[47] S. Grivas, T. Kumar, and H. Wache, "Cloud Broker: Bringing Intelligence into the Cloud," in *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 544--545.

[48] P. Grefen, R. Eshuis, N. Mehandjiev, and G. Weichhart, "Internet-Based Support for Process-Oriented Instant Virtual Enterprises," *IEEE Internet Computing*, vol. 13, no. 6, pp. 65-73, 2009.

[49] OASIS. OASIS UDDI Specification. [Online]. https://www.oasis-open.org/committees/uddi-spec/faq.php

[50] P. Grefen, N. Mehandjiev, G. Kouvas, G. Weichhart, and R. Eshuis, "Dynamic business network process management in instant virtual enterprises," *Computers in Industry*, vol. 60, no. 2, pp. 86-103, 2009.

[51] IBM. (2011) IBM Global Expense Reporting Solution (GERS). Document.

[52] Object Management Group, "Business Process Model and Notation (BPMN)," Specification 2011.

[53] S. Rajan. (2011, November) BpaaS Interoperability as a Service. [Online]. http://srinivasansundararajan.sys-con.com/node/2045641

Working Papers Beta 2009 - 2012

nr.	Year	Title	Author(s)
393	2012	Business Process Technology and the Cloud: Defining a Business Process Cloud Platform	Vassil Stoitsev, Paul Grefen
392	2012	Vehicle Routing with Soft Time Windows and Stochastic Travel Times: A Column Generation And Branch-and-Price Solution Approach	D. Tas, M. Gendreau, N. Dellaert, T. van Woensel, A.G. de Kok
391	2012	Improve OR-Schedule to Reduce Number of Required Beds	J.T. v. Essen, J.M. Bosch, E.W. Hans, M. v. Houdenhoven, J.L. Hurink
390	2012	How does development lead time affect performance over the ramp-up lifecycle? Evidence from the consumer electronics industry	Andres Pufall, Jan C. Fransoo, Ad de Jong
389	2012	The Impact of Product Complexity on Ramp-Up Performance	Andreas Pufall, Jan C. Fransoo, Ad de Jong, Ton de Kok
388	2012	Co-location synergies: specialized versus diverse logistics concentration areas	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v. Donselaar, Jan C. Fransoo
387	2012	Proximity matters: Synergies through co-location of logistics establishments	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v.Donselaar, Jan C. Fransoo
386	2012	Spatial concentration and location dynamics in logistics:the case of a Dutch province	Frank P. v.d.Heuvel, Peter W.de Langen, Karel H.v. Donselaar, Jan C. Fransoo
385	2012	FNet: An Index for Advanced Business Process Querying	Zhiqiang Yan, Remco Dijkman, Paul Grefen
384	2012	Defining Various Pathway Terms	W.R. Dalinghaus, P.M.E. Van Gorp
383	2012	The Service Dominant Strategy Canvas: Defining and Visualizing a Service Dominant Strategy through the Traditional Strategic Lens	Egon Lüftenegger, Paul Grefen, Caren Weisleder
382	2012	A Stochastic Variable Size Bin Packing Problem With Time Constraints	Stefano Fazi, Tom van Woensel, Jan C. Fransoo
381	2012	Coordination and Analysis of Barge Container Hinterland Networks	K. Sharypova, T. van Woensel, J.C. Fransoo

380	2012	Proximity matters: Synergies through co-location of logistics establishments	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
379	2012	A literature review in process harmonization: a conceptual framework	Heidi Romero, Remco Dijkman, Paul Grefen, Arjan van Weele
378	2012	A Generic Material Flow Control Model for Two Different Industries	S.W.A. Haneya, J.M.J. Schutten, P.C. Schuur, W.H.M. Zijm
377	2012	Dynamic demand fulfillment in spare parts networks with multiple customer classes	H.G.H. Tiemessen, M. Fleischmann, G.J. van Houtum, J.A.E.E. van Nunen, E. Pratsini
376	2012	Improving the performance of sorter systems by scheduling inbound containers	K. Fikse, S.W.A. Haneyah, J.M.J. Schutten
375	2012	Strategies for dynamic appointment making by container terminals	Albert Douma, Martijn Mes
374	2012	MyPHRMachines: Lifelong Personal Health Records in the Cloud	Pieter van Gorp, Marco Comuzzi
373	2012	Service differentiation in spare parts supply through dedicated stocks	E.M. Alvarez, M.C. van der Heijden, W.H.M. Zijm
372	2012	Spare parts inventory pooling: how to share the benefits	Frank Karsten, Rob Basten
371	2012	Condition based spare parts supply	X.Lin, R.J.I. Basten, A.A. Kranenburg, G.J. van Houtum
370	2012	Using Simulation to Assess the Opportunities of Dynamic Waste Collection	Martijn Mes
369	2012	Aggregate overhaul and supply chain planning for rotables	J. Arts, S.D. Flapper, K. Vernooij
368	2012	Operating Room Rescheduling	J.T. van Essen, J.L. Hurink, W. Hartholt, B.J. van den Akker
367	2011	Switching Transport Modes to Meet Voluntary Carbon Emission Targets	Kristel M.R. Hoen, Tarkan Tan, Jan C. Fransoo, Geert-Jan van Houtum
366	2011	On two-echelon inventory systems with Poisson demand and lost sales	Elisa Alvarez, Matthieu van der Heijden

365	2011	Minimizing the Waiting Time for Emergency Surgery	J.T. van Essen, E.W. Hans, J.L. Hurink, A. Oversberg
364	2011	Vehicle Routing Problem with Stochastic Travel Times Including Soft Time Windows and Service Costs	Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok
363	2011	A New Approximate Evaluation Method for Two-Echelon Inventory Systems with Emergency Shipments	Erhun Özkan, Geert-Jan van Houtum, Yasemin Serin
362	2011	Approximating Multi-Objective Time-Dependent Optimization Problems	Said Dabia, El-Ghazali Talbi, Tom Van Woensel, Ton de Kok
361	2011	Branch and Cut and Price for the Time Dependent Vehicle Routing Problem with Time Window	Said Dabia, Stefan Röpke, Tom Van Woensel, Ton de Kok
360	2011	Analysis of an Assemble-to-Order System with Different Review Periods	A.G. Karaarslan, G.P. Kiesmüller, A.G. de Kok
359	2011	Interval Availability Analysis of a Two-Echelon, Multi-Item System	Ahmad Al Hanbali, Matthieu van der Heijden
358	2011	Carbon-Optimal and Carbon-Neutral Supply Chains	Felipe Caro, Charles J. Corbett, Tarkan Tan, Rob Zuidwijk
357	2011	Generic Planning and Control of Automated Material Handling Systems: Practical Requirements Versus Existing Theory	Sameh Haneyah, Henk Zijm, Marco Schutten, Peter Schuur
356	2011	Last time buy decisions for products sold under warranty	M. van der Heijden, B. Iskandar
355	2011	Spatial concentration and location dynamics in logistics: the case of a Dutch provence	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
354	2011	Identification of Employment Concentration Areas	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
353	2011	BOMN 2.0 Execution Semantics Formalized as Graph Rewrite Rules: extended version	Pieter van Gorp, Remco Dijkman
352	2011	Resource pooling and cost allocation among independent service providers	Frank Karsten, Marco Slikker, Geert-Jan van Houtum

351	2011	A Framework for Business Innovation Directions	E. Lüftenegger, S. Angelov, P. Grefen
350	2011	The Road to a Business Process Architecture: An Overview of Approaches and their Use	Remco Dijkman, Irene Vanderfeesten, Hajo A. Reijers
349	2011	Effect of carbon emission regulations on transport mode selection under stochastic demand	K.M.R. Hoen, T. Tan, J.C. Fransoo G.J. van Houtum
348	2011	An improved MIP-based combinatorial approach for a multi-skill workforce scheduling problem	Murat Firat, Cor Hurkens
347	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	Joint optimization of level of repair analysis and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Inventory control with manufacturing lead time flexibility	Ton G. de Kok
344	2011	Analysis of resource pooling games via a new extenstion of the Erlang loss function	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Vehicle refueling with limited resources	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger
342	2011	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Bilge Atasoy, Refik Güllü, TarkanTan
341	2011	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Analysis of a two-echelon inventory system with two supply modes	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
334	2010	Flexible Heuristics Miner (FHM)	A.J.M.M. Weijters, J.T.S. Ribeiro

333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan
330	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
		Preventing or escaping the suppression	

320	2010	mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
319	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Minimal Protocol Adaptors for Interacting Services	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	Teaching Retail Operations in Business and Engineering Schools	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
316	2010	Design for Availability: Creating Value for Manufacturers and Customers	Pieter van Gorp, Rik Eshuis.
315	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	Getting trapped in the suppression of exploration: A simulation model	S. Dabia, T. van Woensel, A.G. de Kok
313	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
	2010		
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain	Jan C. Fransoo, Chung-Yee Lee

		Performance	
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis

281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	R. Seguel, R. Eshuis, P. Grefen
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl