Scalable Computing:
Practice and Experience

Scientific International Journal
for Parallel and Distributed Computing

ISSN: 1895-1767

Universitatea de Vest din Timişoara

Volume 13(4) December 2012

SUBSCRIPTION INFORMATION: please visit http://www.scpe.org

Scalable Computing: Practice and Experience

Volume 13, Number 4, December 2012

TABLE OF CONTENTS

Special Issue on Agile and Self-Organizing Enterprise Information Systems: Developing a Cloud Platform:

Introduction to the Special Issue **iii**
Enn Õunapuu and Vlado Stankovski

Agile BPM in the age of Cloud technologies **285**
Jiri Kolar and Tomas Pitner

Assessment of Supply Chain Agility in a Cloud Computing-based Framework **295**
Susana Azevedo, Paula Prata, Paulo Fazendeiro and V. Cruz–Machado

A Panorama of Cloud Services **303**
Dana Petcu

A Simulation Platform for Evaluation and Optimization of Composite Applications **315**
Jānis Grabis and Martins Bonders

Integration of cloud-based services into distributed workflow systems: challenges and solutions **325**
Pawel Czarnul

Regular Papers:

An Algorithm for Trading Grid Resources in a Virtual Marketplace **339**
Benjamin Aziz

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. iii–iv. http://www.scpe.org
ISSN 1895-1767

INTRODUCTION TO THE SPECIAL ISSUE ON AGILE AND SELF-ORGANIZING ENTERPRISE INFORMATION SYSTEMS: DEVELOPING A CLOUD PLATFORM

Dear SCPE readers,

We are moving to a new era with rapid changes. The business context is changing rapidly and we need to change our information system very quickly. We have to change the way we create Enterprise Information Systems. The problem is that despite Moore's law, (the capability of computers doubles every eighteen months, the programs and their creation are becoming slower and slower (Wirth's law – Software is getting slower more rapidly than hardware becomes faster). In summary the creation of information systems is slow and very costly. This is not the only problem with our information systems. The second problem with created information systems is that in many cases they fulfil functional requirements, but at the same time they do not add value to the enterprise. The value requirement is not posed or measured.

In this special issue we analyse possibilities of using concepts of living systems in the context of enterprises and its information systems. The theoretical foundations for creation living systems and the main features of living systems are presented. Then, the conditions of staying alive and emergent behaviour are presented. Main contributions presented in this book will be where the measure of business processes and decision support methods usage are analysed for enacting services from the cloud. Model driven development enables a drastic raise in performance of developers.

The mission of the special issue is to help enterprises to move from older information architectures to the new web based cloud platform. The technical side, and importance of cloud computing, is now more or less clear. The problem with using cloud services and integrating them with legacy systems in enterprise will be a less covered topic.

Enn Õunapuu
Faculty of Information Technology
Tallinn University of Technology, Estonia

Vlado Stankovski,
Faculty of computer and information science
University of Ljubljana, Slovenia

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 285–294. http://www.scpe.org
ISSN 1895-1767

AGILE BPM IN THE AGE OF CLOUD TECHNOLOGIES

JIRI KOLAR* AND TOMAS PITNER

Abstract. This article is focused on application of agile principles during adoption of Business Process Management (BPM) in an organization. We propose some agile techniques for gathering requirements and iterative process design. Such techniques help to obtain realistic processes which are easily adaptable to changing business requirements and do not restrict organization's flexibility. We also discuss general obstacles of BPM adoption process identified by a related research, which confirm the necessity of more systematic approach to BPM adoption process. Further we present an outline of our methodology for agile BPM adoption, which propose a collaborative approach to process design with help of Process Collaboration Environment. At the end we discuss how Cloud technologies can foster BPM agility.

Key words: Business Process Management, Agile BPM, Adaptive Case Management, Small and Medium Enterprises, BPM adoption methodology, BPM framework, Collaborative process design, Business analysis

1. Introduction - Role of BPM in agile enterprise. The purpose of the article is to present Business Process Management (BPM), often considered a rigid approach to managing the organization in agile context. Despite the fact agile manifesto refers to : "Individuals and interactions over processes and tools" [29], we want to present some agile approaches to BPM which show that business processes designed and managed with agility in mind can actually foster interactions and provide hospitable environment for flexible collaboration. This article discusses how BPM can be successfully adopted with respect to agile principles and improves organization's efficiency without loss of flexibility.

To preserve business flexibility we have to carefully maintain the link between organization's processes and organization's strategy and goals [19]. We discuss how to react quickly to any changes in business requirements and reflect them quickly in organization's processes. In today's dynamic business environment, such changes happen often, usually initiated by change of customer requirements or situation on the market [1].We also discuss how to track the impact of process changes by gathering process data and use them as an input for further processes improvement [5].

Adoption of "agile BPM" is especially relevant for Small and Medium Enterprises and Organizations (SME) who can benefit from BPM as well [3]. For such organizations flexibility is often important competitive advantage.

At the end of this article we outline methodology for performing agile BPM adoption and discuss how can modern Cloud technologies simplify the BPM adoption process and provide useful technologies which help with technical aspects of BPM and reduce cost of technical implementation of a BPM solution.

2. Background.

2.1. Enterprise agility and BPM. Dynamic changes of global market of today significantly elevate the importance of enterprise agility. The paper [4] defines Enterprise agility as an ability to detect opportunities for innovation and seize them by assembling requisite assets, knowledge and relationships with speed and surprise.

Business processes have strong relationship to organization's operational agility, which reflects the ability of organization's business processes to accomplish speed, accuracy and cost economy in the exploration of opportunities for innovation and competitive action. In such way organizations can rapidly redesign the existing and create new business processes for exploiting dynamic marketplace conditions [22]. Thus agility is more likely to emerge from a creative process of exploration, and not from mechanistic, prescriptive and commoditized techniques and technologies [9].

2.2. BPM introduction. The concept of Business Process Management to extent we understand it today has relatively short history, most of the first serious remarks having around ten years. The definition of this term very much depends on two different perspectives. For more comprehensive historical overview see [2], [1]. On the one hand, from management perspective, BPM is a way to organize a work flow in an organization. It is a dynamic approach where operations of an organization are described by processes. A process is defined as a repeatable sequence of activities, linked to organizational business goals. Execution of the processes contributes to fulfilment of these goals [1]. On the other hand from technical perspective, BPM is an approach to design

*Masaryk University, Faculty of Informatics Botanicka 68a, 60200 Brno, Czech Republic (kolar@fi.muni.cz).

of Enterprise Information Systems and way how to think about system's behavior. BPM prescribe certain architectural model where services are being orchestrated by a Process engine, which perform actual process execution. Software suites for such process design and execution are called Business Process Management Systems (BPMS). The technical perspective is not mandatory for adoption of BPM in an organization in cases where most of the processes are human-centric and Enterprise Information Systems do not play such important role in the organization [5]. BPM as we know today merges those two perspectives into more holistic model, which encompasses strategy, people, business processes and technology [15]. Also practitioners confirm importance of this link is often omitted and lead to inefficient BPM adoptions [19], [5].

2.3. Recent evolution of BPM. BPM has roots in its predecessor, Workflow Management. [28] The problems addressed by Workflow Management are covered by BPM as well, but BPM cover the whole process lifecycle, starting with business analysis, through process modeling and execution to monitoring and process optimization. At the very beginning the main focus of BPM was inherited from Workflow Management and it focused predominantly on the technology and process components of BPM, often taking a very mechanistic view of business processes [27], considering primary BPM as technique for process automation. In other words, at the beginning the technical perspective was observed to be more important [27] than the management one. However over time the importance of management perspective grew and today we understand that organizational changes towards the process-oriented principles (today we call it BPM adoption) is crucial for success of any BPM-enabled technical solution [5]. The strong focus on technical perspective of BPM turned to be successful in large projects driven by organization with strong need for automation of bureaucratic processes mostly in banking and insurance industry. These organizations usually had to convert their management structure to some flat model due to their size and naturally had some form of role-based process driven management model. Thus implementing such model into their ICT system did not mean complete change of mindset. A bit different situation is in SME sized enterprises and public organizations of that size. They very often stick with functional hierarchical organizational models and business processes are driven ad-hoc without clear process and role definitions [20], [21].

2.4. BPM methodologies and methods. One of serious problems in BPM context is lack of a methodologies and best practices for end-to-end BPM adoption. This problem is confirmed by both practitioners [5] and scientists [19], [10]. They agree especially on deficiency of systematic methodologies guiding through the important early phases of BPM adoption, which involve gathering the information for process modeling, mapping business goals to processes and linking business KPI's to process metrics.

There are existing techniques and methodologies for certain phases of BPM adoption. Despite the fact they do not fit completely for agile approach to BPM adoption, some of them are definitely inspiring and we can leverage some of their principles in agile context as well.

One of the most complete existing methodologies for end-to-end BPM adoption is CBM-BPM-SOMA developed in IBM. It is actually a merge of three separate methods linked to each other. This triplet consists of Component Business Modeling (CBM), which is mainly a technique used for organization assessment and business analysis, originally designed for outsourcing purposes. The second BPM, the core method focused on process analysis. And last, much more technically-oriented Service Oriented Modeling and Architecture (SOMA) technique, mainly focused on efficient identification, definition and composition of services with strong emphasis on service re-usability and governance [6]. This triplet of techniques provides general guidelines for whole BPM adoption cycle, going from business analysis, through process analysis, design of system architecture and ends with implementation of fine grained service oriented solution orchestrated by a BPMS. Nevertheless this methodology is designed for adoption of large scale full featured BPM solution, which includes automation by the usage of one IBM BPM products and the integration of various services and systems. Also for successful use of this methodology it has to be combined with complex knowledge-base of best practices. Several vendors such as Software A.G, Oracle have similar complex methodologies and a set of best practices designed for large solutions, but they keep it carefully a secret. Such solutions fit well complex BPM solutions of large enterprises, similarly CBM-BPM-SOMA and they are not suitable for agile small scale BPM adoptions.

Of course many vendors of BPMSes and related products provide their proprietary techniques and methodologies for gathering BPM requirements and consequent process design, nevertheless those are usually not publicly available and they are tight of characteristics of particular product.

2.5. Adaptive Case Management overview. Probably the most significant approach to fostering agility of BPM-like solutions is Adaptive Case Management (ACM). This approach is designed for environ-

ment with high amount of heterogeneous knowledge-intensive work. Adaptive Case Management reached solid amount of publications [7], [8] and became recognized by subjects focused on trends in management and related technologies such as Gartner or Forester [16].

Motivation for ACM has its primary roots in law investigations at U.S. courts, insurance business and healthcare. Such concept was where it was designed for documenting and investigation of criminal cases, insurance claims or patient's treatments [7]. ACM is focused on knowledge workers, who perform knowledge-intensive work, where rigid predefined processes can be observed as an obstacle. Case participants (investigators, clerks or doctors) collect relevant data about particular case from various sources and perform heterogeneous sequences of activities which vary case to case. They choose the order of tasks themselves and create pattern, which can be repeated or extended next time the same or similar case is being processed [7], [8]. These patterns can be recorded and observed as incomplete processes. Such approach can serve as an inspiration for more agile thinking about processes and we can find research focused on definition of such incomplete processes, such as AGLIPO project [11], [13]. Some other researchers go a little bit further and propose techniques for non-intrusive manual process discovery with techniques introduced by social networks [14].

3. Main section.

3.1. Agile process design. As we mentioned before, one of the key factors of successful BPM adoption especially in SME sized organization is preservation of flexibility. It is crucial to choose a good level of process rigidity. More authoritarian processes definitely set an order in a company and if they are well defined they can lead to good performance. Nevertheless for knowledge workers rigid definition of processes very often mean decrease of productivity [12]. Authoritarian procedures often create obstacles for them. More recent approaches such as Adaptive Case Management can help to challenge this problem [7]. In SME sized organization we have often higher percentage of knowledge workers. More precisely said, the line between a knowledge worker and a routine worker is not as clear as in large-sized organizations. In certain activities of SMEs people often act as knowledge workers, whereas sometimes they do routine work as well. Therefore, thus we cannot simply stick with pure rigid BPM or ACM approach, we have to stay somewhere in the middle. To achieve a good balance between agility and structured processes, we have to keep agility in mind during process definition and modeling phase. We try to define rough process structure and identify sub-processes where we expect different behavior according to particular process instances. Such sub-processes can be easily replaced by their new or ad-hoc versions, or we can create incomplete sub-process without defining its structure [11]. To decide which parts of the process need such specific treatment is not always easy, we have to work closely with process participants to recognize such sub-processes.

3.2. Alignment of processes to business goals. One of the crucial conditions for successful BPM adoption is to establish a linkage between organization's goals and processes [19], [13]. Business plan has to be defined in detail and general goals have to be decomposed to measurable objectives, which are mapped to processes inside the organization [27]. Obviously a process that does not contribute to fulfilment of some objective or goal is useless. To be able to map business elements to processes, usually business strategy should be build according to some scheme. Probably the most complete standard in context of Business analysis necessary in early phases of BPM adoption is Business Motivation Model well specified in Business Motivation Model specification [30]. BMM is one of OMG standards family. This standard specification describes a method for identification and proper definition of vision, business goals, objectives and other related entities, which are the necessary input for goal-oriented modeling and process definition.

Once we manage to establish a link between goals and processes, we want to be able to measure how successful we are in our goals and objectives fulfilment by measuring process data [23], [24]. To obtain well measurable processes aligned with our business goals we have to systematically design our processes with business goals in mind. Also metrics we define in process perspective have to be relevant to our business metrics and vice versa. This fact is very often omitted during early stages of BPM adoption process and later when it comes to implementing the business metrics, process developers try to dig anything related to business data from processes. Considering both business and process metrics in early stages of adoption is crucial for successful measurement. This problem was already described by pioneers of process reengineering [18] and remains alive also in modern literature [5].

3.3. Agile methodology for Collaborative approach to process design. In this section we will present a subset of our methodology focused on small-scale BPM adoptions. This subset is focused primarily on

collaboration of initial process design and also on further collaborative improvement of processes. We will put emphasis on involvement of process participants, as they play key role in gathering of requirements in initial process design as well as consequent iterations focused on process improvement. Early draft of the methodology was applied in practice so far in two case studies. First case study was performed in commercial environment, SME software company: IT Logica s.r.o [25] focused on Web-Application development. Second case study was performed in ICT department of Masaryk University in Brno and was focused primarily on ICT services provided internally to the University [26]. In both cases agility and need for more iterative approach to process design and need for further process maintenance was identified as a drawback of our methodology, so we did recently several changes towards more iterative agile principles.

3.4. Planning the BPM Adoption. Adoption consists of several phases. At the end of each phase results should be reviewed and the plan for forthcoming phases should be detailed. In general estimation of effort for each phase is not easy at the beginning and many details about next phase are uncovered at the end of preceding phase. We should also keep in mind that BPM adoption often means changes in both organizational structure and used ICT technologies. This means that changes should be committed iteratively and all new systems should run in parallel and migration should be very careful. Obvious seems to be usage of conventional project management tools which help project manager to deal with planning complexity and make the plan systematic and understandable.

3.5. Adoption participants. BPM adoption should start with identification of participants. Key participants should be chosen very carefully as their contribution can significantly influence the whole adoption. We have to make sure all participants are properly informed about the adoption process, they understand the adoption goals and they should be convinced about potential benefits of adoption process.

We are going to describe following participant roles:

- Sponsor
- Organization's management
- Adoption coordinator
- Process analyst
- EIS designers and developer
- Process participant
- Process maintainer

3.5.1. Sponsor. This role usually belongs to organization owner or CEO. A sponsor provides resources for adoption process such as funding and allocates internal human resources. His commitment is absolutely necessary for success of adoption and he has to clearly understand potential benefits, risks and overall impact on organization.

3.5.2. Organization's management. Each manager has to be fully familiar at least with impact of adoption on his area of responsibility and also understand the big picture of the adoption. On the side of lower management we face often fear of loss of responsibility and importance. Managers play important role in the adoption and we have to carefully explain all benefits adoption can bring to them and make sure all their fears are dispelled.

3.5.3. Adoption coordinator. Usually member of external "BPM team". He usually acts as Project manager of the adoption and he is the core person responsible for entire adoption process. He has to plan the adoption process carefully, execute it and periodically monitor the progress. He should be familiar with organization's business context, cooperate closely with Sponsor and Organization's management. He should be experienced process analyst familiar with issues of process modeling and manage team of process analysts.

3.5.4. Process analyst. Usually also member of external "BPM team", responsible for interviewing process participants, modeling and documenting organization's processes. Good communication skills are a must. He has to have strong knowledge of process modeling techniques and he should have at least basic knowledge of organization's business domain as well.

3.5.5. EIS designers and developer. Internal or external person responsible for design of EIS in target organization. He should have at least basic knowledge of BPMS technologies if a BPMS is used and understand at least basic BPM concepts. He should be aware of desired impact of adoption on organization's EIS.

3.5.6. Process participant. Internal organization's worker performing activities of modeled processes. He usually has a key knowledge about how the process works in details and he should serve as main sources of information about modeled processes. Similarly to organization's managers, participants are often afraid of negative impact of BPM adoption on his work. Thus we have to carefully explain all benefits adoption can bring to him and make sure he is willing to collaborate.

3.5.7. Process maintainer. Internal person made responsible for further maintenance and improvement of processes after adoption. He should work closely with adoption coordinator and team of process analysts and learn as much as possible. He should learn how to model and modify processes, synchronies changes between organization's business goals & objectives and processes, how to set measures on processes and transform measured data into KPIs. In short, he should be able to perform those steps periodically after end of initial adoption on his own and further develop the organization's processes.

3.6. Setting preceding the adoption. There are several activities, which should be done shortly after kickoff the adoption process.

3.6.1. Introductory meeting. There should be a meeting which introduce the plan of adoption and create common understanding across all involved subjects.

Such meeting should be attended at least by:

- Sponsor and part of organization's management directly involved in adoption process
- Adoption coordinator, eventually some process analysts
- As much as possible process participants
- Process maintainer

On such meeting we should present most important facts about the adoption and provide space for discussion Presentation should cover:

- Basic facts about the adoption, such as purpose, goals and expected outcomes
- Highlight the importance collaboration across all the involved subjects
- Outline the whole adoption plan and rough time schedule
- Brief introduction of process used process modeling technique
- Introduction of used PCE
- Rough structure of process interviews

3.6.2. PCE setting. We have to make sure all users of our PCE are able to access it and know how to use it. We should also provide a person supporting PCE users to achieve maximum contribution. There should be some example processes as well as feedbacks, so users can use it as a template.

3.7. Adoption phases. Adoption consists of several phases performed in a recommended order. However in some cases the sequence of these phases has to be tailored to the situation. For example when the business goals and objectives of the organization are relatively simple, but the business of the organization itself is built on critical mass of EIS components and ICT services, the analysis of those systems turns to be more important and it can be performed earlier. However this leads to the bottom-up approach to BPM adoption, which is not really in scope of the researched methodology.

We are going to describe following phases:

- Organization assessment phase
- Initial process mapping phase
- Iterative process improvement

3.7.1. Organization assessment phase. In this phase we gather context information about organization and its business, collect business related information and use it as an input for process analysis and design. These activities are done by Adoption coordinator by performing interviews with organization's management and root stakeholders.

Roles involved: Sponsor, Organization's management, Organization's management, Adoption coordinator

Phase inputs:

- Previous efforts of organization assessment
- Business plan
- Any documents describing organization structure
- Definitions of metrics and previous business data

- ICT services documentation

Phase activities:

1. Review and refine business plan & vision
2. Review and refine goals and objectives (G&O)
3. Review and specify business metrics and KPIs mapped to objectives
4. Describe in detail organizational structure, including roles and responsibilities
5. Describe business components (organization units)
6. Describe ICT services both consumed and provided internally and externally
7. Create priority list of business activities
8. Create complete list of relevant processes mapped to business activities

We first collect the AS-IS state, discuss it with the management and define initial TO-BE state. Nevertheless TO-BE state should not involve much reengineering at this stage. It can involve:

- Business plan re-engineering
- KPIs and metrics definition and re-engineering
- Estimation of quality and costs of ICT services
- proper mapping of G&O to processes
- clear definition of roles

For more formal description of organization business plan&vision and Goals&Objectives we can use some more formal techniques such as Business Motivation Model [30]. However BMM is quite complex technique and can fit only for organizations with more complex business planning. Phase outputs:

- Refine business plan, vision,
- G&O and related KPI definitions
- Description of organizational structure with subordinations, roles and responsibilities
- Prioritized list of business activities mapped to existing processes

3.7.2. Initial process mapping phase. To obtain realistic processes that correspond to reality, the involvement of each process participant to the process definition in "design time" is crucial. Otherwise we can easily end up with idealistic process definitions dreamed by management that have nothing to do with reality. The more intuitive technology we use for sharing the modeled processes with process participants, the more efficient collaboration we achieve.

Phase inputs:

- Prioritized list of business activities mapped to existing processes (from previous phase)
- Any documents describing activities involved in modeled processes
- KPI definitions (from previous phase)

Phase activities:

1. Complete prioritized process list (existing and new) with process owners assigned
2. Interview process participants and define initial processes
3. Create Detailed BPMN 2.0 models of chosen processes and write complementary descriptions
4. Define roles within processes and map them to organization's roles
5. Identify and refine process metrics linked to KPI's
6. Set up PCE and publish processes there.

Phase outputs:

- Prepared PCE
- Complete list of prioritized processes with assigned owners and roles
- Initial version of process BPMN 2.0 models and descriptions published in PCE
- Clear definitions of process metrics and mapping to KPI's
- Initial feedbacks about processes from participants stored in PCE

The main responsibility of good process design of the modeled processes lies on Adoption coordinator. It is generally assumed that the processes should be modeled by Process Analysts who are dedicated to this activity, but they do not usually understand each process in detail. Thus they have to cooperate with process participants who are involved in the activities performed within the process. Initial set of defined processes should be also approved by organization's management and sponsor of the BPM solution. Steps of the initial process mapping phase are described in Fig. 3.1. Here the adoption coordinator captures the scope of the organization and creates list of processes. Then he models and describes the selected processes and publishes

the draft to the PCE. At this step the process participants and organization's management should provide rich feedback and comments, they have to identify parts of the process which are faulty, unclear or too general. Such feedback is stored in the PCE. After the predefined period of time, Adoption coordinator collects the provided feedback and closes the initial phase.

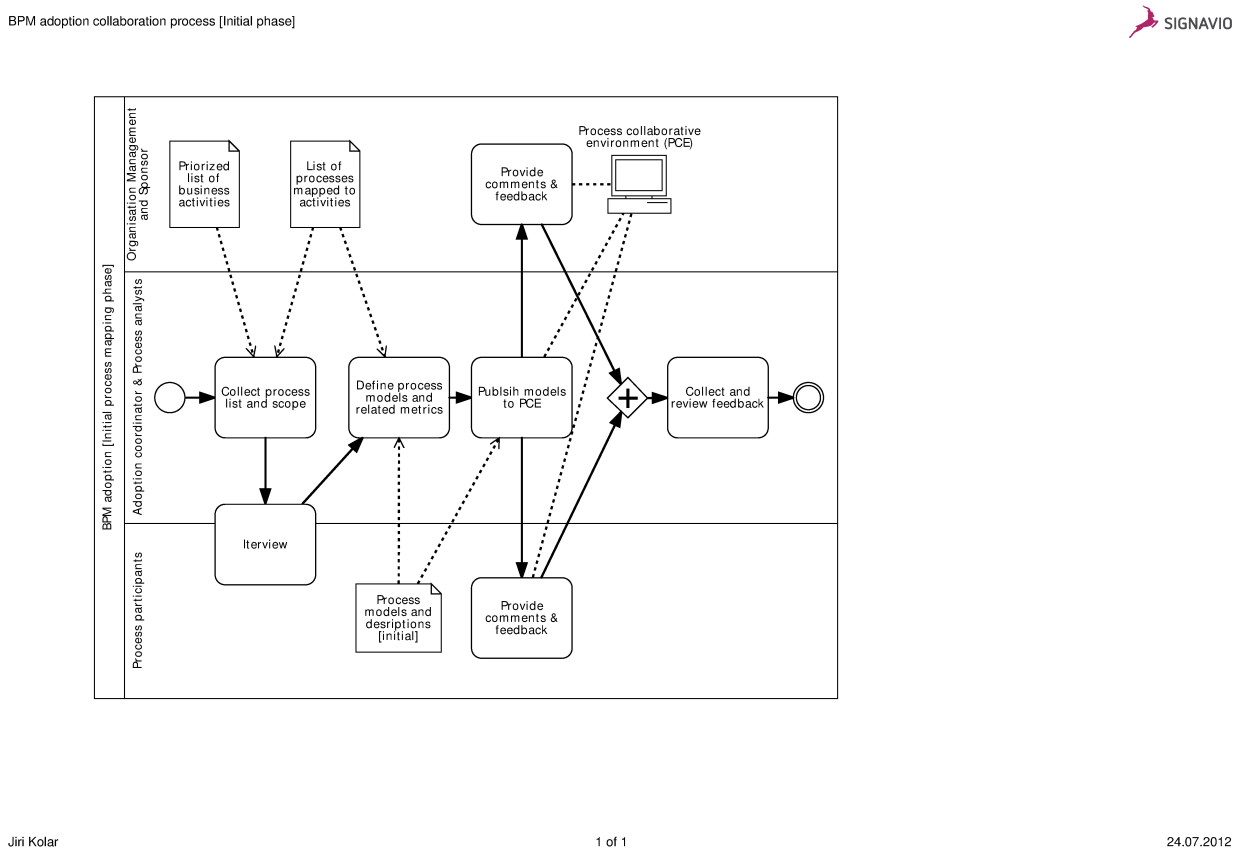

Fig. 3.1: Initial process mapping phase

3.7.3. Iterative process improvement. This phase should be performed in short iteration cycles (I would recommend 1-6 months), the anticipated changes should be also of reasonable size, corresponding to the available human resources. Phase inputs:

- Feedbacks about processes from participants and management stored in PCE
- Process update requests (2+ iteration)
- Process data (2+ iteration)

Phase activities:

1. Modify process models and descriptions according to feedbacks and change requirements
2. Discuss changes and get approval with Organization's management and Sponsor
3. Publish updated processes to PCE and open for discussion
4. Implement changes in processes in EIS
5. Measure process execution automatically or manually
6. Collect process data
7. Let the Organization's management and Sponsor to evaluate measured data
8. Collect Process update requests from Organization's management and Sponsor

Phase outputs:

- Modeled and described processes published to PCE
- Updated processes implemented in organization's EIS
- Process data
- Process update requests for next iteration

The steps of this phase are described in Fig. 3.2. Here the Adoption coordinator initiates first iteration of improvement phase, reviews collected feedbacks and modifies defined process models according to it. Modified models are reviewed by organization's management and are either approved or disapproved and send back for further modification. In case of approval the solution designer publishes modified version to the PCE and implements the approved processes in EIS. Implementation depends on the agreed level, it can start from simple modification of existing activities in EIS for completing process-engine based implementation in a BPMS. By completing these steps the implementation processes are measured. In case of basic implementation of conventional EIS processes, they have to be measured manually, by collecting events indicating performance of particular activities or even by noting progress per process. In case of automated monitoring tools, data are collected automatically by such tool. After the period of measurement, process data are evaluated by Organization's management, and process changes are requested for processing to the next iteration.

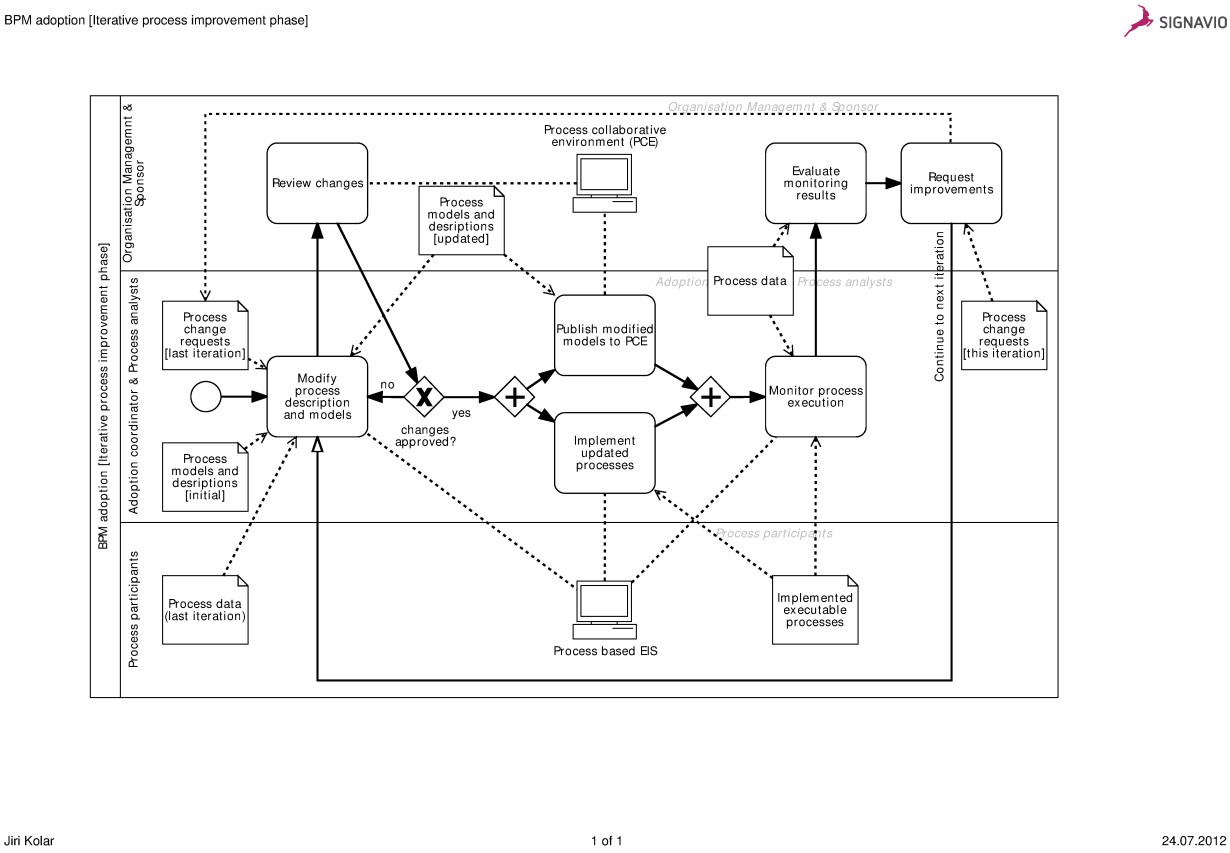

Fig. 3.2: Iterative process improvement phase

4. Future research directions.

4.1. BPM in Cloud. With respect to all currently existing cloud-based BPMSes we can say that nowadays purely cloud-based BPMS is still just a dream. Once it comes to choosing reliable BPMS during BPM adoption, we have to stick with existing product-licensed BPMS from traditional vendors. Battle on the field of cloud-enabled BPMS is already started, here the most visible players are current BPMS vendors who try to migrate

their existing products into cloud and provide them in SaaS mode. The main advantages of cloud-enabled BPMS include generally accepted SaaS benefits and several others specific for BPM context.

Probably the more convenient way to leverage cloud technologies is to use BPMS for integrating cloud services relevant to our business. We can leverage such services as email, messaging, document management, web-services, external software components, existing EISes deployed in cloud environment, and integrate them together with locally-hosted BPMS.

Many of todays BPM vendors visible in Gartner's magic quadrant such as IBM, Signavio, Intalio, Pega [17] and others make quite extensive efforts to develop server-side environments for collaborative process design. Nevertheless, most of them allow only local installations on private servers, which get them closer to "private cloud" concept. Public cloud services often rely on open-source technologies. In that sense probably the most popular is Oryx visual editor developed as open source project [31], tailored by some BPMS vendors such as Signavio and Alfresco. However Oryx is a visual modeling tool, and for full blown PCE we need some advanced features such as documentation tool embedded to process modeling in order to ensure collaborative functionalities for participant's feedback and wider collaboration.

There are potential advantages of moving PCE into cloud environment:

- Cloud enables efficient sharing and real-time collaboration
- Cloud enabled PCE is easily accessible from any environment and OS, it does not require any local installations of the dedicated tools
- Centralized storage allows proper versioning, tracking of changes and history

Same as many other technologies and services which are slowly migrated from local-based SW into Cloud, the BPMSes are on their way to cloud as well. However, this is a matter of several years. One of the major reasons is Cloud interconnection. Once you have local-based SW solution which consumes several services from different cloud environments, and you integrate them together, then the situation is quite simple. You just make sure that each of your cloud connectors is working properly. Once you want to migrate the orchestration component into some could, the situation is getting more complicated. As long as you consume the services from the same cloud where the orchestration engine is located, you are quite safe because both of them are in "uniform" cloud environment. Problems come when you want to consume services from different cloud providers. The interoperability across different clouds is something not well established, as a matter of fact most of SaaS cloud providers are competitors while the cloud interoperability is not really their business goal.

5. Conclusion. We briefly introduced BPM and its history, discussed common issues of BPM adoptions and highlighted the need for agility in context of a BPM adoption. We presented some contemporary research efforts which confirm the need for more systematic approach to adoption of BPM in organization with emphasis on SME sized solutions. Furthermore we reviewed some existing techniques and methodologies for BPM adoption and we outlined part of the methodology for more agile adoption of BPM. At the end we discuss the situation in cloud technologies related to BPM. Our findings show that there are many open questions concerning the problem of increasing agility in the BPM adoption process where cloud technologies can help significantly and simplify the technological perspective of this discipline. Nevertheless the maturity of existing BPM technologies in Cloud environment is low and there is still lots of work to be done towards development of something we call Cloud BPM.

Acknowledgments. This work was supported by the European Union's territorial cooperation program between Austria and the Czech Republic of the under the EFRE grant M00171, project "iCom" (Constructive International Communication in the Context of ICT).

REFERENCES

[1] Smith, H. and Fingar, P. (2003). Business Process Management: The Third Wave, Meghan-Kiffer Press, Tampa.
[2] Harmon, P. (2003) Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes, Morgan Kaufmann, San Francisco.
[3] Raymond, L., Bergeron, F. and Rivard, S. (1998) Determinants of business process reengineering success in small and large enterprises: an empirical study in the Canadian context, Journal of Small Business Management, vol. 36, no. 1, pp. 72-85.
[4] Goldman, S. L., Nagel R.N. and Preiss, K., Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer, Van Nostrand Reinhold, NY, 1995.
[5] Jeston, J. and Nelis, J.: Business Process Management: Practical Guidelines to Successful Implementations, Butterworth-Heinemann, 2008
[6] Fiammante J. , Dynamic SOA and BPM: Best Practices for Business Process Management and IBM Press, 2009

[7] Swenson, K. D., Jacob P. Ukelson, J. T., Khoyi, D., Kraft, F. M., McCauley, D., Palmer, N., et al. Mastering the Unpredictable. Tampa, FL, USA: Meghan-Kiffer Press., 2010
[8] Swenson, K. D., Kraft, F. M., Palmer, N., n al., e. . Taming the Unpredictable. Lighthouse Point Florida : Future Strategies Inc. , 2011
[9] Desouza K. (ed), Agile Info. Systems, Elsevier, 2007.
[10] Singer, R. and Zinser, E.: Buchwald, H. and Fleischmann, A. and Seese, D. and Stary,Business process management s-bpm a new paradigm for competitive advantage? S-BPM ONE Setting the Stage for Subject-Oriented Business Process Management, Springer Berlin Heidelberg, 2010.
[11] Silva, A.R., Meziani, R., Magalhaes, R., Martinho, D., Aguiar, A., Flores, N.: AGILIPO: Embedding Social Software Features into Business Process Tools. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 219230. Springer, Heidelberg (2010)
[12] Word, J. and Magal, S.: Essentials of Business Processes and Information Systems, Wiley, 2009, 978-0470418543.
[13] Meziani, R.; Saleh, I.; , "Towards a collaborative business process management methodology," Multimedia Computing and Systems (ICMCS), 2011 International Conference on , vol., no., pp.1-6, 7-9 April 2011
[14] David Martinho, Antnio Rito Silva: Non-intrusive Capture of Business Processes Using Social Software - Capturing the End Users' Tacit Knowledge. Business Process Management Workshops (1) 2011: (pp. 207-218)
[15] Gartner Research: Gartner position on Business Process Management, Gartner Research note, ID: G00136533, 2006.
[16] Gartner Business Process Management Summit April 2011, Gartner Research note 2011
[17] Gartner Business Process Management, Gartner Magic quadrant overview 2012
[18] Davenport, T. and Short, J.: The new industrial engineering: Information technology and business process redesign, Sloan Management Review, 1127, 1990
[19] Indulska, M., Chong, S., Bandara, W. ,Sadiq,S. , Rosemann, M. (2007). Major issues in business process management: a vendor perspective. Proceedings of the Pacific Asia Conference on Information Systems (PACIS2007).
[20] Chapman, R., and Sloan, T. (1999) Large firms versus small firms - do they implement CI the same way?, The TQM Magazine, vol. 11, no. 2, pp. 105-110.
[21] Spanos, Y., Practacos, G., and Papadakis, V. (2001) Greek firms and EMU: contrasting SMEs and large-sized enterprises, European Management Journal, vol. 19, no. 6, pp. 638-648.
[22] Sambamurthy, V., Bharadwaj, A., Gover V. Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms, MIS Quarterly, Vol 27, No. 2, pp.237-263/June 2003. (2003)
[23] Kolar, J. Business Activity Monitoring. Unpublished Master Thesis. Masaryk University, 2009. Retrieved July, 2012, from http://is.muni.cz/th/72655/fi_m/lang=en
[24] Kolar, J. A framework for Business Process Management in small and medium enterprises. Unpublished Dissertation Proposal. Masaryk University, 2011
[25] Kolar, J.: Procesni analyza pro Centrum Informacnich Technologii Filozofke fakulty MU, Unpublished commercial case study, September 2011, (2011b)
[26] Kolar, J.: Procesni analyza pro spolecnost IT Logica s.r.o., Unpublished commercial case study, July 2011, (2011c)
[27] Marjanovic, O.; , "Inside Agile Processes: A Practitioner's Perspective," System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on , vol., no., pp.1-10, 5-8 Jan. 2009
[28] W.M.P. van der Aalst, A.H.M. ter Hofstede, M. Weske, Business process management: a survey, in: W.M.P. van der Aalst, A.H.M. ter Hofstede, M. Weske (Eds.), Proceedings of International Conference on Business Process Management, 2003.
[29] Highsmith, J.,(2001). The Agile Manifesto. Retrieved, June, 2012, from: http://www.agilemanifesto.org/
[30] OMG. Business Motivation Model (BMM) Specification, v1.0. 2008. Retrieved, June, 2012 from: http://www.omg.org/cgi-bin/doc?formal/08-11-02.pdf.
[31] Oryx website.Retrieved, June, 2012, from http://bpt.hpi.uni-potsdam.de/Oryx/

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 27, 2012
Accepted: Jan. 09, 2013

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 295–302. http://www.scpe.org
ISSN 1895-1767

ASSESSMENT OF SUPPLY CHAIN AGILITY IN A CLOUD COMPUTING-BASED FRAMEWORK

SUSANA AZEVEDO*, PAULA PRATA†, PAULO FAZENDEIRO‡, AND V. CRUZ–MACHADO§

Abstract. This paper presents an approach to evaluate the supply chain agility behaviour consisting in the development of an integrated index, with the data gathering, transmission and processing supported by a cloud-computing environment. The proposed approach relies on the development of two agility indices: one to assess the individual company agile behaviour, and the other one to determine the same behaviour for the entire supply chain. The supply chain is presented as a living, self-organizing open system that has the ability to incorporate new efficient agents and to remove the weakest ones. A special emphasis is given to the living subsystem responsible for the agility assessment, namely regarding the conceptual details of the components necessary to gather, process, coordinate and control the flow of information in the cloud.

Key words: Agility index, cloud computing, agile supply chain management.

1. Introduction. A supply chain (SC) can be described as a chain that links various agents, from the customer to the supplier, through manufacturing and services so that the flow of materials, money and information can be effectively managed to meet the business requirements [8]. In present-day business there is the assumption that SC's compete instead of companies. Supply Chain Management (SCM) is considered a strategic factor for enhanced competitiveness, better customer service and increased profitability.

The static connections between enterprises are typically insensitive to the changes in the business environment. Instead, flexible, agile, short and dynamic connections that facilitate seamless information flow across different value chains are needed for dynamic business partnership formation to take place [4].

The extent of business-to-business (B2B) interactions and communication could be overwhelming between the various parties and services especially in an adaptive environment where a lot of information needs to be exchanged. This requires higher capabilities of interaction and communication among enterprises. The necessity to improve these capabilities makes the SCs adoption of a more agile behaviour an urgent matter, altogether with a deeper awareness about their implementation level of the main practices associated to the agile SCM paradigm. In this context, the main objective of this paper is to propose a cloud-based framework enabling that global supply chains can assess their Agile index, that is, to get information on the level of agility of their practices.

Our own view of the SC as an open living, self-organizing system that has the ability to interact with its environment is presented. This takes place by means of information, material and money exchanges as well as by the dynamic incorporation of new efficient agents and removal of non-agile ones. It is becoming clear that the SCs must be able to adopt the right decisions regarding this latter issue in order to survive. Thus the capability to assess their agility level is a critical asset in maintaining a high fitness to a volatile environment (economic, social and business processes).

More specifically in this paper, without forgetting the metabolic processes of material and money exchanges, we focus our attention at a conceptual level on the subsystems necessary to process information for the coordination, guidance and control of the agility assessment system.

2. Background. The changing conditions of competition, the increasing levels of environmental turbulence and requirement for organizations to become more responsive and also more efficient are driving the interest in the concept of supply chain agility [17]. The agile paradigm is related to market sensitiveness and confers the ability to read and respond to real demand [6]. Since customer requirements are continuously changing, it is more difficult for SCs to deliver the right product, in the right quantity, in the right condition, to the right place, at the right time, at the right cost. To overcome these conditions, Hoek et al. [9] suggest that

*UNIDEMI - Department of Business and Economics, University of Beira Interior, 6200-001 Covilhã, Portugal. (sazevedo@ubi.pt).

†Instituto de Telecomunicações (IT), Department of Informatics, University of Beira Interior, 6200-001 Covilhã, Portugal. (pprata@di.ubi.pt).

‡Instituto de Telecomunicações (IT), Department of Informatics, University of Beira Interior, 6200-001 Covilhã, Portugal. (fazendeiro@ubi.pt).

§UNIDEMI- Department of Mechanical and Industrial Eng., Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2829-516 Caparica, Portugal. (vcm@fct.unl.pt).

SCs should be agile to respond appropriately to market requirements and changes. Khan et al. [13] also state that agile SC practices support an appropriate response to market instability, responding in real time to the unique needs of customers and markets. That is, the SCs should have flexible and responsive capabilities in terms of their processes, networks and how they are integrated across other organisations [18]. Agile Supply Chain Management is crucial since it intends to create the ability to respond rapidly and cost effectively to unpredictable changes in markets and increasing levels of environmental turbulence, both in terms of volume and variety [1, 5, 10].

A new business model is introduced by Cloud Computing where consumers can have access to hardware and software, in a pay-per-use manner as we do with public utilities like water or electricity. According to Mell [16] cloud computing has five main characteristics: on-demand self-service, the user can access computing capabilities when needed without human interaction with the service provider; broad network access, capabilities are available over the network and accessed by standard mechanisms as internet protocols; resource pooling, the computing resources are pooled to serve multiple consumers; rapid elasticity, capabilities can be elastically provisioned and released according to consumer demand; measured service, resource usage can be monitored, controlled, and reported.

According to the type of provided resources three service models are considered [16, 23]: Infrastructure as a Service (IaaS), when it offers storage, computation and network capabilities; Platform as a Service (PaaS), when it offers facilities to develop software products; Software as a Service (SaaS) when the user can buy a subscription to some on-line software.

Intending to clarify the differences between cloud and conventional computing Armbrust et al. [3] highlight three new aspects in the cloud computing model from an hardware provisioning and pricing point of view: i) "The appearance of infinite computing resources on demand"; through virtualisation technology, cloud computing provides access to a wide range of machines as virtual instances whose number varies depending on the amount of required resources. ii) "The elimination of an up-front commitment by cloud users". iii) "The ability to pay for use of computing resources on a short-term basis as needed". Users can start using a small amount of resources and increasing it as their needs grow. Moreover, they just pay for what they used.

According to several authors [12, 21, 24] in a time where business environment changes very fast, cloud computing appears to be a way to enable companies facing the constant change of customer demands and market conditions, building more dynamic supply chains. While traditional software systems automate processes within a single enterprise, supply chain management may require the collaboration between companies across the entire world. Deploying a supply chain platform in the cloud provides the opportunity for all supply-chain partners for sharing data on the Internet as a pay-as-you-go service.

Since the main objective of this paper is to suggest an agility index assessment model supported on a cloud computing environment, a review of the main agile SCM practices found in the literature was made. Table 2.1 presents a summary of the agile SC practices identified in the literature. The reviewed practices are deployed at three levels of analysis according to the observable dyadic relations in the supply chain: (i) agile practices developed upstream; these are associated directly with interactions between a firm and their suppliers; (ii) agile practices deployed by firms in their daily, internal operations; these depend only on the firms' decisions to enforce an agile behaviour, and (iii) agile practices deployed downstream; these are those incorporating agility concerns in all kinds of flows (materials and information) between the firms and their downstream partners involved in delivery activity.

3. Agility assessment. The numerical assessment of the SC agility level relies on the determination of the initial set of agile practices subject to analysis. Any member (an individual company) of the SC should be able to produce a report on the implementation level of each practice of such set. At the same time a set of relative weights of each practice reflecting the overall evaluation policy of the SC must be available.

After the identification of the target practices, the attribution of relative weights to each practice can be seen as the second critical task for the agile index elicitation. From the entire range of possibilities to establish the weights of the Agile practices the Delphi approach is appealing due to its commitment to a consensual result when resourcing to a panel or committee of experts.

According to Linstone and Turoff [15] the key steps in preparing a Delphi study are: (i) the definition of experts and their selection; (ii) the number of rounds; and (iii) the questionnaire structure in each study round. Generally, the number of rounds ranges from two to seven and the number of participants varies between three and fifteen [20].

Table 2.1: Agile practices in the supply chain context

Agile practices	**References**				
First tier supplier ⟶ Focal firm	[19]	[7]	[14]	[1]	[22]
To use IT to coordinate/integrate activities in design and development.				√	√
To use IT to coordinate/integrate activities in procurement.					√
Ability to change delivery times of supplier's order.					√
To reduce the development cycle time.					√
Focal firm	[19]	[7]	[14]	[1]	[22]
To use IT to coordinate/integrate activities in manufacturing.			√	√	√
To integrate supply chain/value stream/virtual corporation.	√				
To use centralized and collaborative planning.				√	
To reconfigure the production process rapidly.	√				
To produce in large or small batches.		√			
To accommodate changes in production mix.					√
To reduce manufacturing throughput times to satisfy customer delivery.					√
To reduce development cycle times.					√
To minimize set-up times and product changeovers.		√			
To organize along functional lines.			√		
To facilitate rapid decision making.			√		
Focal firm ⟶ Customer	[19]	[7]	[14]	[1]	[22]
To use IT to coordinate/integrate activities in logistics and distribution.					√
To increase frequency of new product introductions.			√	√	√
To speed up adjustments in delivery capability.					√
To speed up improvements in customer service.				√	√
To speed up response to changing market needs.					√
To capture demand information immediately.			√		
To retain and grow customer relationships.			√		
To develop products with added value for customers.			√		

Whenever the linear additive model assumption is verified, the Agility index assessment of the SC is simply reduced to the (weighted) average of the individual companies' indices. From the focal company perspective it may happen that the different roles of the individual components of the SC present diverse relative importance concerning the determination of the Agility index of the SC.

3.1. Agility index for an individual company. To compute the company agility behaviour it must be possible to grade the levels of implementation of the focused agile practices. These indicators form a representative quantification of the n agile practices implemented by each company $(P_{A_1}, P_{A_2}, \ldots, P_{A_{n-1}}, P_{A_n})$. Each indicator is associated with a relative weight reflecting the practice's importance according to the global SC policy and can be measured in a 5 points Likert scale (1 means "practice not implemented" and 5 "practice totally implemented").

For each company the *Agility Behaviour* (AG) index is proposed representing the set of agility-related practices implemented. It is supposed that for each company this index can be computed aggregating the correspondent individual indicators (agile practices) according to their importance. For each company a generic weighted average can be used to compute the AG:

$$AG = \frac{\sum_{i=1}^{n} w_{A_i}.P_{A_i}}{\sum_{i=1}^{n} w_{A_i}}, \tag{3.1}$$

where P_{A_i} represents the implementation level of agile practice and w_{A_i} is the relative weight of the same practice. A total of n practices are considered. The positive weights reflect the relative importance of each practice in the SC. Equation 3.1 shows that the company agility behaviour is a function of each agile practice implementation level (P_{A_i}) and corresponding weight (w_{A_i}). Notice that when the Delphi methodology is followed the denominator of Eq. 3.1 equals to the unity, however different weights attribution schemas can be envisioned.

3.2. Agility index for the supply chain. To synthesize the SC's Agility index, the individual companies' agility behaviours will be considered as sub-indicators which are aggregated in a weighted average to obtain the SC Agility index ($Agility_{SC}$):

$$Agility_{SC} = \frac{\sum_{j=1}^{m} w_{C_j}.AG_j}{\sum_{j=1}^{m} w_{C_j}}, \tag{3.2}$$

where m is the number of companies considered in a particular SC, AG_j stands for the j-th company's agility behaviour given by Eq. 3.1 and w_{C_j} represents the relative contribution of the j-th company to the overall Agility index.

It is assumed that the practices weights are common for all companies belonging to the same SC. Notice that $\sum_{j=1}^{m} w_{C_j} = 1$, hence provided that all the individual agility indices are defined in the range [1, 5] the SC agility index is also described in the same range, where 1 means that the agile indicators are not put into practice by the SC companies and 5 represents an absolute adhesion, from all of the SC companies, to the agile practices.

4. Assessment Example. In this section the construction of the Agility index is illustrated. Seven companies from an automotive SC were chosen to illustrate the Agility index application. The research companies comprise one automaker and four first-tier suppliers, one second-tier supplier and also one first-tier customer. Each company was considered equally important regarding the SC agility assessment, i.e. $w_{C_j} = 1/7$, $j = 1, \ldots, 7$, in Eq. 3.2.

The literature review was used as a guidance to identify the most suited practices for the studied SC, a subset of 15 practices adapted from Table 2.1 was selected. In a first stage the data related to the individual assessments is collected, each company produces the implementation levels, P_{A_i}, $i = 1, \ldots, 15$, of the selected agile practices. This makes it possible to register the agility behaviour of each company, AG_j, $j = 1, \ldots, 7$, and also to compute the Agility index, $Agility_{SC}$, to the entire SC. Table 4.1 presents the collected data and the results of the computation of the agility indices.

Table 4.1: Agility behaviour for individual companies and supply chain.

Practices	**Weight**(w_{A_i})	**Implementation level per company**						
		C1	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*
P_{A_1}	0.063	2	3	2	3	1	4	1
P_{A_2}	0.066	3	5	4	3	5	4	2
P_{A_3}	0.067	3	3	3	3	5	3	2
P_{A_4}	0.067	3	4	3	4	1	3	1
P_{A_5}	0.070	5	4	5	3	1	3	1
P_{A_6}	0.070	5	4	5	4	1	2	1
P_{A_7}	0.069	4	5	2	3	1	3	1
P_{A_8}	0.069	2	5	2	4	5	3	1
P_{A_9}	0.063	3	5	4	5	5	3	5
$P_{A_{10}}$	0.083	3	5	4	5	5	3	5
$P_{A_{11}}$	0.076	5	5	5	5	5	4	2
$P_{A_{12}}$	0.072	5	5	5	5	5	5	5
$P_{A_{13}}$	0.065	3	4	1	4	1	5	3
$P_{A_{14}}$	0.052	3	4	3	3	5	3	3
$P_{A_{15}}$	0.048	3	4	3	3	1	4	1
Company Agility, see Eq. 3.1		3.513	4.368	3.457	3.859	3.192	3.457	2.315
SC Agility Index, see Eq. 3.2					**3.452**			

According to the agility values of the companies $C1$ to $C7$, it is possible to identify the case study company with the better and worst agility behaviour. The better agility performer is the company $C2$ and the worst is the company $C7$. This result comes from the implementation degree of the agile practices in each company. Company $C2$ has totally implemented almost all the analysed agile practices, which makes it an agile performer. Contrary, the company $C7$ only has totally implemented three of the fifteen selected practices.

Summing up and analysing Table 4.1 we can say that the overall agility behaviour of the companies is positive, as reflected in the aggregated SC Agility level, however there is a link in the SC that presents a

negative (less than 3 in the Likert scale) agile value implying that this weak behaviour is perhaps the first that should be corrected in order to enhance the Agility of all the SC. This indicator gives insight on the average behaviour of all the companies that contributes for the production of the final product, which means that if one partner in the SC has not the required flexibility this could compromise the agility behaviour of all the SC and its public image.

5. Cloud Computing Supporting the Living Agility Assessment. Cloud computing has transformed how global business networks interact, delivering a flexible, collaborative model. According to Aljabre [2] Cloud computing provides the ability for multiple users to collaborate on projects or documents in the cloud. This point has been reiterated and reinforced recently as a major selling point to businesses. This makes the cloud computing a great option for the assessment of the proposed Agility index since all the SC partners around the world can provide their level of implementation of the deployed agile practices in the cloud and all the filled information can be treated in order to provide a clear overview of both the individual companies and the whole SC's Agility behaviour.

According to several authors and practitioners, cloud computing is an unavoidable path for SCM. Kefer [12] proposes three ways in which cloud computing can improve SC operations: a cloud solution can give real-time visibility to where a product is at any given time; moving to the cloud implies standardize the data from all the partners and define security rules; a cloud platform builds a collaborative community. Schramm et al. [21] signal several changes that adoption of cloud computing will drive into supply chains: new competitors; speed to market for new products and services; large-scale transformation. Wrigth [24] and Schramm et al. [21] pointed out some SC processes best suited to cloud computing: planning and forecasting; logistics; sourcing and procurement; service and spare parts management.

Notwithstanding these well-known advantages of performing the SCM in a cloud environment, to which one could add up its inherent environmental gains, the cloud can also be the enabling mean of a thorough SC agility assessment process. This can be viewed as a suite of integrated applications (processes) and tools that support a specific, major business capability or need – in a close agreement with the definition of a virtual business environment (VBE) presented in Iyer and Henderson [11] . Such VBE involves different processes that together can be seen as a living system as depicted in Fig. 5.1.

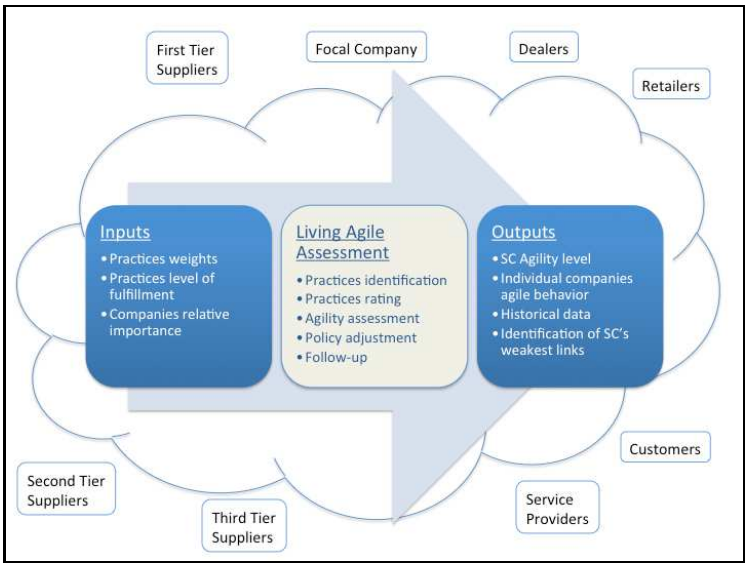

Fig. 5.1: Agility index assessment process for the SC

Despite that much of the organizational knowledge and expertise is scattered across the SC (even a single undifferentiated employee may contribute with the elicitation of an unforeseen key practice) one should expect that at the top of the institutional management hierarchy there are enough experts with a brighter wide enough vision able to capture all the relevant aspects of the agile practices. However the dimension of the supply

chain, the various sources of uncertainty and the specificities of the member enterprises as well as their complex interrelationships make the identification of practices a difficult task. Moreover a proper communication of the evaluation policies as well as the participation in their definition may promote the enthusiastic enrolment in their enforcement.

To build and enable this new collaborative infrastructure the managers can choose to deploy a cloud service promoting the discussion around a set of practices based on the ones presented in the previous section, allowing the introduction of new practices and the removal of inadequate ones. This common environment will likely allow the emergence of unsuspected collaborations between knowledge workers in different enterprises of the SC – in some cases sharing a common view on a particular set of practices, in others disagreeing or even fighting for opposite ones. The resulting group authorship, regardless of the intensity of the discussion (being moderated or not), has the advantage of being readable by anyone in the organization making the entire process of knowledge elicitation persistently visible. After a stabilization period it is expected that a homeostatic state is achieved through the assimilation and acceptance as legitimate of the ground rules for the assessment of the whole SC.

Regarding the practices' rating task several methods for defining the set of weights for the SC agile practices can be devised. The VBE can be enriched with some sort of polling user interface through which selected contributors can cast their perception of the importance of each practice. This can be simply an extension of a numeric Delphi study to a wider group of participants (some possibilities include experts and academics inside and outside organization with a mix of selected workers in key points of the SC).

The Agility index assessment can be an effective tool to be used not only by strategic decision makers, who need comprehensive models to support their decisions aimed to the enrichment of the SC, but also by the individual SC partners aiming at improving their added value in the SC. In this regard it is important that the entire evaluation be made simple, preferably in a familiar sound environment. The VBE should offer a controlled interface simplifying the innovation of applications and services as well as the control of their introduction. Frequent assessment loops (or even a continuous one) and location independence access to the differentiated evaluation assets should help to improve the overall SC performance by enabling a clear perception on what agile practices should be reinforced by each individual company.

Moreover, the Agility assessment process presents a dynamical open-system time-variant nature. The assessment panel composition can change (or simply their believes and knowledge about the assessment process), a new set of practices can be devised (by the inclusion of new practices, removal of obsolete ones or adaptation of their relative weights), the perceived relative importance of each individual company to the agility of the SC can change, or even the SC structure can suffer a rearrangement due to the inclusion/exclusion of some companies and to the establishment of new partnerships.

All these factors contribute to the need for frequent policy adjustments mediating successive agility assessment cycles. The flexibility and reusability offered by the VBE, allowing not only abandoning an environment and moving to the next but also the retrieval of a previously used environment at a future date, are key features for this task.

The follow-up task is also critical since the practical adhesion of the SC's enterprises and employees to the agile practices should be higher if they feel themselves as an integral part of the collective intelligence system that is being formed. To this aim it is essential that the individual components be aware of the current status of the evaluation process and can maintain track of the previous evaluations conditions and results. Moreover the usefulness of the VBE can be highly improved if complemented with a simulator of the evaluation process giving a greater feedback on the effective role of the SC partners, and his real impact on the agility index of the organization.

From a top management perspective the ubiquitous location independent access altogether with the addressability and traceability are major features of the deployed VBE. To have direct access to historical data of individual companies, to have an immediate glimpse of the evolution of the assessment process, or to possess in real time a benchmark of the different companies (actual and putative) are some of the examples of valuable assets for the decision-making activity.

6. Conclusion. The SC can be viewed as an open living, self-organizing system that has the ability to interact with its environment. Arguably the major effect of this interaction consists on the dynamic incorporation of new efficient agents and removal of non-agile ones. In this regard the capability to assess the agility level is a critical asset in maintaining a high fitness to a volatile environment. Conceptually the subsystems necessary to process information for the coordination, guidance and control of the agility assessment system can themselves

be seen as part of a living system that finds just the right environment in the cloud.

Cloud computing introduces a new business model where consumers can have access to hardware and software, through the Internet, in a pay-per-use manner as we do with public utilities. From a SC perspective cloud computing has transformed how global business networks interact, delivering a flexible, collaborative model. The establishment of a dedicated virtual business environment in such a common infrastructure offers a controlled interface simplifying not only the introduction of the proposed agility assessment model to the entire SC but also its validation and subsequent analysis of historical data.

The proposed approach supports the development of two agility indices: one to assess the individual company agile behaviour, and the other one to determine the same behaviour, but for the entire SC. Managers can use the proposed assessment model as a mean to adjust the organizations' behaviour according to the reached agility index score in order to improve the company efficiency. Moreover, it makes it possible to implement functional benchmarking approaches in the SC and to do a ranking among the companies, according to the agility index value. This serves as a motivation to companies try to reach better position among their partners and to be more rigorous in establishing priorities, targets and goals, in terms of agility.

REFERENCES

[1] A. Agarwal, R. Shankar, and M. Tiwari, *Modeling agility of supply chain*, Industrial Marketing Management, Vol. 36, No.4 (2007), 443–457.

[2] A. Aljabre, *Cloud Computing for Increased Business Value*, International Journal of Business and Social Science, Vol. 3, N 1 (2012), 234–239.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia *A view of cloud computing.* Communications of the ACM 53:4 (2010), 50–58.

[4] L. Camarinha-Matos, and H. Afsarmanesh, *Design of a virtual community infrastructure for elderly care*, in L. Camarinha-Matos, ed., Collaborative Business Ecosystems and Virtual Enterprises, Kluwer Academic Publishers, Boston, 2002.

[5] M. Christopher and D. Towill, *An integrated model for the design of agile supply chains*, International Journal of Physical Distribution & Logistics Management, 31:4 (2001), 235–246.

[6] J. Collin and D. Lorenzin, *Plan for supply chain agility at Nokia*, International Journal of Physical Distribution & Logistics Management, 36:6 (2006), 418–430.

[7] T. Goldsby, S. Griffis, and A. Roath, *Modeling lean, agile, and leagile supply chain strategies,* Journal of Business Logistics, Vol. 27, No. 1, (2006), 57–80.

[8] P. Gunasekaran and E. Tirtiroglu, *Performance measures and metrics in a supply chain environment*, International Journal of Operations & Production Management, 21:1/2 (2001), 71–87.

[9] V. Hoek, A. Harrison and M. Christopher, *Measuring Agile capabilities in the supply chain*, International Journal of Operations & Production Management, 21:1/2 (2001), 126–48.

[10] H. Ismail and H. Sharifi, *A balanced approach to building Agile supply chains*, International Journal of Physical Distribution & Logistics Management, 36:6 (2006), 431–444.

[11] B. Iyer and J. Henderson, *Preparing for the Future: Understanding the Seven Capabilities of Cloud*, Computing, MIS Quarterly Executive 9:2 (2010).

[12] G. Kefer, *Three Ways Cloud Computing Can Improve Supply Chain Operations for the Chemical Industry*, IHS Chemical Week, February 2012.

[13] K. Khan, A. Bakkappa, B. Metri, and B. Sahay, *Impact of agile supply chains' delivery practices on firms' performance: cluster analysis and validation*, Supply Chain Management: An International Journal, 14:1 (2009), 41–48.

[14] C. Lin, H. Chiu and P. Chu, *Agility index in the supply chain,* International Journal of Production Economics, Vol. 100, No. 2 (2006) , 285–299.

[15] H. Linstone, and M. Turoff, (eds) The Delphi Method: Techniques and applications, Addison-Wesley, 1975.

[16] P. Mell, and T. Grance, *The NIST Definition of Cloud Computing*, NIST - National Institute of Standards and Technology, September 2011, 7 pages.

[17] J. Mistry, *Supply chain management: a case study of an integrated lean and agile model*, Qualitative Research in Accounting & Management, 2:2 (2005), 193–215.

[18] I. Mohammed, R. Shankar and D. Banwet, *Creating flex-lean-agile value chain by outsourcing: An ISM-based interventional roadmap*, Business Process Management Journal, 14:3 (2008), 338–389.

[19] B. Naylor, M. Naim, and D. Berry, *Leagility: Integrating the Lean and Agile manufacturing paradigms in the total supply chain*, International Journal of Production Economics, 62:10 (1999), 107–118.

[20] G. Rowe, and G. Wright, *The Delphi technique as a forecasting tool: Issues and analysis*, International Journal of Forecasting, 15:4 (1999), 353–375.

[21] T. Schramm, S. Nogueira, and D. Jones, *Cloud computing and supply chain: A natural fit for the future*, Logistics Management Magazine, March 14, 2011.

[22] P. Swafford, S. Ghosh, and N. Murthy, *Achieving supply chain agility through IT integration and flexibility*, International Journal of Production Economics, 116:2 (2008), 288–297.

[23] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, *A break in the clouds: towards a cloud definition*, SIGCOMM Computer Communication Review, 39:1 (2009), 50–55.

[24] J. Wrigth, *An introduction to cloud computing in supply chain management*, Supply Chain Asia Knowledge, February 24,

2011.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 28, 2012
Accepted: Jan. 06, 2013

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 303–314. http://www.scpe.org
ISSN 1895-1767

A PANORAMA OF CLOUD SERVICES

DANA PETCU*

Abstract. Cloud computing paradigm has attracted a lot of attention in the last five years as coming during an economic crisis with an appealing offer in reducing the infrastructure and maintenance costs. After first wave of enthusiasm in adopting the concept, a clearer image has been formed about the benefits and limitations of Cloud computing and a lot of different supporting technologies were developed. As consequence a new threat is raised by the high number of the proprietary technologies that makes difficult the decision of the proper technological selection according to the real business needs. In this context the aim of this paper is to offer a snapshot on the current concepts and the available technologies, especially of the ones that can allow the development of a solid market of Cloud services and applications. A particular attention is given to the trend of federating and brokering Cloud services in the process of forming new markets. Moreover, we propose a classification of groups of services from multiple Clouds based on models similar to the ones used in computer graphics to express colors. Furthermore, a technological solution aligned to the market requirements is presented as case study, pointing also to the role of open-source codes for promoting the Cloud service usage on large scale.

Key words: Multi-Cloud, Cloud Federations, Cloud service markets

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction. The term of Cloud Computing has been coined one half decade ago to name a new approach of providing services via Internet. The Cloud term is not accidentally or trendy: it is related to the already classical form of representing of the Internet connections between their multiple end users. By this image it catches in a simple word a long-time expressed desire to see the connectivity, storage, processes or applications as utilities connected via the Internet (which becomes finally The Computer by incorporating in it these utilities).

The interest in the utility concept 'pay-as-you-go' for e-infrastructure services promoted by Cloud computing supporters has been amplified by the context of the economic crisis, creating the illusion that the future is Cloudy. The hype of Cloud computing sustained by the big companies has rapidly created an ad-hoc market of services that are quite diverse due to several reasons, like different understanding of the concepts, complexity of the underlying software stacks, or need to promote earlier legacy software that are able to support the new concepts.

The technologies and services that are supporting the previous described concepts have been developed into a very large and fast evolving pool of Cloud computing offers, creating an ad-hoc e-market in which the main actors, developers, providers and end-users have clear roles. The providers are offering new services that are allows them to create a benefit from sharing their un-spend e-infrastructure resources. The users are primarily interested in the high availability, reliability and ubiquity of the services. The developers are interested to enlarge the base of end-users of their products. Unfortunately this market is driven by the providers and developers needs and end-users are struggling with the diversity of the concept approaches and the lack of uniformity in what concerns the interfaces or protocols (leading to a vendor lock-in).

In this context we consider useful to start the next section with a light presentation of the terminology currently used in Cloud computing and to point towards one particular problem emerging from the diversity of the current Cloud service offers (vendor lock-in). Section 3 is devoted to the emerging Federations and Markets of Clouds. We propose a classification of different views on these groups of services using models similar with the ones used in computer graphics. Moreover, we identify the main problems to be solved in order to build Federations and Market of Clouds. A particular example of middleware supporting the Federations and Markets of Clouds is exposed in Section 4. The last section is dedicated to the conclusions and future expectations.

2. Overview of the Cloud services' offers. The aim of this section is to provide an overview of the categories of services that are currently offered and to point towards the limitations of the market offer. A special attention is provided to the open-source as solution for the vendor-lock in problem and the need of the free movement in the market of Cloud services.

2.1. Basic terminology. Despite the interest in the new concept, the definition of what Cloud Computing is still not generally accepted, and its borders and relationship with other distributed computing paradigms are still discussed. We consider here only two definitions of well known authorities: Expert Group of European

*Institute e-Austria Timişoara and West University of Timişoara, Romania, (petcu@info.uvt.ro).

Commission on Cloud Computing [28] and NIST [8]. The definition of the Expert Group is focusing at Clouds as execution environments, concluding that an environment can be called Cloudified, if it enables a large dynamic number of users to access and share the same resource types, respectively service, whereby maintaining resource utilisation and costs by dynamically reacting to changes in environmental conditions, such as load, number of users, size of data. In the NIST definition, the Cloud is seen as a model: 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.

Pros and contras for these definitions can be formulated by different stakeholders, like services providers, developers or end-users as they views can be different, and such debate is not subject of this paper. Important are the main characteristics of the Clouds, for which an agreement is close to be reached. According to the above mentioned NIST report, the essential characteristics are: (a) on-demand self-service; (b) broad network access; (c) resource pooling with multi-tenancy; (d) rapid elasticity; (e) measured service. These characteristics are perceived mainly from a user perspective. They are further split and re-grouped into two categories by the Expert Group, from a provider and developer point of view:

1. intrinsic characteristics, specific to the Clouds: like elasticity; multi-tenancy, high availability, and automated management;
2. extrinsic characteristics, that are extended or inherited from parental domains of utility computing, service architectures, or general IT: like virtualization, pay-per-use, market mechanism (from utility computing), resource management, metering (from service architectures), tool support, programming, or data management (from general IT).

One of the advances that Cloud computing is bringing and is not pointed in the above mentioned documents, but which we consider highly relevant for this paper is the implementation of the idea of programmable e-infrastructures. Using the programming tools available to the developers of applications, e-infrastructure services can be allocated, de-allocated or configured. This is a big step forwards to the self-adaptability of the execution environments as well as agility at the level of applications.

2.2. Classification of the services and tools. Despite the controversial disputes on the Cloud definition, there is an almost well-established consensus on the delivery and deployment models in Cloud computing. We remind them in what follows for the sake of continuous flow of presentation, after which we present other controversial or new classifications.

One of the basic concepts in Cloud computing is the delivery as-a-Service. The delivery is done using the Internet protocols and standards. Three main categories of service models (or delivery models) are recognized (e.g. in NIST report by [8]):

1. Infrastructure as a Service (IaaS) a consumer can get service from a full computer infrastructure through the Internet (Internet-based services such as storage and databases are typically considered a part of the IaaS).
2. Platforms as a Service (PaaS) offers full or partial application development environments.
3. Software as a Service (SaaS) provides a complete turnkey application via the Internet.

IaaS delivers a computing hardware infrastructure over the Internet and is enabled to split, assign and dynamically resize these resources to build custom infrastructures, just as demanded by customers. What makes the Cloud a novelty is the self-management capabilities it offers, the possibility of an almost immediate resizing of the assigned resources, and the application of the pay-per-use revenue model.

PaaS offers an additional abstraction level: rather than supplying a virtualised hardware infrastructure, they provide the software platform where customer services run on. Sizing of the hardware resources demanded by the execution of the user services is made by the PaaS provider in a manner transparent to the user. IaaS and PaaS systems have in common their aim to be a platform for their users.

SaaS, in contrast, groups together Cloud systems in order to create a final aggregated service itself. These services are software products that can be in the interest of a wide variety of users.

Many other resources can also be offered as Cloud services, such as Storage as a Service, Messaging as a Service, Network as a Service, Data as a Service, Communication as a Service, Database as a Service, Information as a Service, Process as a Service, Application as a Service, Integration as a Service, Security as a Service, Management as a Service, Testing as a Service etc. They are usually just particular types of one of the three groups of delivery models mentioned above (IaaS, PaaS, and SaaS).

A particular attention is given recently to the emerging technologies for a new model of Business-Processes-as-a-Service (BPaaS). Early implementations of this concept are no earlier than two years ago [23]. It was proposed by [30] to set BPaaS to the same level as the other three models, instead in SaaS category, due to the impact that can have on the business community.

The basic deployment models are the Private and the Public Clouds. In a Private Cloud the services are provisioned for the use of a single organization with multiple known members and these services are relying on- or off- premise e-infrastructures. A Public Cloud is designed to serve a general public, the owner of the resources (hardware and software) being the Cloud provider. Between the two models is the Community Cloud that serves two or more organizations that have agreed about the membership, security, mission and other common concerns, and can comprise one of more Private Cloud installations. The combination of the two or more Clouds bound only by technologies that are enabling data and application portability are classified by NIST report as Hybrid Clouds.

Several taxonomies and ontologies were already published to differentiate the Cloud concepts and terms and their inter-relationships and with the particular terms that are used by different providers. An example of such ontology can be found in the book chapter by [20].

Without willing to complicate the existing classifications, for the purpose of this paper and its understanding, we consider that the Cloud technologies and tools that are available to support the above described models should be split into two main categories: hosted services and deployable services. A hosted service is an integration of hardware and software exposed as a service compliant with the Cloud characteristics on a wide area network by a certain organization. A deployable service is a software that includes an interface of a service compliant with the Cloud characteristics and that is installable on certain e-infrastructures and is. Deployable services can be used to build hosted services residing on- or of-premises e-infrastructures.

Already classical examples can be used as examples to distinguish between the two categories. At the IaaS level, Amazon EC2 is a hosted service, while Eucalyptus is a deployable one (managing virtual machines); Amazon S3 is a hosted service, while Riak is a deployable one (for key value store); Amazon SQS is a hosted service, while RabbitMQ is a deployable one (for message queues). At the PaaS level, Google App Engine is a hosted service, while VMWare CloudFoundry is a deployable service. At SaaS level, Google Mail is a hosted service, while VMWare Zimbra is a deployable service.

The hosted services are the most common services and therefore the common understanding of Cloud services is referring to this group. The way in which their interfaces are conceived is very convenient for the application developers as they are hiding complex processes and heterogeneous resources. On another hand, the variety of the design of the interfaces of the hosted services creates a dependence between the application that is developed and the Cloud for which is developed (the vendor-lock-in problem) and hinders the interoperability between multiple Cloud services (the interoperability problem).

The deployable services have the potential to overcome the vendor-lock in and interoperability problems if they are adopted by several Cloud providers on a wide scale. Moreover, most of them are offered as open source, so that the developer community can help their improvement and adaptation to their or community needs.

2.3. The vendor lock-in problem. The heterogeneity of multiple Clouds is reflected in the variety of services offered by various Cloud providers, their interfaces, as well as in the variety of the hardware and software stacks that are used. The developers of Cloud-aware applications are facing a big problem in selecting the proper Cloud services that are matching their application needs.

On another hand, the usage of services from multiple Clouds has been introduced first with the idea of Hybrid Clouds, when Private Clouds are combined with the Public Clouds. The outages of Public Clouds have brought into discussions the migration of the applications and data from one Cloud to another. Moreover, small Cloud providers who have emerged recently are facing the problem of limited resources and they are interested to make agreements to other providers to support scalability beyond their resources in peak cases. These scenarios are more often discussed in the latest years in conjunction with a technical problem that has arrived with the increase the number of Cloud providers: vendor lock-in.

The reasons of vendor lock-in are various: proprietary APIs of the services, lack of accepted standards, particular services that are subject of high investments and so on. The problem is not due to the vendor will, but instead is a reflection of the large set of hardware and software stacks needed to build a Cloud service. Heterogeneity is encountered to both low and high levels, from virtualization technologies, to programming environments. It is expected that the middleware provided by the Cloud provider or even meta-providers like

Cloud brokers are hiding this heterogeneity. If this is happening at the Cloud provider level to a certain level (at least from the point of the view of the users), the meta-level is still lacking break-through offers beyond the research prototypes. Therefore we considered useful to identify which are the challenges in building middleware to deal with heterogeneity between Clouds.

3. Dealing with the multiple Cloud services. This section intends to make a survey of the solutions that are involving multiple Cloud services. The next sub-section is discussing the different concepts that were considered in the context of the meta-level of multiple Clouds. The second subsection is introducing a new classification method. The third subsection is devoted to the challenges associated with two specific meta-levels, Federations and Markets.

3.1. How to Name Each Case of Grouping Services from Multiple Cloud?. In this sub-section we present an overview of various approaches to nominate the multiple Cloud usage scenarios.First we should remind that the meta-computing idea was coined almost two decades ago to point the idea of grouping several computing e-infrastructures, and the idea of grouping Clouds has several commonalities with the meta-computing.

NIST recent report [8] has divide the usage scenarios in two categories, according to the number of Clouds involved at a moment of time: sequential, when Clouds services from different providers are used one after another, or simultaneously, when Cloud services from different providers are used in the same time. Sequential is encountered in the case of migration from one Cloud to another, in the case of the selection of the service at deployment (contrary to the selection at the design time, scenario in which only one Cloud is involved), or when interfaces for software and data transfer are build between Cloud providers in agreement between them. Migration can be required from various reasons, like changing to adapt to resource availability, to the resource cost or to adapt to the changes in application requirements (like emerging deadlines). Simultaneous usage is often encountered in the Hybrid Clouds, when parts of the applications are residing on-premise resources (Private Cloud) and parts on Public Cloud resources.

InterCloud term was introduced by [1] in analogy with the Internet and based on a similar vision: to connect individual Cloud infrastructures and giving control to the users. The initial term has supposed a certain agreement between Clouds in what concerns the interfaces. Another term that is used is Cloud-of-Clouds as analogy with the Grid which is a Cluster-of-Clusters. Other terms like Cross-Cloud or Sky Computing [11] have introduce the brokerage of Cloud services.

A two-level classification is provided by [4] where the multiple Clouds scenarios are split in two, in another dimension, according to the software stacks: Horizontal Federations when Cloud providers are federate for scale and capacity enlargement reasons; and Vertical Supply Chain when Cloud providers are leveraging services from other providers.

In the paper by [6] the multiple Cloud usage scenarios are split in three cases: (a) Bursting Private Clouds (expansion of Private towards Public ones); (b) Federated Clouds (partnership); (c) Multi-Clouds (providers working with external services).

According the article of [19], the coupling between the acquired resources is considered as criteria to split the multiple Cloud usage scenarios: (a) loosely coupled federation in which the inter-operation is low (monitoring is limited, no control on external resources, no migration of virtual machines); (b) partially coupled federation, when an agreement between the providers has been established concerning different issues like interchanging monitoring information, virtual networks across Cloud boundaries, or control over remote resources; (c) *tightly coupled federation*, when the agreement allows full control on remote resources and their monitoring, creation of cross-site networks or virtual storage across Cloud boundaries.

In the same paper [19] the authors are discussing four potential architectures of the frameworks supporting the multiple Cloud usage scenarios: (1) Hybrid Cloud (Cloud bursting), coupling on-premise infrastructure with remote resources from Public Clouds (in the loosely coupled category); (2) Cloud broker with a broker that serves users and has access to several Public Clouds (loosely coupled); (3) Aggregated Clouds when several providers aggregate their resources (partially coupled); (4) Multi-tier Clouds when a hierarchical agreements are established so that a Cloud provider has full control over the resources of different Cloud sites (tightly coupled).

Browsing the literature, we see that the above terms are often used with different meanings, therefore we considered necessary to propose a classification scheme that tries to cover as much as possible the above described cases.

Without any intention to complicate the image, but for reasons exposed in the next section, in this paper we introduce another keyword, namely Market of Clouds. This market is expected to provide a single interface for

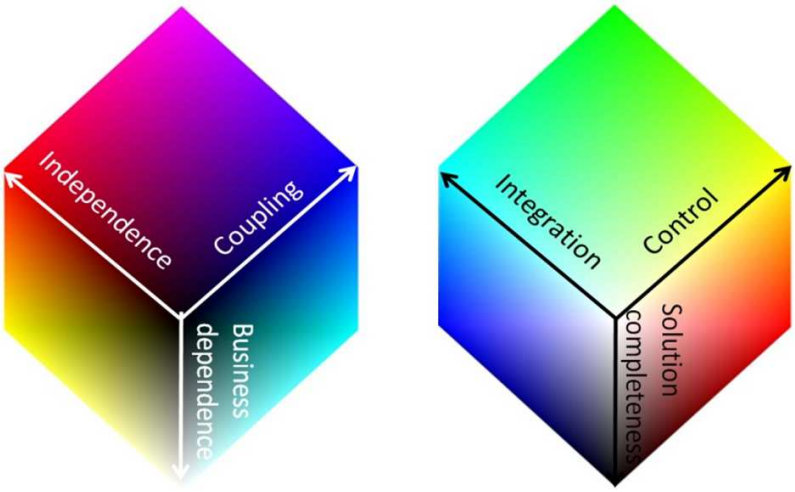

Fig. 3.1: Users' requirements in RGB model vs. Providers' requirements in CMY model

a consumer to address resources from multiple Clouds. The key element of a market is the broker mechanism, which operates outside of the Clouds, monitors the connected Clouds, detects their failures and react to them in order to comply with the clients requests by having the permission to move virtual machines, appliances, applications or data from one Cloud to another. While Federations of Clouds can be similar with Grids, Markets can be seen as following the Web services concepts.

3.2. Colors of user and providers. In what follows we propose a classification of the cases of grouping services from multiple Cloud. We use an analogy with the basic color models from graphics: Red-Green-Blue (RGB), used for example by displays, and Cyan-Magenta-Yellow (CYM), used for example by printers.

We consider that in RBG model (Fig. 3.1) we have on the axes:

x axis (Red): the degree of the independence from the Cloud provider. Zero is associated with the Federation since the user is addressing one Cloud and this Cloud is deals with the multiple Cloud services. One is associated with the Market which allows the user to select the Cloud.

y axis (Green): the degree in which new business are build. Zero is associated with the Horizontal Federation or one Cloud. One is associated with the case of Vertical Supply Chain.

z axis (Blue): the degree in which the coupling is done between Cloud services. Zero is associated with lack of coupling. One means a tight coupling.

The origin of the RGB-like system is black corresponding to the Horizontal Federation with no coupling (collection of isolated Clouds). Hybrid Clouds are red: (1,0,0). Brokers are yellow: (1,1,0). Aggregated services are examples of points in one axes plane. Multi-tier Clouds are cyan: (0,1,1). The 'maximum' of all values, (1,1,1), is represented by Market of tightly coupled services allowing vertical supply chains. This is an expression of the desire of the Cloud users; therefore we consider that the RGB representation reflects their wishes with black being the worst case.

We consider that in the CMY model (complementary to the RGB model) we have the opposite:

x axis (Cyan): opposite to the x axis from RGB model, expresses the integration degree with other Clouds, zero being the Market, and one, the Federation;

y axis (Magenta): opposite to the y axis from RGB model, expresses the completeness of the solutions offered by a certain offer, at zero being the Horizontal Federation or the single Cloud, and at one the Vertical Supply Chain;

z axis (Yellow): opposite to the z axis from RGB model, expresses the control over own resources in a Federation or Market, at zero being the full control, and at one being full controlled.

The origin of the CMY-like model is white, corresponding to the above mentioned 'maximum' wish of the users. The black is the 'maximum' of all values; in this case represents the Horizontal Federation in which loosely coupled services are offered. This maximum can reflect the wish of the providers to have full control

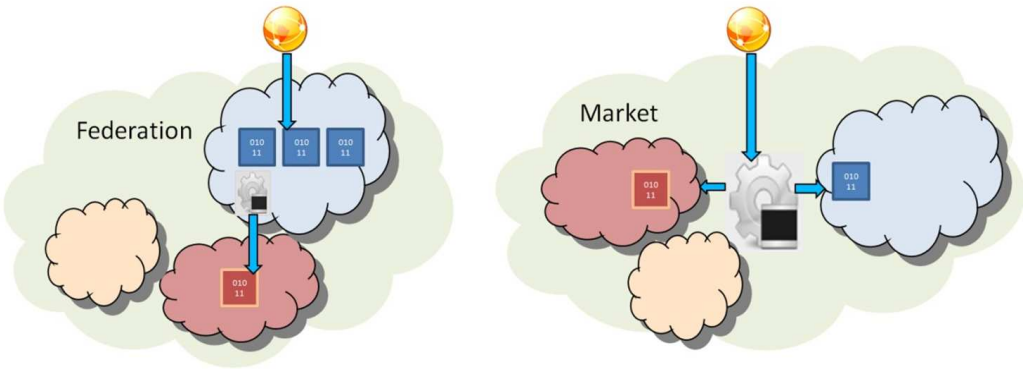

Fig. 3.2: Federation vs. Market of Clouds (the application is represented by the ball, the services rented in the Cloud are represented by boxes)

on their resources, to satisfy all the requirements of the users, but being in a Federation with other providers without high obligations (or even alone, one Cloud).

The fact that the two models are opposite to each other reflects also the current status of the offer of the Cloud service markets. The user expectations are not always in agreement with the offers. The consensus and a market equilibrium can be probably found where the colors of the parties are identical: in the middle of the cubes, at a middle gray.

3.3. Challenges in Federation of Clouds and Markets of Clouds. The implementation of middleware supporting multiple Clouds is not trivial. One reason is the fact that the existing models for interoperability and orchestration of the services are designed for static environments, while Clouds are typically dynamic and even in a Federation agreements among the Cloud providers are dynamically established.

In this section we will analyze just one dimension - the red one from the previous RGB model in an effort to identify the research and development issues that are making the complete middleware for Federations and Markets still unavailable.

We remind that the two cases are different in the way they are treating the user and the cooperation between the Clouds. Figure 3.2 is trying to express this difference in a graphical way.

The main problems identified until now in the middleware developments for Federations of Clouds are the followings:

1. Supported by the interoperability inside the Federation, there is a need for a component placed at the Cloud provider site (similar to a broker, named here manager) allowing the match-making with available external services and authentication procedures for these external services.
2. Live virtual machine migration should be coupled with load balancing to increase power efficiency. Problems that should be surmounted are for example related to the migration beyond the network boundaries without losing the already established network connections and storage of virtual machines on shared file systems on large scale.
3. An interoperability framework based of common understanding of Cloud providers on the main terms that are used is a key element of an efficient Federation. We have identified recently the interoperability and portability issues [24].
4. Cloud computing providers are limiting the connectivity of the virtual machine and their network traffic. Network overlay technologies can be used as a solution to overcome these limitations.

Table 3.1 is presenting some examples of current prototypes that are implementing some innovative solutions to these problems.

Several other issues have been treated until now only at theoretical level:

– Meta-schedulers are needed to support a coordinated distribution of different Clouds workloads. The process is slowed also due to the fact that several Clouds do not support scalable load balancing.
– The scheduling in such environments is a complex decision mainly due to their dynamics: resources behaviour

Table 3.1: Issues in Federation of Clouds and available solutions

Subject	Examples of prototypes
Federation manager supporting the external services selection	CCFM [4] is a Cross-Cloud Federation Manager with discovery, match-making and authentication features
	ORCA [17] enables computational and network resources from multiple clouds and network substrates to be aggregated into a single virtual resource
Live migration of virtual machines	Shrinker [27] is a modification of KVM hypervisor based on the detection of inter-virtual-machines data similarities
Interoperability frameworks	PSIF [15] models and tries to resolve semantic interoperability conflicts raised during the deployment or the migration of an application between PaaSs
Network virtualization techniques for distributed resources in different administrative domains	TinyViNe [32] for Nimbus installations with MPI jobs benchmarks. Other solutions are presented in the same paper

is unpredictable, local schedulers should interact with each other, resource sharing is based on service level agreements that can be changed dynamically. A review of approaches at theoretical level is provided by [31].

- The level of integration of different security technologies should permit a new provider to join the Federation without changing his security policies or authorisation processes. Moreover, the user already authorized to a certain Cloud should be able part of the resources of the resources (like in Grids).
- A monitoring meta-system, hopefully independent from any provider solution, of the resources of different providers from the Federation should be designed and developed.
- Automated operations should use intelligent management systems. An approach using rule-based techniques was proposed by [12]. Currently Cloud providers must manually resolve sub-optimal configurations, and maintain an on-going balance between capacity utilization, cost, and service quality [3]. Self-adaptability to the changes of each provider service availability is strictly necessary in the near future.
- Integration-as-a-Service is a way to abstract the technical details and the interaction with the cloud services and to provide a way to treat these interactions as part of the abstract description of a Cloud-based solution. Referring to this idea, the Cloud Blueprinting, introduced in by [21], includes a detailed deployment plan of an applications and a high-order packaged integration solution that provides a description of the integration needs for the interaction between Cloud services provided by different providers.

In the case of Markets of Clouds, the main issues for research and development are related to the followings:

1. Brokers are acting as intermediaries between providers and clients, being able to allocate resources among multiple Cloud offers. A broker assists the clients in selecting the appropriate service that best suits (using several criteria) their requirements and needs. Potentially, the request is split by the broker such that different providers receive sub-requests for provisioning or instantiation of resources. The broker should provide a single entry point for a specific market and, in the best case, a delegation mechanism in what concerns the user credentials should be part of the broker and the output of the brokering process should be the allocation of the proper resources. An overview of the requirements for a broker is provided by [18]. Note that beyond the research prototypes enumerated in Table 3.2 there are already commercial offers (like Rightscale's Multi-Cloud Engine that is able to broker capabilities related to virtual machine placement in several Public Clouds) or in-production research prototypes like OpenCirrus.
2. Using the same APIs the dynamic allocation of the resources and binding components of the applications to the acquired resources should be possible.
3. Search engines with matching algorithms and based on semantic technologies are needed; several user requirements should be supported (functional and non-functional ones).
4. The diversity of services complicated the service selection. A methodology to compare Cloud service

Table 3.2: Issues in Markets of Clouds and available prototypes

Subject	Examples of prototypes
Brokers	Cloudbus [2] uses several brokers which are interacting with a coordinator
	Zeel/i [7] allows single-sign (using the Cloud credentials of the Zeel/i) and the selection of Cloud resources according to specific requirements
	Extension of OpenFlow [9] which is solving the selection problem expressed as mixed integer program
	SORMA [22] use bidders and sellers to represent the beneficiaries of the brokering system
	SERA [5] is using a multi-agent system with agents representing the beneficiaries of the brokering system augment with special duties of scheduling or controlling the resources or monitoring, registering or recovery.
Uniform APIs	SAGA [16] dynamically allocate resources via a job interface and bind sub-jobs to these resources
Search engines	Cloudle [10] is a Cloud service search engine based on a specific Cloud ontology.
Benchmarks	CloudCmp [14] is a set of benchmarking tools for comparing services from elasticity, persistence of storage, intra-cloud and WAN communications.

based on multiple criteria is expected. Comparison criteria can vary from cost, policies, performance and so on. For the performance measurements independent observer services need to be built.

Other problems that should be solved in Markets of Clouds are related to:

- Automated approaches for deploying virtual appliances are expected to emerge.
- Deployment description languages should target application run-time aspects. Key requirements are stated by [13].
- Service aggregator that combines services from different Clouds, including dashboards or smashups should be build.
- Expert systems for recommending systems are expected to emerge. Artificial intelligence techniques, from reasoners to evolutionary computing or even multi-agent systems can found interesting applications in this field.
- Multi-Cloud governance - high level management. The paper [29] are proposing autonomic approach based on a governance model where a high-level manager dynamically adapts the behaviors of the low-level managers by fine-tuning their policies.
- Portability in this context is the ability to migrate applications between different Clouds (subject of the next section). Standardized and open interfaces and protocols to manage Cloud services are required.

4. Case study of a support for Markets of Clouds: mOSAIC, an open-source Platform-as-a-Service. We have recently contributed to the development of the open-source platform-as-a-service named mOSAIC. It is designed to allow the portability of applications on top of different infrastructure-as-a-services. The applications are expected to be built from components and to communicate via a message passing system. An event-driven approach should be adopted when dealing with the Cloud resources that are interfaces through the vendor-independent API.

mOSAIC (Open source API and Platform for Multiple Clouds) is developed in the frame of a multi-national collaborative project funded by the European Commission in the period 2010-2013, and it involves more than forty persons, software engineers and programmers, as well as application designer and developers.

The open-source middleware, currently in a stable version, is deployable and available at https://bitbucket.org/mosaic. Details about the proposed API can be found in the article proposed by [25], in on-line documentations (http://developers.mosaic-cloud.eu) or demos (YouTube, key-phrase mOSAIC Cloud computing), and the project site (http://www.mosaic-cloud.eu).

mOSAIC system has a complex architecture (Figure 3) that includes:

(a) core platform services: from scheduler, load balancer, deployer, provisioner, scaler, monitor, component discoverer, specific virtual appliances, and so on, that are independent from the Cloud services;
(b) market services: broker based on multi-agent technologies, service discoverer, semantic engine based on Cloud ontology to match the functionality of the system and the services with the user requirements,

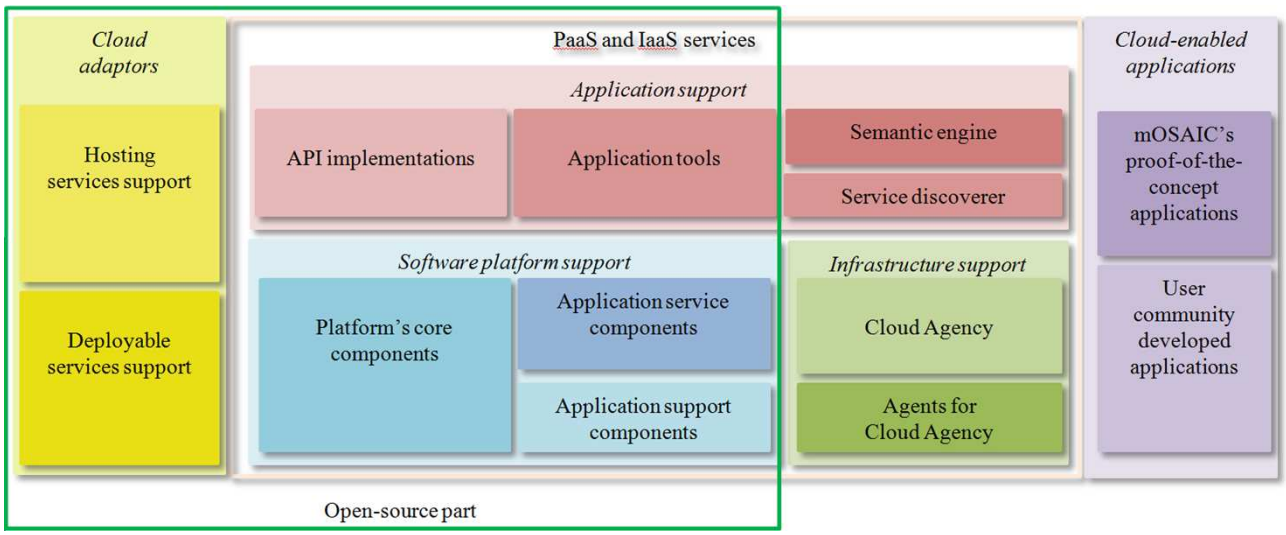

Fig. 4.1: General overview of the component architecture of mOSAIC solution

as well as service-level-agreements negotiation mechanisms;

(c) Cloud connectors and agents: to Public Cloud services as well as support to deploy open-source Cloud technologies (from more than ten providers);

(d) application development support: from a Desktop Cloud allowing the debugging of the developed application on desktops, to web interfaces of the platform to control the life-cycle of the components.

Similar efforts in providing open-source PaaS are undertaken currently by companies like VMWare (Cloud Foundry) or RedHat (OpenShift). While these are oriented mainly to web applications, mOSAIC intends to support also other types of applications, like scientific ones, or even business processes.

In this section we present the compliance of mOSAIC features with the ones requested for Federations and Market of Clouds, which we identified in the previous section. Table 4.1 synthesizes the results of this analysis. We consider that the criteria exposed in Table 4.1 can be further used to compare several solutions for Federations or Markets of Clouds.

5. Following the trends. We presented in this paper a particular view on the status of Cloud services and the current efforts to build Federations and Markets of Clouds expected to be the next step in the development of the Cloud services. We have also pointed towards to the potential support for building Markets of Clouds coming from a new open-source and deployable platform as a service, namely mOSAIC. Moreover, we consider that the usage in Cloud computing of open-source software can be a signal for a level of maturity of the Cloud technologies. When the diversity of the open-source stack will be reached the Cloud will expand beyond its current limitations, like the vendor lock-in problem.

The lessons learned in the development of mOSAIC embrace several topics. We mention here only few general ones. The diversity of the Cloud services has reached a certain degree to which the finding a common denominator is almost impossible and therefore the proposal of new standards in the field should be complemented by frameworks that are leaving the door opens for innovation in a competitive market. The availability of deployable solutions enables the fast development of new technologies and the most mature ones can be considered embeddable and trustable bricks in building a solid platform for building applications consuming Cloud services. The degree of automatisation that is expected from any group of Cloud services is hardly faced by the current mechanisms (like schedulers, auto-scalers, resource provisioners and so on) and the development of new solutions tailored for the case of Federations and Markets of Clouds are needed.

Several European collaborative projects, partially funded by the European Commission, and involving tens of research and development teams of Cloud technologies as well as users of Cloud services, are currently working to realize the vision of Federation of Markets of Clouds. We remind here only few on them, beyond the one already mentioned, mOSAIC (details about these projects and other similar ones can be found at least in the book [26]). Contrail (www.contrail-project.eu) is providing a solution for the Federation of Clouds. TClouds

Table 4.1: Compliance of mOSAIC with the requirements of Federations or Markets

Subject	mOSAIC solution
Federation manager/Broker	The Cloud agency augmented with the broker, vendor agents and the SLA mechanism are ensuring the selection of one or more services that are satisfying the requirements of the application (based on an application descriptor, the final result being the provisioning of the resources)
Live migration of VMs	Not supported at the level of the platform. But live migration of application components (not encountered in other middlewares), yes.
Interoperability framework	The semantic engine assist the developer of the applications to find the right functionality of the API and the Cloud services, based on a Cloud ontology.
Network virtualization techniques	A naming service was designed based on DNS service extensions
Uniform APIs	The APIs are vendor-independent
Search engines	Under development, architecture and services already established
Benchmarks	The benchmark framework allows to setup a custom benchmark which measures the performances of the target application under well known workloads
Meta-schedulers	Based on genetic algorithms for multi-criteria optimizations
Integration of security technologies	Credential service and an Intrusion-detection-as-a-Service
Monitoring meta-system	Not supported
Automated operations	Scaler and scheduler based on agent technologies and rules. Self-adaptability is in research phase
Integration-as-a-Service	The platform uses application descriptors, call for proposals (of resources) and application deployment descriptors that are matching the Cloud blueprinting idea
Deployment description languages	The above mentioned descriptors are described in a kind of XML based language
Automated deployment of VA	Virtual appliances are prepared on the fly (virtual machines with the platform controllers and deployable Cloud technologies) and deployed
Service aggregator	Aggregator should be part of the deployed application. Aggregation at the platform level is resumed to the component discovery mechanisms
Recommending system	Not supported
Portability	Possible if the component-based applications are compliant with the rules related to communications, architectural style (event-driven) and programming languages (currently Java and Python)
Multi-Cloud governance	Under development, architecture and services already established

(www.tclouds-project.eu) is offering security, privacy and resilience mechanisms for Federations and Markets of Clouds. 4CaaSt (4CaaSt.morfeo-project.eu) is proposing a BluePrint for registering the Cloud services in an e-Market. Optimis (www.optimis-project.eu) is providing brokerage mechanisms. Innovative technologies that are enabling the design of Federation and Markets are developed in the frame of: Vision Cloud (www.visioncloud.eu) which is looking in details to the issues of data management in Clouds; Cloud4SOA (www.cloud4soa.eu) which is dealing with semantic based interoperability at platform level; Remics (www.remics.eu) which is dealing with migration of legacy applications to Clouds; Cloud-TM (www.cloudtm.eu) proposing a new programming paradigm for Clouds. The integration of the research results in the production lines of the commercial partners of these projects are expected to happen in a range of two-there years.

Acknowledgments. This work was partially supported by the grant of the European Commission FP7-ICT- 2009-5-256910 (mOSAIC) for the Sections 4-5 and Romanian National Authority for Scientific Research, CNCS UEFISCDI, PN-II-ID-PCE-2011-3-0260 (AMICAS) for Sections 1-3.

REFERENCES

[1] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, M. Morrow, *Blueprint for the Intercloud - protocols and formats for cloud computing interoperability*, Procs. ICIW '09 (2009), pp. 328–336.
[2] R. Buyya, R. Ranjan, R. Calheiros, *Intercloud: utility-oriented federation of cloud computing environments for scaling of application services*, LNCS 6081 (2010), pp. 13–31.
[3] H. Cai, K. Zhang, M. Wang, J. Li, L. Sun, X. Mao, *Customer centric cloud service model and a case study on commerce as a service*, Procs. IEEE Cloud'09 (2009), pp. 57–64.
[4] A. Celesti, F. Tusa, M. Villari, A. Puliafito *How to enhance cloud architectures to enable cross-federation*, Procs. 3rd IEEE Cloud (2010), pp. 337–345.
[5] J. Ejarque, R. Sirvent, R. Badia, *A multi-agent approach for semantic resource allocation*, Procs. 2nd CloudCom (2010), pp. 335–342.
[6] E. Elmroth, J. Tordsson, F. Hernandez, A. Ali-Eldin, P. Svard, M. Sedaghat, W. Li, *Self-management challenges for multi-cloud architectures*, LNCS 6994 (2011), pp. 38–49.
[7] T. Harmer, P. Wright, C. Cunningham, R. Perrott, *Provider-independent use of the cloud*, Procs. Euro-Par'09 (2009), LNCS 5704, pp. 454–465.
[8] M. Hogan, F. Liu, A. Sokol, J. Tong, *Nist Cloud computing standards roadmap-version 1.0*, Special Publication 500-291 (2011).
[9] I. Houidi, M. Mechtri, W. Louati, D. Zeghlache, *Cloud service delivery across multiple cloud platforms*, Procs. SCC'11 (2011), pp. 741–742.
[10] J. Kang, K.M. Sim, *Cloudle: a multi-criteria cloud service search engine*, Procs. APSCC '10 (2010), pp. 339–346.
[11] K. Keahey, M. Tsugawa, A. Matsunaga, J. Fortes, *Sky computing*, Internet Computing 13 (5) (2009), pp. 43–51.
[12] G. Kecskemeti, M. Maurer, I. Brandic, A. Kertesz, Z. Nemeth, S. Dustdar, *Facilitating self-adaptable inter-cloud management*, Procs. PDP '12 (2012), pp. 575–582.
[13] A. Lenk, C. Danschel, M. Klems, D. Bermbach, T. Kurze, *Requirements for an iaas deployment language in federated clouds*, Procs. IEEE SOCA'11 (2011), pp. 1–4.
[14] A. Li, X. Yang, S. Kandula, M. Zhang, *CloudCmp: comparing public cloud providers*, Procs. 10th Conf. Internet measurement (2010), pp. 1–14.
[15] N. Loutas, E. Kamateri, K. Tarabanis, *A semantic interoperability framework for cloudplatform as a service*, Procs. 3rd IEEE CloudCom (2011), pp. 380–387.
[16] A. Luckow, L. Lacinski, S. Jha, *SAGA bigjob: an extensible and interoperable pilot-job abstraction for distributed applications and systems*, Procs. CCGRID'10 (2010), pp. 135–144.
[17] A. Mandal, Y. Xin, I. Baldine, P. Ruth, C. Heerman, J. Chase, V. Orlikowski, A. Yumerefendi, *Provisioning and evaluating multi-domain networked clouds for hadoop-based applications*, Procs. 3rd IEEE CloudCom (2011), pp. 690–697.
[18] H. Mearns, J. Leaney, A. Parakhine, J. Debenham, D. Verchere, *An autonomic open marketplace for inter-cloud service management*, Procs. UCC'11 (2011), pp. 186–193.
[19] R. Moreno-Vozmediano, R. Montero, I. Llorente, *IaaS cloud architecture: from virtualized data centers to federated cloud infrastructures*, Computer 99 (2012).
[20] F. Moscato, R. Aversa, B. Di Martino, D. Petcu, M. Rak, S. Venticinque, *An Ontology for the Cloud in mOSAIC*, In Wang L. et al, Cloud Computing: Methodology, Systems, and Applications, CRC Press (2011), pp. 467–486.
[21] D.K. Nguyen, F. Lelli, Y. Taher, M. Parkin, M.P. Papazoglou, W.J. van den Heuvel, *Blueprint template support for engineering cloud-based services*, LNCS 6994 (2011), pp. 26–37.
[22] J. Nimis, A. Anandasivam, N. Borissov, G. Smith, D. Neumann, N. Wirstrom, E. Rosenberg, M. Villa, Sorma - business cases for and open grid market: Concept and implementation, LNCS 5206 (2008), pp. 173–184.
[23] M. Pathirage, S. Perera, I. Kumara, S. Weerawarana, *A Multi-tenant Architecture for Business Process Executions*, Procs. ICWS'11 (2011), pp. 121–128.
[24] D. Petcu, *Portability and interoperability between clouds: challenges and case study*, LNCS 6994 (2011), pp. 62–74.
[25] D. Petcu, G. Macariu, S. Panica, C. Craciun Portable cloud applications - from theory to practice, Future Generation Computer Systems, doi: 10.1016/j.future. 2012.01.009 (2012).

[26] D. Petcu, J.L. Vazquez-Poletti (eds.) *European Research Activities in Cloud Computing*, Cambridge Scholars Publishing, UK (2012).
[27] P. Riteau, *Building dynamic computing infrastructures over distributed clouds*, Procs. IPDPS'11 (2011), pp. 2097–2100.
[28] L. Schubert, K. Jeffery (eds.), *Advances in Clouds Research in Future Cloud Computing*, Expert Group Report, Public version 1.0 (2012).
[29] M. Sedaghat, F. Hernandez, E. Elmroth, *Unifying cloud management: towards overall governance of business level objectives*, Procs. CCGrid'11 (2011), pp. 591–597.
[30] D.M. Smith, *Hype cycle for cloud computing*, Gartner Research G00214915 (2011).
[31] S. Sotiriadis, N. Bessis, N. Antonopoulos, *Towards inter-cloud schedulers: A survey of meta-scheduling approaches*, Procs. 3PGCIC'11 (2011), pp. 59–66.
[32] M. Tsugawa, A. Matsunaga, J. Fortes, *User-level virtual network support for sky computing*, Procs. 5th IEEE e-Science (2009), pp. 72–79.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 28, 2012
Accepted: Jan. 10, 2013

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 315–324. http://www.scpe.org
ISSN 1895-1767

A SIMULATION PLATFORM FOR EVALUATION AND OPTIMIZATION OF COMPOSITE APPLICATIONS

JĀNIS GRABIS* AND MARTINS BONDERS

Abstract. Composite applications are developed by integrating independent web services and deployed in a dynamic cloud based environment. An ability to modify the composite applications in response to changing business needs significantly contributes to agility of enterprise information systems. Deployment and execution in the cloud based environment allows to requisition resources necessary for efficient execution of the composite applications. However, properties of the composite applications directly depend upon characteristics of external services used and environmental factors, which in the case of public networks, exhibit high degree of variability. In order to address this issue, the objective of this paper is to develop a simulation and business process modelling based platform for evaluation of composite applications to ensure that the applications developed deliver expected performance. The combined approach allows for comprehensive evaluation subject to stochastic and dynamic factors, and the platform integration reduces the modelling overhead. Application of the simulation platform is demonstrated using an example of designing a composite application for a taxi call center.

Key words: Composite applications, optimization, simulation, web service selection

1. Introduction. Composite applications are developed by combining existing information technology resources to provide new business capabilities [15]. They are characterized by a high level of flexibility and agility and can be set up relatively quickly to capture new business opportunities or to adjust to changes in business processes. Most frequently composite applications are designed by composing external services such as web services. This fact allows to attain benefits associate with software assets reuse and to reduce infrastructure maintenance efforts. However, properties of the composite applications directly depend upon characteristics of external services used and environmental factors, which in the case of public networks, exhibit high degree of variability. Therefore, the selection of appropriate and reliable services is of major importance. Multiple methods have been elaborate for selection of such services from the set of candidate services providing similar functionality [17]. These methods often use Quality-of-Service (QoS) measurements as selection criteria and rely on optimization techniques for choosing the appropriate web services. This approach has a number of limitations. Optimization techniques have well-known limitations [14] and they cannot account for all factors affecting the service selection, particularly, stochastic and dynamic factors. The selection process is also decoupled from the design process of composite applications. Therefore, the web services selection might not adequately represent performance of the composite application as a whole and there could be a significant overhead associated with the web service selection process leading to increased effort and reduced agility of developed composite applications.

In order to address these limitations, the objective of this paper is to elaborate a platform for comprehensive evaluation of composite applications. The evaluation should ensure that applications developed deliver the expected performance.The platform combines optimization techniques with simulation for the selection of services and the evaluation of the composite application. It also uses a business process model underlying the composite application to be developed as the evaluation basis to reduce effort associated with development of multiple evaluation models. The platform includes a module for web service selection and a module for simulation of performance of the composite application depending upon the web services selected and environmental parameters. The web services are selected using the mathematical programming model, which accounts for both functional and non-functional requirements expressed in the terms of costs associate with application usage. The simulation module evaluates expected performance of the composite application and identifies key requirements for the execution requirement. The main contributions of this paper to the state of art are: 1) accounting for both functional and non-functional factors in the service selection; 2) providing of the simulation environment for evaluation of performance of the composite applications; and 3) integration of the simulation and optimization models with business process and executable process models. Application of the simulation platform is demonstrated using an example of designing a composite application for taxi call center.

The rest of paper is organized as follows. Section 2 reviews related research. The evaluation platform is introduced in Section 3. Section 4 elaborates optimization and simulation models used for evaluation of composite applications. Section 5 demonstrates application of the platform, and Section 6 concludes.

*Institute of Information Technology, Riga Technical University, Kalku 1, Riga, LV-1658, Latvia, (grabis@iti.rtu.lv).

Table 2.1: Overview of Web service selection methods.

Source	Method
Canfora et al. (2008) [5]	Genetic Algorithms
Cai et al. (2009) [4]	Artificial Neural Network
Hou and Su (2006) [8]	Analytic hierarchy process (AHP)
Huang et al. (2009) [9]	Linear programming techniques for Multidimensional Analysis of Preference
Lin et al. (2008) [11]	QoS Consensus Moderation Approach
Ma and Zhang (2008) [12]	Convergent population diversity handling genetic algorithm
Menasce et al. (2007) [13]	Integer programming
Sun et al. (2007) [18]	AHP and the BrownGibson (BG) methods
Wang et al. (2007) [22]	Fuzzy-based UDDI with QoS support
Wang et al. (2010) [23]	Fuzzy linear programming
Wu and Chang (2007) [25]	QoS meta-model as the basis for the QoS and AHP modelling

Table 2.2: Overview of QoS characteristics used in web service selection.

Source	Execution					Security			Strategic				
	Response time	Accessibility	Compliance	Successability	Availability	Encryption	Authentification	Access control	Cost	Reputation	Organizational arrangement	Payment method	Monitoring
Badr et al. (2008) [1]	x	x	x	x	x	x	x	x	x	x	x	x	x
Canfora et al. (2008) [5]	x	x			x				x				
Diamadopoulou et al. (2008) [7]	x	x			x		x	x					
Lin et al. (2008) [11]	x								x				
Tran et al. (2009) [19]	x	x		x	x				x		x	x	
Wang et al. (2007) [22]	x	x		x	x								

2. Literature review. Performance of composite applications directly depends upon performance of constituent web services and efficiency of their composition. The selection of appropriate web services has been an active research area.

A number of web service selection methods have been elaborated and several typical QoS measurements used in the web service selection can be identified by analyzing these methods. Table 2.1 surveys selected web service selection methods. All these methods are multi-criteria selection methods because the web service selection is an essentially multi-criteria problem. Analytical Hierarchical Process (AHP) is the most frequently method used. It is often used together with other methods. Different methods from the artificial intelligence domain such as fuzzy algorithms and artificial neural networks are also frequently considered to account for factors, which are difficult to express analytically.

The literature review suggests that there are two main categories of attributes used in the web services selection: QoS properties and business properties category [1]. The QoS properties category may be divided into two sub categories: execution and security properties. Table 2.2 lists nonfunctional QoS characteristics considered in selected papers. Response time, accessibility and availability are the most universally used characteristics in the QoS properties category. Cost is the most frequently used business related characteristic.

However, service consumers are equally concerned about both functional and nonfunctional characteristics of services and there have been attempts to expand the QoS concept in the case of web service selection by defining it as "the degree to which a system, component or process meets customer or user needs or expectations" [9]. This definition includes evaluation of both functional and nonfunctional requirements. Unfortunately, formal evaluation of functional characteristics in the framework of web service selection is more difficult than evaluation of nonfunctional characteristics. A functional quality of service approach [10] uses similarity measures to

identify interoperable web services. A QoS-aware service selection algorithm includes functional requirements in the model though these are represented only by a binary variable indicating either complete satisfaction or complete dissatisfaction of the requirement [20]. Generally, evaluation of functional characteristics either involves expert judgement or has limited resolution. Additionally, the service selection is an inherently multi-objective problem. Preemptive optimization and weighting based approaches are usually used to account for different often contradicting objectives. However, these methods again rely on judgemental appraisal of relative importance of each selection criterion. In this paper to account for different factors and objectives, an approach of expressing impact of all factors in terms of costs is used as suggested in [2].

Recently, it has been acknowledge that complexity of the service selection problem is increasing and more comprehensive service selection methods are needed. For example Vescoukis et al. (2012) [21] develop a decision support system for the service evaluation to managed environmental crises.

3. Evaluation platfom. Lifecycle of service-oriented and composite applications includes modelling, assembly, deployment and monitoring phases [24]. The evaluation platform elaborated is intended for addressing composite application design issues during the modelling and assembling phases. It has three main purposes:

1. Selection and composition of appropriate services used in design of the composite application;
2. Prediction of performance of the composite application;
3. Determination of performance requirements towards the composite application's deployment environment.

The main principles used to elaborate the evaluation platform are determined by the nature of the services selection and composition problem and the need to reduce the evaluation process overhead. In order to address the former issue, models capable to account for multiple objectives and uncertainty are used. To deal with the latter issue, model transformation and information reuse are utilized.

Figure 3.1 shows the main components of the evaluation platform. It is assumed that there are a number of candidate services proving functions required by the composite application and QoS data are available for these services. The evaluation of candidate services is performed using the optimization and simulation models. An optimization model in a form of mathematical programming model is formulated and solved using the optimization module. The optimization model selects services, which satisfy the functional requirements and have the optimal non-functional characteristics. A business process model using the Business Process modelling Notation (BPMN) notation shows composition of the services selected by the optimization model. QoS data are also represented in the business process model. The BPMN model supplemented with parameters specific to simulation purposes can be simulation using the simulation platform in order to evaluate performance of the composite application. The BPMN model can be transformed into an executable business process model (e.g., Business Process Execution Language (BPEL) model), where links with actual web services used during the execution are established. The final BPEL model is loaded into an execution platform, and the composite application is executed. The execution platform can be provided as a cloud based service. The composite application evaluation process using the proposed evaluation platform is shown in Figure 3.2. The service selection is performed jointly using the optimization and simulation model following principles of the hybrid simulation based optimization approach [6]. This approach utilizes the strength of optimization to evaluate a large number of possible service combinations and the ability of simulation to evaluate impact of stochastic factors what is important in case of using remote services. That allows for comprehensive evaluation of the selected services and their composition. If simulated performance of the composite application is not satisfactory, the evaluation process is repeated by changing candidate services, their composition or other parameters of the composite application and evaluation models.

A BPMN business process model is used as the main method for defining the composite application. It is capable of representing information required for simulation purposes. It can be used by the simulation platform and can be transformed into an executable BPEL model, which serves as a basis for implementation of the composite application. Using the transformations from the BPMN business process model to the simulation model and from the BPMN business process model to the executable BPEL model helps to reduce overhead associate with development of different evaluation models.

4. Evaluation models. The quantitative evaluation is performed using optimization and simulation models. The particular formulation of these models is case dependent though the main parameters and decision variables are common across multiple quantitative models used in design and evaluation of composite applications. The evaluation platform can be used together with different types of optimization and simulation

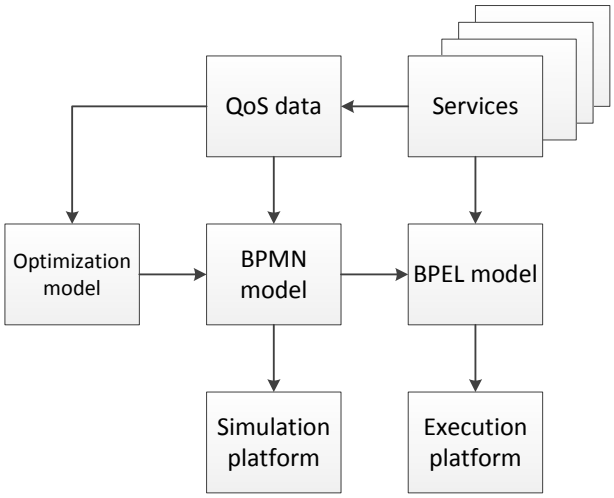

Fig. 3.1: The evaluation platform

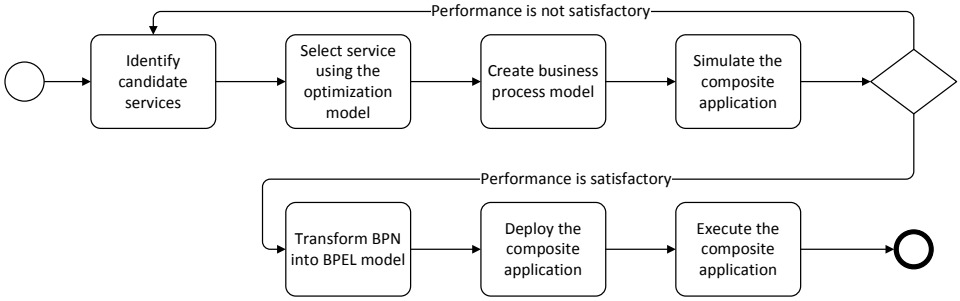

Fig. 3.2: The evaluation process

models.

4.1. Mathematical programming model. The mathematical programing model selects the most appropriate services for development of the composite application. It should account for both functional and non-functional requirements as well as to take into account multiple selection criteria. To achieve that, similarly as in [2] all selection criteria are expressed in terms of costs. These costs represent expenses associated with using the services selected from both functional and non-functional perspective. The following assumptions are made about features of the composite application:

- user requests are of different types depending upon input data provided;
- a number of candidate services provide similar functionality;
- all services can processes all types of the user requests though some of the services might need additional post-processing for some of the requests;
- if service returns an error, it is required and a positive response is received;
- if service is down for some time periods then the user requests are allocated to another service;
- each selected service incurs fixed costs (e.g., service integration costs, maintenance costs, usage fees).

The model objective function minimizes the total cost (TC) of using the selected web services over a definite planning horizon. The total cost is composed of the cost associated with service response time, the cost associate with requiring the service because of response errors and fixed costs due to using the selected web service (e.g., integration costs, maintenance cost, usage cost). Notations used to define the mathematical model are given in Table 4.1. The objective function 4.1 consists of four cost terms. The first term represents costs (denoted C_1)

Table 4.1: Notation

Notation	Description
i	index used to identify a service
j	index used to identify type of user request
N	number of candidate services
M	number of request types
$S_i \in \{0,1\}$	a decision variable indicating whether service is selected or not
X_{ij}	number of request of type jth assigned to ith service
r_j	number of request of type jth
t_{ij}	post-processing time for ith service for request of type jth
q_i^1	response time for ith service
q_i^2	percentage of requests returning an error for ith service
q_i^3	percentage of uptime for ith service
c^T	hourly composite application operating cost
c_i^F	fixed cost of using ith service
P	a large number

due to time spent on receiving responses from the selected web services, for instance, a user of the composite application who is paid an hourly rate waits till the response is received. The second term represents costs (C_2) due to the time spent on requerying services returning an error. The third term represents costs (C_3) due to time spent on post-processing of the results returned. The fourth term represents fixed costs (C_4) for using the selected services. The objective function is minimized by finding the optimal values of $\mathbf{S} = (S_1, ..., S_N)$ and $\mathbf{X} = (X_{11}, ..., X_{1M}, ..., X_{NM})$.

$$\begin{aligned} TC(\mathbf{S},\mathbf{X}) = &\sum_{i=1}^{N}\sum_{j=1}^{M} c^T q_i^1 X_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{M} c^T q_i^1 q_i^2 X_{ij} \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{M} c^T t_{ij} X_{ij} + \sum_{i=1}^{N} c_i^F S_i \rightarrow min \end{aligned} \tag{4.1}$$

$$\sum_{i=1}^{N} X_{ij} = r_j, \forall j \tag{4.2}$$

$$\sum_{j=1}^{M} X_{ij} = PS_i, \forall i \tag{4.3}$$

$$\sum_{j=1}^{M} X_{ij} \leq \sum_{j=1}^{M} q_i^3 r_j, \forall i \tag{4.4}$$

Eq. 4.2 implies that all user requests should be satisfied. Eq. 4.3 imposes that the requests can be assigned only to the services included in the composite application. Eq. 4.4 represents that a fraction of the user request cannot be met due to the service downtime if its reliability is less than one. As the result multiple services should be selected to provide a backup in the case of service unavailability. On the other hand requerying due to response errors in represented directly in the second term of the objective function. This representation of service downtime is simplified though more advance representation of this factor could make the optimization model intractable.

4.2. Simulation model. Simulation modelling is used to evaluate the composite application subject to dynamic and stochastic factors. In this case, simulation is performed using business process modelling tools, which usually have fewer simulation features than general purpose discrete event simulation tools [3] while provide a more business user friendly modelling environment and a set of concepts relevant to information

Table 5.1: List of candidate services and their properties.

Service	Geocoding by address	Geocoding by point of interest	Geocoding by intersection	q_i^1, s	q_i^2, %	q_i^3, %	c_i^F
Service 1	+	-	+	0.30	1	100	1000
Service 2	+	-	-	0.70	5	100	1000
Service 3	+	-	-	1.00	0	90	1500
Service 4	+	-	-	1,2	0	95	1800
Service 5	-	+	+	0.70	5	100	1000
Service 6	+	+	-	1.00	1	99	800

systems development [16]. In the case of composite applications, the use of business process modelling based simulators is also preferential because of their compatibility with BPEL or other executable business processes. In order to represent the composite applications and uncertainties associate with using external services, the required simulation modelling features are:

- representation of stochastically arriving user requests initiating the process execution;
- representation of stochastic service invocation response time;
- representation of random service invocation response errors and service downtime.

These features are supported by majority of business process simulation tools such as IBM Business Modeler and iGrafx Process. These tools also support simple mechanisms for allocating user requests to appropriate services though more advanced allocation mechanisms should be custom-coded (e.g., rerouting of the request during the service downtime).

5. Application example. Application of the evaluation platform is demonstrated using an example of taxi ordering call center. The company receives customer requests for taxi services. The customers order taxi by referencing their address, point-of-interest or intersection (these define the type of customer request). Call center operators lookup the particular location and identify available taxis using web services. Functionality and QoS characteristics of the candidate web services are given in Table 5.1 (these are real-life public web services though they are not named because their characteristics change continuously and exact data might not be valid at the time of publication). The table shows that, for example, Service 1 is able to geocode locations referenced by an address or a point-of-interest. Upon receiving the location information from web services, data post-processing is required. If accurate information is returned by the web service then post-processing is shorter and only includes confirmation of the customer request. However, if inaccurate information is returned (i.e., the service supports location search but not by the particular type of customer request) then post-processing takes longer and also includes manual checking using map services.

A composite application is developed to fuse results given by different web services and to minimize time operators spend on locating customers and assigning taxies to customer requests. The evaluation platform is used to identify appropriate web services and to evaluate expected performance of the composite application. Three scenarios are experimentally evaluated:

1. Standard scenario (S1) using a list of actual web services and their real-life functional and QoS characteristics;
2. Scenario with dedicated services (S2) each service is able to process only a specific type of customer requests;
3. Scenario with unreliable services (S3) service reliability is reduced to only 90% to evaluate the composite application in the case of network failure.

For the first scenario, it also important to investigate dynamical properties of the composite applications since number of customer requests and responsiveness of web services varies throughout the day. The scenarios are evaluated to determine the total cost of operating the composite application and to determine the customer request processing time. Initially, the optimization model is used to select appropriate services out of the candidate services. The optimization is performed for 1,750,000 user request made over one year. One half of the requests are by address, one third is by point of interest and one sixth is by intersection. The selected services satisfy all functional requirements and minimized the total ownership cost. The total cost breakdown for all three scenarios is given in Figure 5.1 ($c^T = 5$). Two services are selected for the first and the third scenarios, while three services are selected for the second scenario. That leads to increasing fixed costs in the case of the second scenario. The post-processing cost has the largest share since any manual operations are

much more time consuming than automated service calls. The optimization model gives the same result for the first and the second scenario because optimization model has limited means to represent impact of downtime.

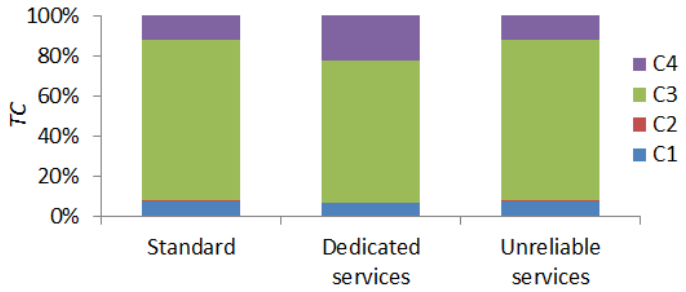

Fig. 5.1: The total cost breakdown

According to the optimization results, a BPMN business process model underlying the composite application is developed (Figure 5.2). The process starts with request receive activity representing a service for registering the user request and assigning request to a particular service depending upon the request type. The service also checks whether the service chosen is not unavailable. If the service is unavailable the request is reassigned to another service what might lead to increasing post-processing time. The appropriate location services are invoked and request post-processing is performed.

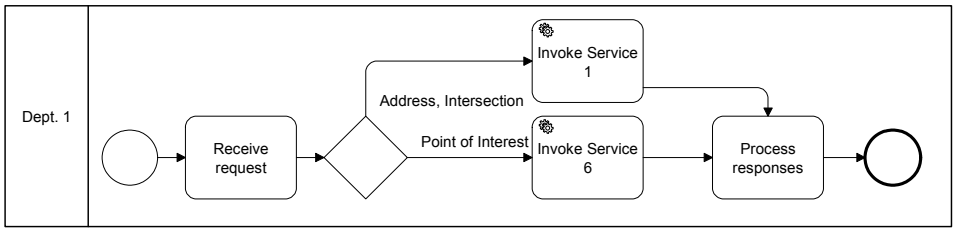

Fig. 5.2: Business process model underlying the composite application

The business process model is supplemented with data necessary for performing simulation based evaluation of the composite application. Randomly distributed execution time is specified for each activity and random service downtimes are also modeled. The customer requests are modeled as entities arriving at randomly distributed discrete time moments. Two cases are considered: 1) arrival rate is constant (R1); and 2) arrival rate varies throughout the day (R2). The case R2 represents the actual empirically observed customer requests arrival distribution. In the second case, a variable service response time is also used following the response time patterns identified by [26]. These patterns show that the service response time also exhibits the hourly variations. Therefore, impact of changes in customer requests and response time can be dynamically evaluated. Performance of the composite application is measured by a cycle time, i.e. process execution time from receiving the request till the final response to customer, and by interarrival rate of customer requests posted to external services. The latter measure is important to identify possibilities of clogging the external service.

Figure 5.3 shows a histogram of cycle time distribution for selected cases evaluated using simulation. The average cycle time for the three cases evaluated are 15.7, 17 and 16.7, respectively. These differences are statistically significant. The cycle time is the most predictable in the of uniform arrival rate of the customer request. The variable pattern of the customer requests, what is not account for in the optimization model, leads to less predictable and stable cycle times. The cycle time increase due to the service downtime also was not accounted for in the optimization model. Particularly, there are a number of requests with twice as long cycle time due to unavailability of the most appropriate service. Feeding back these results into the optimization model might result in selection of additional back-up services. Although the cycle time differences are numerically small these might lead to a necessity to higher more operators at the taxi call center over the long planning horizon.

Figure 5.4 shows interarrival time between subsequent customer requests. It can be observed that in the case of the variable customer requests pattern, there are more occasions with a short interarrival period. This particular composite application does not create a large load on external services but for other applications this result cloud be important to identify requirements for the execution platform concerning a number of simultaneous requests it is able to process.

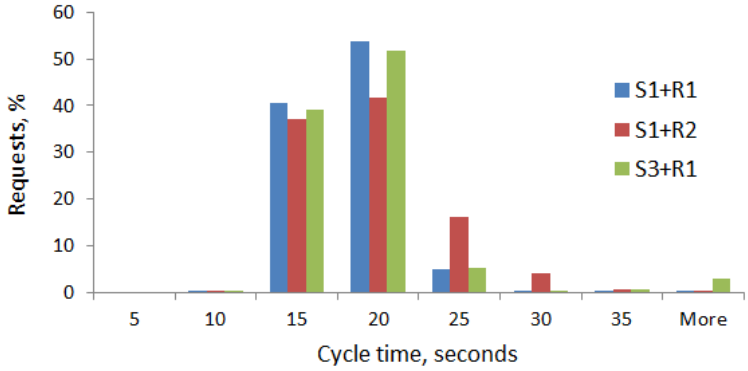

Fig. 5.3: The simulated cycle time of the composite application execution

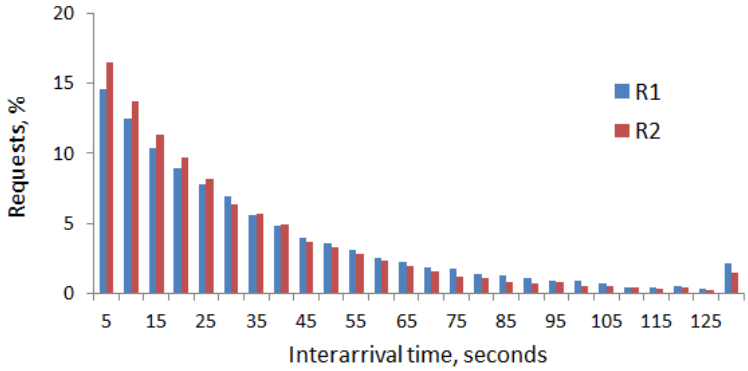

Fig. 5.4: Interarrival time between subsequent customer requests for scenario S1

For the standard scenario, performance of the composite application for demand pattern R2 is also investigated. Figure 5.5 shows the relative response time increase according to the hour of the day as suggested in [26], the actually observed number of customer requests and cycle time of the customer request processing by the composite application. It can be observed that the cycle time strongly correlated with the performance of the composite application. The number of customer requests does not have impact on the response time since service workload created by this single composite application is negligible with the global workload. However, it can be observed that, especially in the late afternoon, the increase of customer requests coincides with deteriorating service response time performance. As the result, the composite application gives the worst performance exactly when it is most frequently utilized.

In order to obtain the aforementioned results, a prototype of the simulation based evaluation platform was developed. IBM Rational System Architect is used as the core component of the platform. It is used to define all concepts relevant to development of the composite application, to develop the business process model and to

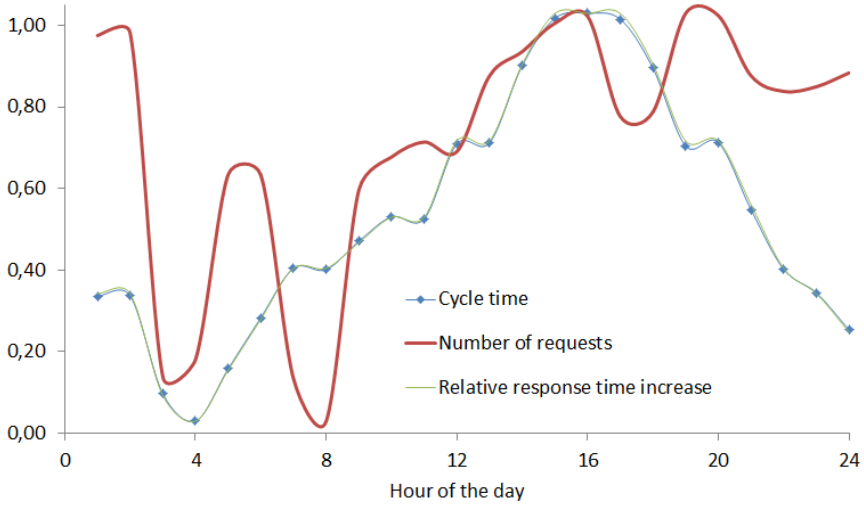

Fig. 5.5: Hourly variations of performance of the composite application. All measures are scaled to vary from 0 to 1.

perform business process simulation using the built-in Witness simulator. The optimization is performed using Lingo Solver. Executable business processes are handled using IBM Business Process Manager, which imports the business process model from IBM Rational System Architect. Data exchange between different models is performed using spreadsheet tools.

6. Conclusion. A simulation platform for development and evaluation of composite applications has been elaborated in this paper. It supports development of multiple interlinked models enabling for comprehensive evaluation of the composite applications. The experimental results show that the platform is particularly valuable to evaluate dynamic and stochastic features of the composite applications. These features cannot be effectively evaluated by just using optimization models because they become computationally intractable. The simulation results also can be used to set requirements for application execution environment. For example, the dependence of the cycle time on variable customer requests arrival pattern sets requirements for scalability in the cloud environment, where the cloud services provider should ensure that the service quality does not deteriorate at the time periods crucial for businesses support by the composite application. The obtained results are significant because without using the platform the effort of evaluation of candidate services would be much more significant and using just a single optimization model without the simulation model would not allow to fully appraise uncertainty of using internet based services in development of composite applications. The current business process modelling tools do not provide an adequate support for experimenting with business process models. In the platform prototype these functions are implemented using spreadsheets. As indicated in the literature review different types of web service selection models are available. The optimization model used in this paper can be replaced with another service selection model if appropriate, and the platform can still be used to for evaluation of the web services selected.

The composite applications are fully fledged applications including user interface and persistent data storage. The platform currently focuses on the process composition, and evaluation of other parts of the composite applications is a subject for future research.

REFERENCES

[1] Badr, Y., Abraham, A., Biennier, F. & Grosan C., *Enhancing Web Service Selection by User Preferences of Non-Functional Features*, Proceedings of the 2008 4th International Conference on Next Generation Web Services Practices (2008), pp. 60-65.

[2] Bonders, M., Grabis, J. & Kampars, J., *Combining Functional and Nonfunctional Attributes for Cost Driven Web Service Selection*, in Frontiers in Artificial Intelligence and Applications, Barzdins, J., Kirikova, M. (eds) 224 (2011), 227-239.
[3] Bosilj-Vuksic, V., Ceric, V., & Hlupic, V., *Criteria for the evaluation of business process simulation tools*, Interdisciplinary Journal of Information, Knowledge, and Management, 2(2007), 73-88.
[4] Cai, H., Hu, X., L, Q., & Cao, Q., *A novel intelligent service selection algorithm and application for ubiquitous web services environment*, Expert Systems with Applications, 36(2009), pp.2200-2212.
[5] Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., *A framework for QoS-aware binding and re-binding of composite web services*, Journal of Systems and Software, 81(2008), 1754-1769.
[6] Chandra, C. & Grabis, J., *Supply Chain Configuration: Concepts, Solutions, and Applications*, Springer: New York, 2008.
[7] Diamadopoulou, V., Makris, C., Panagis, Y., & Sakkopoulos, E., *Techniques to support web service selection and consumption with QoS characteristics*, Journal of Network and Computer Applications, 31(2008), pp.108-130.
[8] Hou, J., & Su, D., *Integration of web services technology with business models within the total product design process for supplier selection*, Computers in Industry, 57(2006), pp. 797-808.
[9] Huang, A. F. M., Lan, C., & Yang, S. J. H., *An optimal QoS-based web service selection scheme*, Information Sciences, 179(2009), pp. 3309-3322.
[10] Jeong, B., Cho, H., & Lee, C., *On the functional quality of service (FQoS) to discover and compose interoperable web services*, Expert Systems with Applications, 36(2009), pp. 5411-5418.
[11] Lin, W., Lo, C., Chao, K, & Younas, M., *Consumer-centric QoS-aware selection of web services*, Journal of Computer and System Sciences, 74(2008), pp. 211-231.
[12] Ma, Y., & Zhang, C., *Quick convergence of genetic algorithm for QoS-driven web service selection*, Computer Networks, 52(2008), pp. 1093-1104.
[13] Menasc, D. A., Ruan, H., & Gomaa, H., *QoS management in service-oriented architectures*, Performance Evaluation, 64(2007), pp. 646-663.
[14] Nolan, R.L. & Sovereign, M.G., *A recursive optimization and simulation approach to analysis with an application to transportation systems*, Management Science, 18(1972), pp. 676-690.
[15] Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F., *Service-oriented computing: State of the art and research challenges*, Computer, 40(2007), pp. 38-45.
[16] Ren, C., Wang, W., Dong, J., Ding, H., Shao, B. & Wang, Q., *Towards a flexible business process modelling and simulation environment*, Proceedings - Winter Simulation Conference (2008) pp. 1694.
[17] Strunk, A., *QoS-aware service composition: A survey*, Proceedings of the 8th IEEE European Conference on Web Services, ECOWS 2010 (2010), pp. 67-74.
[18] Sun, Y., He, S., & Leu, J. Y., *Syndicating web services: A QoS and user-driven approach*, Decision Support Systems, 43(2007), pp. 243-255.
[19] Tran, V. X., Tsuji, H., & Masuda, R., *A new QoS ontology and its QoS-based ranking algorithm for web services*, Simulation Modelling Practice and Theory, 17(2009), pp. 1378-1398.
[20] Tsesmetzis, D., Roussaki, I., & Sykas, E., *QoS-aware service evaluation and selection*, European Journal of Operational Research, 191(2008), pp. 1101-1112.
[21] Vescoukis, V., Doulamis, N. & Karagiorgou, S., *A service oriented architecture for decision support systems in environmental crisis management*, Future Generation Computer Systems, 28(2012), pp. 593-604.
[22] Wang, H., Lee, C., & Ho, T., *Combining subjective and objective QoS factors for personalized web service selection*, Expert Systems with Applications, 32(2007), pp. 571-584.
[23] Wang, P., Chao, K., & Lo, C., *On optimal decision for QoS-aware composite service selection*, Expert Systems with Applications, 37(2010), 440-449.
[24] Woolf, B., *Introduction to SOA governance*, IBM developerWorks, http://www.ibm.com/developerworks/library/ar-servgov/, 2007.
[25] Wu, C., & Chang, E., *Intelligent web services selection based on AHP and wiki*, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007, pp. 767-770.
[26] Wu, Q., Iyengar, A., Subramanian, R., Rouvellou, I., Silva-Lepe, I., & Mikalsen, T., *Combining quality of service and social information for ranking services*, 7th International Joint Conference on Service-Oriented Computing , Lecture Notes in Computer Science, Vol. 5900 (2009), pp. 561-575.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 27, 2012
Accepted: Jan. 10, 2013

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 325–338. http://www.scpe.org
ISSN 1895-1767

INTEGRATION OF CLOUD-BASED SERVICES INTO DISTRIBUTED WORKFLOW SYSTEMS: CHALLENGES AND SOLUTIONS

PAWEL CZARNUL*

Abstract. The paper introduces the challenges in modern workflow management in distributed environments spanning multiple cluster, grid and cloud systems. Recent developments in cloud computing infrastructures are presented and are referring how clouds can be incorporated into distributed workflow management, aside from local and grid systems considered so far. Several challenges concerning workflow definition, optimisation and execution are considered. These range from configuration, integration of business and scientific services, data management, dynamic monitoring and tracking, reusable workflow patterns, semantic search and distributed execution of distributed services. Finally, the author recommends a solution to these challenges based on the BeesyCluster middleware for distributed management of services with static and dynamic rescheduling within a market of services.

Key words: workflow management, cloud computing, services on the cloud, service integration

1. Introduction. Integration of services into distributed workflow applications has been covered widely in the literature for grid based systems [4, 5]. Several solutions have been proposed for:

- a conceptual model of the workflow including DAGs with extensions,
- QoS modelling, integration and management,
- actual implementations of workflow management systems for both business and academic applications.

Since cloud-based computing including SaaS, IaaS and PaaS has become more and more popular and widely used, it is expected that knowledge and solutions to service integration developed for grid-based workflow solutions would be adopted and extended for cloud-based environments.

From the point of view of an enterprise, integration into workflows is of key importance as it allows to model processes such as processing orders, banking, payroll, B2B cooperation, flow of production processes and many others. Cloud computing adds new potential to this but at the same time several challenges arise that will be formulated in the paper and for which solutions will be presented:

- need for uniform interfaces and middleware for: service management, data transfer and handling in the cloud environment especially in the context of automatic service discovery and matching across clouds,
- uniform QoS monitoring and assurance across various cloud providers and types of services,
- algorithms for composition of workflows in the cloud environment including dynamic learning of service information including QoS,
- dynamic changes of the QoS parameters and cloud availability,
- incorporation of several clouds to avoid the vendor lock-in problem for effective implementation of workflow management in sky computing,
- integration of both ready-to-use SaaS, IaaS, PaaS as well as human interactions and decision making also in workflows spanning several enterprises,
- integration of legacy applications, SOA, grid and cloud computing into workflows effectively merging the recent paradigms for distributed computing.

In this paper, the aforementioned challenges and solutions to these will be presented from the following perspectives: a conceptual model or models proposed for the given challenge, technological aspects, APIs, implementations if already exist. Otherwise, suggestions and hints on how to adapt the existing state-of-the-art to provide future elastic, efficient and cost-effective workflow systems in clouds will be provided.

2. Related work.

2.1. Workflow definition and execution. Integration of distributed services is often modelled as workflow applications which are most often described as DAGs (Directed Acyclic Graphs). In a DAG G(V,E) a set of vertexes V corresponds to tasks that are needed to accomplish a complex scenario while the set of directed edges E corresponds to time dependencies between the tasks they connect. Such a DAG describes a recipe for a complex scenario, either a business or a scientific application. It does not yet refer to any executables that are supposed to perform the tasks thus such a workflow is called abstract. For each task, there may be several services capable of executing it, albeit at different QoS terms. Each of such services may differ in:

*Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 11/12 Narutowicza Street, 80-233, Gdansk, Poland, (pczarnul@eti.pg.gda.pl) http://fox.eti.pg.gda.pl/~pczarnul.

execution time, cost, location which may impact the communication cost of staging in and out data, reliability, availability, conformance to standards etc. Assignment of a particular service to each workflow task results in a complete solution to how the abstract workflow is to be executed and makes the workflow concrete. Such a set of services results in the final resulting QoS, in particular the execution time and cost of the workflow. A Workflow Management System (WfMS) is a system that allows a user to:

1. define a workflow, often using a graphical editor to specify tasks, draw edges of the workflow G graph,
2. for each task:
   - specify functional requirements i.e. what the task is supposed to do or
   - assign a set of services, each of which is capable of executing the task, possibly at different QoS terms,
3. define QoS goals for the workflow in terms of a global (for the whole workflow) optimisation goal and global and/or local (for a particular task) constraints, possibly with multiple criteria [29, 25]. For instance, the optimisation goal might be to select such services (one service per task) such that the workflow execution time is minimised and the cost of the selected services is below a given threshold [36, 33],
4. perform workflow scheduling and optimisation i.e. select such services to meet the optimisation goal. In general, optimal selection will be infeasible computationally for large configurations which forces to fall back to heuristic [31] service selection and scheduling algorithms,
5. execute the workflow in a real distributed environment,
6. track the statuses of previously run workflow applications,
7. fetch results of the workflows.

Workflow applications can be categorised into the following types:

- scientific [34], characterised by:
  - structure: in which mainly compute-intensive tasks are executed on input data of large or moderate size; usually many parallel paths execute parts of computations on possibly smaller fractions of data,
  - QoS: traditionally mainly the workflow execution time and cost (corresponding to e.g. hiring HPC resources) were used.
- business, characterised by:
  - structure: usually operates on data of smaller size than scientific workflows; control flow is more complex (possibly requires more control structures than in the regular DAG) than in scientific workflows,
  - applications: document flow, processing orders in B2B scenarios, handling and processing client orders,
  - QoS: many more metrics considered for services and as the QoS goal than in scientific workflows e.g. availability, accessibility, security, reputation [24, 38].

2.2. Cloud systems. The fundamental assumption of cloud computing is outsourcing compute and storage capabilities which brings several consequences:

- the pay-as-you-go policy instead of the fixed initial cost and only running costs of on-site or grid systems [16],
- no need for maintenance and upgrades of equipment, software updates, handling of compatibility issues among software and hardware components,
- letting the cloud provider to manage computations, data and networking among the software components run on the cloud. This brings data privacy concerns for some businesses and may rule out cloud computing for some of them,
- the possibility of falling into the vendor lock-in trap if the client becomes too much invested and depending on just one provider.

Cloud systems may expose various kinds of services (each next layer makes use of the former in a layered architecture, from bottom to top closer to the cloud client):

- IaaS, Infrastructure as a Service. In this case the cloud provider exposes basic components such as computers for computing (such as virtual machines), storage, networks, load balancers, firewalls. Ex-

amples include: Google Compute Engine[1], Amazon Elastic Compute Cloud (EC2)[2], RackSpace Cloud Servers[3], Rack Space Cloud Files[4],
- PaaS, Platform as a Service. In this case, the whole operating platform is provided by the cloud provider that may include components such as an operating system, database server, web server. Clients can run software on the cloud using these components. Examples include: Aneka [23], Google AppEngine[5], Windows Azure[6], RedHat Openshift[7], RackSpace Cloud Sites[8],
- SaaS, Software as a Service. Cloud providers offer software installed on the cloud while cloud clients use the software. The software can be run transparently on virtual machines, updated and maintained by the cloud provider. Examples include: Google Apps[9], Salesforce[10].

There are several software packages that allow building of private or public clouds such as: Eucalyptus[11] (for IaaS with interfaces compatible with Amazon EC2 and S3), OpenStack[12] (for IaaS with interfaces compatible with Amazon EC2 and S3) with OpenStack Compute, OpenStack Storage, OpenStack Networking and OpenStackDashBoard, Open Nebula[13] for building IaaS datacenter vitalisation with a choice of interfaces such as AWS, OCCI and hypervisors such as Xen, KVM, VMWare, Nimbus[14] (provides an implementation of Amazon EC2 interface) for IaaS through deployment of virtual machines on resources and offering to users, Cumulus to provide storage cloud implementation (interface compatible with Amazon S3).

2.3. Workflow management in grid and cloud systems. This section presents the state-of-the-art and recent developments regarding management of distributed workflow applications on clouds, especially compared to running workflows on local and grid systems.

2.3.1. Running Workflow Applications on Cloud vs Grid Systems. Usefulness of cloud computing for large-scale workflows is evaluated against the typical use of grid systems in [16, 28]. One of the crucial differences between running in these two environments is the pay-per-use scheme in cloud computing compared to the one cost policy in grids or local cluster systems [16].

For very large workflows, it is advised to cluster smaller jobs into batches to minimise the scheduling overhead and the overhead of handling too many jobs [28]. FutureGrid was used for distributed processing of workflows across several distributed sites using Eucalyptus and Nimbus. This in fact implemented running on several clouds i.e. sky computing. Additionally, experiments were performed on three separate clouds: Magellan (with Eucalyptus), Amazon (with EC2) and FutureGrid (with Eucalyptus) with very similar results in terms of performance taking into consideration particular configurations.

Cloud-based systems offer several benefits compared to grid systems for running distributed workflow applications [17]:
- dynamic provisioning of resources that can be ordered dynamically at runtime; the Aneka Cloud [23] is able to scale horizontally to acquire more resources as these are needed. Such resources can be obtained from other clouds such as Amazon EC2 to allow the application submitted to Aneka to complete in the desired time frame. This can be especially useful for scientific workflows in which many paths and services are requested to be executed in parallel. More resources can be ordered at runtime for parallel execution, albeit at the increased cost.
- easier possibility to run legacy applications as particular hardware/software configurations of cloud resources can be booked as opposed to using the fixed set of grid computing sites and nodes.

It should be noted that wide area networks offer much larger latency than clusters which can impact workflow execution times severely [17]. However, this is more important for scientific rather than business

[1] http://cloud.google.com/products/compute-engine.html
[2] http://aws.amazon.com/ec2/
[3] http://www.rackspace.com/cloud/cloud_hosting_products/servers/
[4] http://www.rackspace.com/cloud/cloud_hosting_products/files/
[5] https://developers.google.com/appengine/
[6] http://www.windowsazure.com
[7] https://openshift.redhat.com/app/
[8] http://www.rackspace.com/cloud/cloud_hosting_products/sites/
[9] http://www.google.com/Apps
[10] http://www.salesforce.com/eu/
[11] http://www.eucalyptus.com/
[12] http://openstack.org
[13] http://opennebula.org/
[14] http://www.nimbusproject.org/docs

oriented workflow applications.

2.3.2. Systems and Environments. Many workflow management systems have been proposed, especially for grid computing. These include:

- Taverna[15], Kepler[16] [21], Triana[17] [22], Galaxy, Pegasus [11, 12], Askalon [30], Conveyor [20],
- Tavaxy [1] a system for definition and execution of workflow applications based on patterns. It integrates Taverna and Galaxy and allows to either run the whole system in a cloud or launch a part of the workflow on the cloud,
- Meandre[18] – a system for composition semantic enabled flows for data processing. It allows creation of reusable components and RDF is used to standardise publishing.

There are also other business oriented workflow solutions such as:

- Chronos Workflow Platform[19] - a system for automation of repeated business processes. It handles complex flows, parallel processes and external interfaces. It is suitable for any many types of business process including finance, CRM, HR, R&D, marketing, administration, logistics, production.
- Affinity Live[20] - a platform for managing business processes in one place in the cloud. Suitable for e.g. project management, sales, invoicing. It integrates e.g. with Google Apps, Microsoft Exchange Server.

The workflow representation can be made in various forms [15], starting from XPDL, ebXML through Petri-Nets used in Triana, BPEL in Akogrimo up to OWL-based such as OWL-WS in NextGrid. Some systems such as BeesyCluster can use proprietary representations which can be also exported to widely known formats such as BPEL. SHIWA (SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs)[21] aims at interoperability between various workflow systems and transformation of workflow representations.

Running workflow applications on top of several different, distributed resources is presented in [23]. Through an Aneka plugin the workflow engine can use the Aneka Cloud. An EC2 plugin allows to access Amazon Web Services. Additionally, a local cluster with a fixed number of resources can be utilised by the workflow engine. This in fact implements running a workflow on top of several distinct environments including clusters, grids and clouds. The workflow management system presented in [23] supports Aneka, Globus, PBS. The plugins allow to transfer data to and from resources, monitoring statuses of started workflow tasks along with energy consumption. It is demonstrated that a scientific workflow application for evolutionary multi-objective optimisation scales well in terms of the number of iterations of the algorithm when adding new virtual machines to the system.

There are several solutions for integration of software and resources on clouds, oriented on business cases and workflows rather than scientific applications:

1. Metastorm Smart Business Workspace [27] deployed on Microsofts Azure cloud as a version of the on-premise version of the Metastorm software,
2. Amazon Simple Workflow[22] that allows to define, run and control business workflows spanning cloud-based, on-premise or both types of systems focusing on the business logic. The workflow is to represent a business scenario such as processing orders on a Web site including: management of orders, various payment options including charging credit cards, notification, management of shipping items, inventory, returns etc. Several concepts are introduced including:
   - actions corresponding to workflow tasks,
   - activity workers implementing the tasks i.e. services,
   - decider that decides on the workflow logic i.e. checking if a condition is satisfied so as to process appropriate actions,
   - domain a collection of related workflows. Workflows can be managed through the AWS Management Console. The payment scheme for running workflows is pay-as-you-go with the usual AWS charges for data transferred out of the workflow.

[15] http://www.taverna.org.uk/
[16] https://kepler-project.org/
[17] http://www.trianacode.org/
[18] http://seasr.org/meandre/documentation/architecture/
[19] http://www.chronosworkflow.com/
[20] http://www.affinitylive.com/product/benefits/powerful-workflow-and-business-processes/
[21] http://www.shiwa-workflow.eu/
[22] http://aws.typepad.com/aws/2012/02/amazon-simple-workflow-cloud-based-workflow-management.html

3. RunMyProcess platform for development of business workflows using Google Apps. Several examples are cited[23]:
   - purchase-order management in aerospace consultancy,
   - incident management system for a travel agency,
   - project approval process by a bank.
4. OneSaaS[24] is a SaaS cloud integration platform for integration of separate software systems running on separate clouds or sites. It focuses on applications such as CRM, eCommerce, invoicing, email marketing, event management, project and team management, accounting.
5. Questetra BPM Suite[25] - a SaaS system for business process management with a web-based interface.

2.3.3. Scheduling algorithms. Workflow scheduling [32, 35] requires in fact two steps in order to achieve the stated QoS goal (such as minimisation of the workflow execution time) while keeping other QoS constraints (such as the total cost of selected services must not exceed the given threshold): selection of a service for each task, running the service on the given resource at a particular moment in time.

Static scheduling takes place when services are selected and scheduled before the workflow application is executed. This requires the knowledge about services capable of executing particular tasks upfront. It is often the case that some services fail during execution, become unavailable while the workflow is in progress or new, more interesting in terms of QoS optimisation, services appear. In such a case, dynamic rescheduling must be adopted that refreshes the list of available services at runtime. Consequently, execution time and cost may differ from run to run.

Since solving the workflow scheduling problem optimally is NP-hard, heuristic algorithms need to be adopted for large workflows. There is a wide spectrum of literature on workflow scheduling algorithms on grid systems [32, 4, 5, 14]. The following types of algorithms were suggested for workflow scheduling on grid systems:

1. ILP (Integer Linear Programming) methods [13],
2. genetic algorithms [34],
3. divide-and-conquer where the initial DAG is partitioned into smaller DAGs with modified constraints and solutions to smaller problems (sub-graphs) constitute the final solution,
4. GAIN [25] in which a viable solution is found first and then iteratively improved by selection of better (in terms of QoS) services.

There is a survey of workflow scheduling algorithms for cloud environments in [2]. Nine algorithms meant for cloud environments are compared in terms of scheduling methods, scheduling parameters, goals and supporting tools. It is concluded that most of the algorithms consider workflow execution time and cost (as those developed previously for grid systems), some consider resource utilisation. According to the survey, none of the algorithms take into account reliability nor availability.

Optimisation of workflow makespan on cloud systems using a traditional list ordering is proposed in [18]. First, urgency (high or low) of jobs (tasks) determines ordering of these. Secondly, importance (high or low) calculated for resources by a resource manager determines how resources are used first.

2.3.4. Applications. Apart from business workflows run in dedicated systems mentioned above, there are several scientific workflow applications which were suggested and executed in distributed environments, some involving cloud environments. Some of them are listed below:

1. protein analysis workflow (run in Taverna and Tavaxy) [1],
2. metagenomics workflow (the most compute-intensive part of the workflow was run on a cloud using the Amazon WS cloud along with the S3 storage) [1],
3. generation of periodograms that aims at identification of periodic signals in light curves that record brightness of stars over time. This can be done by handling separate frequencies in parallel [28],
4. evolutionary multi-objective optimisation: several instances of genetic algorithms can be run in parallel; it is demonstrated that adding new virtual machines (EC2 compute resources) can significantly increase the number of iterations in the simulation [23].

3. Integration of distributed business and scientific workflows using grid and cloud computing. In this section, we list both the challenges already raised in the literature as well as additional aspects as we

[23] http://www.runmyprocess.com/
[24] http://www.onesaas.com/
[25] http://store.questetra.com/en/

see to be needed for seamless integration of both scientific and business services in integrated grid and cloud computing environments.

Furthermore, solutions to these problems are suggested as extensions to the BeesyCluster[26] environment which already contains a workflow management subsystem for distributed service-based workflows with dynamic rescheduling, a market of services and a pluggable architecture for scheduling algorithms.

3.1. Challenges and Problems.

3.1.1. Preparation and Configuration for using Clouds. In case of ready to use solutions such as IaaS, one needs to be acquainted with the API offered by the cloud provider and have a client ready to use it. In case of private clouds, these require setting up.

Setting up a virtual cluster, virtual machines, preparation of images to be uploaded to cloud resources is discussed in [16]. When setting up own clouds for scientific workflows, much configuration and installation of tools must be done for running distributed workflows using virtualised cloud resources. Tools such as Virtual Workspace Service, Xen, Nimbus were used for virtualised resources. Pegasus, Condor DAGMan as well as GridFTP and GRAM were used to run workflows [16].

On the one hand, preparation of virtual machine images with necessary configuration for individual applications results in provision of necessary scalability thanks to using clouds compared to a local environment. On the other hand, though, it requires both effort in preparation of the images and overhead of setting up an environment [28]. The process of setting up, configuration and deployment of virtual clusters out of collections of virtual machines is called contextualisation [17]. It is complex to perform manually thus tools such as Nimbus Context Broker are suggested to make this process more automatic [17]. It is used to manage the virtual cluster and start appropriate services.

3.1.2. Software Stack for Distributed Workflows. There are more tools necessary to run workflows on clouds than just configuration of the virtual machines and clusters on cloud resources [17]. This involves a layer on top of clouds if workflows are to be run on geographically distributed resources, either cloud or grid based.

Firstly, if more different clouds, grids and local systems are to be used within one workflow, an integrating layer is needed with proper plugins for all underlying cloud, grid providers and local cluster systems such as PBS, LSF etc. This will allow sky computing [19] by using all clouds in a single workflow.

Secondly, in order to execute workflow applications on top of these resources, proper workflow management software communicating through the plugins with the resources are needed. Examples of such tools include:

- Pegasus and Condor DAGMan [16],
- Cloudbus WfMS (Workflow Management System) along with one or more of the following: Aneka, PBS, Globus [23],
- BeesyCluster with its WfMS and PBS, LSF on clusters [6]

Complex configuration and the lack of appropriate tools to set up and run workflows on clouds is listed as one of crucial challenges [17].

3.1.3. Efficient Data Management and Storage for Running Distributed Workflow Applications. Running workflow applications requires copying input data to locations of computing services and output data to following services. Communication cost, in case of large data sizes, can visibly impact the total workflow execution time if locations of such services are far from each other or the client. For instance, in [28] larger workflow execution time on an Amazon cloud compared to Magellan and FutureGrid located in the same state as the client is attributed to poor WAN performance apart from slightly slower processors on this cloud. Cloud systems may offer shared file systems that reduce communication costs within the cloud.

Furthermore, as we proposed in [10], software agents may be engaged for management of data at the workflow management layer and migrate to nodes closer to computing services so that communication time between services is minimised. The very same approach can be used for sky computing in case of geographically distributed clouds.

Several problems exist related to data management in distributed workflow applications run on clouds [17] [37]:

[26] `https://lab527.eti.pg.gda.pl:10030/ek/AS_LogIn`

- costly data movement between cloud resources, especially if more clouds are used for sky computing. It may be impossible to track the actual locations of data where it is stored. Cloud providers may charge for the amount of data transferred.
- it is not straightforward to set up a shared file system for use on a virtual cluster from a cloud provider. This may not be sufficient if more cloud providers are used. A widespread virtual file system would be an ultimate goal from the workflow perspective in terms of ease of use.
- particular clouds can offer specific APIs for submission and management of data. For instance, in Aneka [23] there is a Storage Service that allows to store input and output data of submitted tasks. Data can be obtained from the client machine, an FTP server or an Amazon S3 storage.
- the next problem related to data when running workflows on top of several clouds and in general often raised for sky computing is data exchange among several clouds in terms of coherency [23] and formats.

3.1.4. Integration of Business and Scientific Services. Most workflow management systems and solutions, as presented above, are dedicated to either scientific or business applications, not both. In some cases both scientific and business aspects appear in a single workflow. For instance, a company performing designs of buildings and bridges must cooperate with external customers and businesses (business part) as well as use HPC (High Performance Computing) services in order to perform multiple analyses of how their designed structures stand various parameters of wind, flood etc. Thus, one workflow application mixing both component types would be needed to model such a project. Such a capability would need to allow modelling and execution of the following types of services in one workflow application:

1. processing documents (e.g. orders, specifications or files),
2. involvement of human actors (e.g. to approve of or order another execution of the same service, or engage more human actors in voting on which path of the workflow should follow),
3. launching HPC long-running simulations hiding low-level details such as queueing systems, access to HPC resources etc.,
4. involvement of several distinct administrative parties (e.g. different companies trying to check if an initial task defined by a client can be solved by a design company or allowing cooperation of various partners in a consortium).

Integration of both types: scientific and business is analysed in [26]. The proposed approach is that there is a business workflow with possibly human tasks that controls the main flow and scientific, lower-level workflows are launched in certain tasks of the business workflow. Nodes of scientific workflows can spawn scientific simulations. Human tasks correspond to e.g. to provide information or make a decision. Still, there are new requirements and innovations that are suggested by the author:

1. incorporation of workflow patterns into workflows with business, scientific and human elements,
2. describing all services (business, scientific and human) using same ontologies in terms of both functional and QoS descriptions and incorporation of these into scheduling.

3.1.5. Dynamic Monitoring and History of Reliability and Availability. Other than dynamic provisioning of resources which is the advantage of cloud computing, there is the issue of QoS of the cloud services being offered: IaaS, PaaS or SaaS. According to the survey of workflow scheduling algorithms for clouds [2], there is a clear need for consideration of reliability and availability in scheduling workflows. None of the nine algorithms for clouds consider these.

The above challenge is closely related to the need for a registry of cloud services from various cloud providers [23]. Then, a proper scheduling policy considering many users who want to run multiple workflows should be developed in order to offer fair distribution of resources. This very much resembles the already known solutions in queueing systems for clusters such as PBS, LSF, LoadLeveler albeit at the higher level of the software stack. This should take into account storage and communication costs. This is again a known problem in scheduling workflows [3] but now defined higher at the multi-cloud level for sky computing [19].

Thus, this results in the need for monitoring and ranking of the following features:

- reliability,
- availability,
- rate of changes in QoS terms (such as the cost of computing and storage power in IaaS, price of access to a platform configuration in PaaS, price of access to software in SaaS, rate of software version updates in SaaS). While there exist tools for runtime monitoring and selection of best e.g. IaaS offers for desired

settings (such as required compute power, memory size, storage capacity) such as Cloudorado[27], this needs to be extended for the other metrics as well.

3.1.6. Reusable Workflow Patterns/Templates. It is much easier to construct workflow applications out of ready-to-use and reusable patterns. Most of the workflow management systems use the DAG model in which the following control statements are available:

- sequence,
- fork different ways of partitioning data among following workflow nodes are possible [1], [7] for instance, same data may be sent to each of the following nodes or data may be partitioned among following nodes.
- join.

Paper [1] suggests several additional patterns for workflow definitions, considering for control patterns also:

- multi-choice fork one of several following tasks is executed depending on the user-defined condition,
- iteration a certain task is repeated a predefined number of times or the number of iterations may also depend on the output data and a condition set on it

and for data patterns the following ones:

- list operations are performed on each of the list elements separately,
- product out of two input lists with elements, operations are performed on pairs of elements on corresponding places in the lists (dot product) or all combinations of elements in the two lists (cross product),
- data select depending on the outcome of a user-defined test, one or the other of input data is passed,
- data merge concatenation of input data lists is passed further.

Still, according to the author, more patterns are needed, especially for integration of business and scientific workflows. The author suggests addition of the following:

- consideration of a human behaviour as a service in the workflow. This would accept input data and produce output data as the other services and be accessible through various endpoints such as email, SMS etc.
- pattern: inititate_m_of_n pass through a given task if m out of n services associated with the task have fired. This can implement e.g. voting in a company if 2 out of 5 board of directors need to approve of the given resolution.

3.1.7. Distributed Management of Workflows using Clouds. Another aspect of the workflow management is how distributed the workflow execution can be. Many workflow management systems invoke distributed services but are centralised in nature. Distributing the execution engine can bring several benefits [10]:

- optimisation of data transfer costs between services/clouds as the managing party can be closer to the services/clouds if direct transfer is not possible (there are reasons for this e.g. not wanting to pass security credentials to one service/cloud to access another),
- no need for costly global synchronisation when executing the workflow ability to parallelize execution better.

3.1.8. Incorporation of Semantic Search. Semantic search for services is a challenge that is needed for automatic building of workflow applications that realise a goal defined by a user. First, out of knowledge acquired previously by the system, a graph of tasks needed to perform a given complex task (workflow) is built. Secondly, based on semantic and intelligent search methods [9], services capable of executing particular tasks are found and selected to optimise the given QoS goal and meet additional QoS constraints.

In the context of cloud utilisation (next to grid and on-site services) for workflow applications, semantic search can be useful for searching for:

- IaaS on which executables could be run alternatively to the services already assigned to workflow tasks (with proper QoS values such as compute power resulting in certain execution time of the service and the cost of the given IaaS),
- particular SaaS that might perform the given task in the workflow (with proper QoS values such as the execution time and cost).

4. Solutions and Recommendations. This section presents some solutions and recommendations on how solutions to the challenges identified in the preceding section might be designed and implemented when

[27] `http://www.cloudorado.com/`

running distributed workflow applications using clouds. As an example, the existing BeesyCluster middleware and the workflow management system are used as the basis and extensions suggested by the author for these.

4.1. BeesyCluster Middleware and Workflow Management System. BeesyCluster [6] is a middleware that allows distributed users to access and use distributed resources. It offers WWW (Figure 4.1) and Web Service interfaces [8]. Users can manage resources such as clusters or servers as well as develop and use software installed there through system accounts on these resources. BeesyCluster allows single sign-on to use those multiple resources. Furthermore, applications, whether run on regular servers or clusters, can be published as BeesyCluster services to which BeesyCluster users or groups are granted privileges. Such services can be incorporated into the embedded workflow management system module (WfMS) [6]. The BeesyCluster WfMS allows modelling workflow applications as DAGs defined before with assignment of services to workflow tasks (Figure 4.2). Such services may be either own services developed by the workflow author or made available by others, from either other clusters or grids. The WfMS allows monitoring statuses of previously run workflow applications (Figure 4.3). The following section shows how the aforementioned solutions, including incorporation of services from clouds can be used in this WfMS.

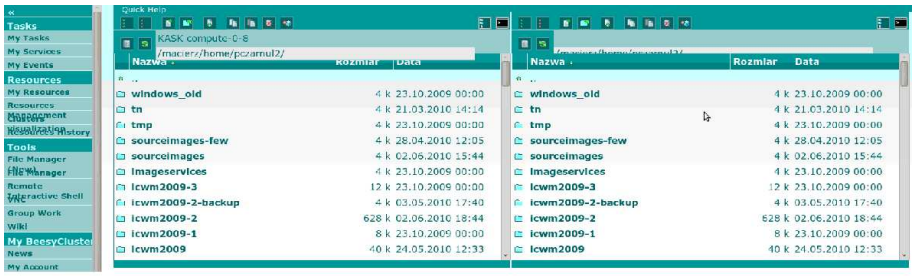

Fig. 4.1: BeesyClusters WWW interface.

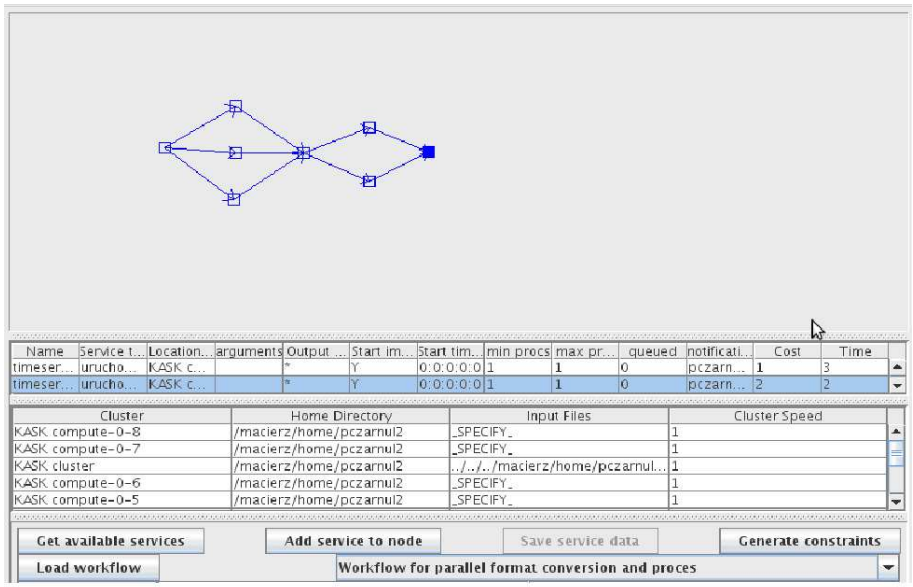

Fig. 4.2: BeesyClusters WfMS editor.

4.2. BeesyCluster Architecture and Extensions for Cloud Computing. The following extensions are proposed to solve the challenges identified in the previous section: integration of cloud-based services into

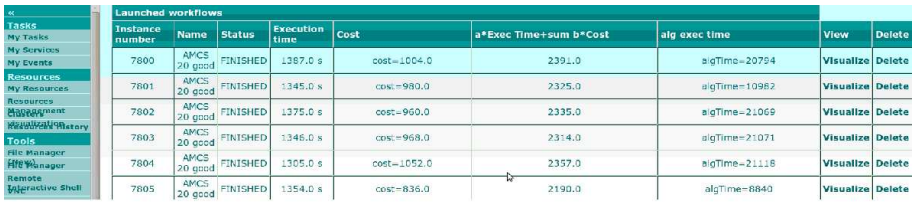

Fig. 4.3: Results of workflows runs.

workflow applications discussed as extensions of the already implemented solution (Figure 4.1):

- architecture and plugins the architecture of BeesyCluster is easily extended with support for cloud providers as follows:
  - similarly to cluster queueing systems (stored in table ra_qsystem), grid middlewares are added (stored in table ra_gmiddleware) and cloud interfaces (stored in table ra_cloudapi),
  - similarly to particular clusters (stored in table ra_cluster), grid middlewares are added (stored in table ra_gridmiddleware) and clouds (stored in tables ra_cloud _iaas, ra_cloud_paas, ra_cloud_saas). Proper interfaces are referenced in tuples from these database tables.
- workflow scheduling using clouds. In BeesyCluster, as in many other workflow management systems, services are distinguished each of which is assigned to a workflow task that it can perform. Each service has executable code associated with it along with QoS parameters at least the execution time and the cost. In BeesyCluster, a user may deploy a service on a user account of a cluster or a server available to them or make an executable available for download (which also constitutes a service i.e. that the executable can be downloaded for a fee). In the latter case, it is possible to find a matching (in terms of the architecture) environment (such as IaaS) for the executable to upload and run. In this case, two possibilities appear for clouds registered in BeesyCluster:
  - IaaS publish several services (corresponding to one executable run on the given IaaS) i.e. options of running the executable on the IaaS with various sets of parameters such as memory size, compute power and obviously corresponding execution time and cost.
  - SaaS publish a SaaS service as a BeesyCluster service with proper cloud access as defined above.
- services in BeesyCluster - in order to maintain compatibility with the existing service subsystem [6], IaaS and SaaS services are registered in the system along with standard cluster based ones. This allows incorporation of these services into the already available static and dynamic scheduling methods. Thus, the many algorithms available in BeesyCluster, in particular ILP-based, genetic algorithm, divide-and-conquer, GAIN can be used for running workflow applications on cluster, grid and cloud resources at the same time. Out of these, the genetic approach, in which a chromosome represents assignment of particular services to workflow tasks and launching services at particular moments in time, is the most general in terms of QoS goal and constraints. It does not impose e.g. linearity on constraints such as ILP. It can be noted that the algorithms developed for matching of services based on compatibility of their inputs and outputs [10] can be reused in the proposed environment as well. Information on success or failure of service invocation is used to update information of compatibility of inputs and outputs of pairs of services.
- semantic and intelligent search mechanisms developed in [9] can then be used to search for:
  - IaaS resources and registration for particular executables of interest (searching for IaaS cloud providers),
  - SaaS software and registration as BeesyCluster services (searching by name, input, output data formats). It would certainly be useful if SaaS is described in a format that makes such search easier e.g. in OWL-S with description of the service in appropriate fields.
- patterns including human interaction extension of the regular DAG:
  - control patterns with a service corresponding to a human interaction (such a service is invoked by sending an email to a specified address with attachments that correspond to input data; it then finalises its execution by invoking a proper Web Service in BeesyCluster returning data), more human actors can be registered as several services and one is selected based on the history e.g.

availability, learnt response time, cost (if defined) etc.
    * implementing the initiate_m_of_n services in this case, n services are launched in parallel and as soon as m have returned, their output is concatenated and passed further,
  – data patterns with:
    * operations on either individual data items (represented as files in BeesyCluster) or invoking the service for all data files at once,
    * data partitioning among forked, following tasks with several possibilities: send all data to all tasks, partition the data to minimise the QoS goal [6], denote what data needs to be sent to all tasks and partition the rest, denote manually what data is sent to what task.
- workflow model for both business and scientific workflow applications BeesyCluster supports this by allowing easy registration of new resources (be it university clusters, company servers or public clouds). Additionally, the basic description of each service with its execution time and cost can be easily extended by the author of the workflow with any additional QoS metrics, specifically suitable for either business or scientific uses. This is then immediately reflected in the constraint model and considered during optimisation [6]. This means that particular QoS metrics can vary from workflow to workflow. Additionally, it is worth to note that in BeesyCluster there is a natural (by design) market of services. Such services (either scientific or business) can be put on auction and BeesyCluster users can bid to win a privilege to use the given service for a fee. So, on the one hand BeesyCluster has support for low-level HPC queueing, on the other it inherently support a business oriented market of such services. As mentioned above, it is proposed by the author that human services are introduced into the workflow. However, different than in [26], the author proposes that these can be services. This means that several human services can be assigned to a workflow task, out of which one can be selected. The auction functionality applies to any type of BeesyCluster service and could also be used for human services (e.g. for consulting). Furthermore, workflow patterns can be used in such a workflow.
- distributed execution on cloud enabled systems using software agents after the various types of services have been deployed as proposed, both workflow execution methods available in BeesyCluster can be used to manage the execution of the workflow:
  – centralised using a Java EE server on which BeesyCluster and its WfMS are deployed [6],
  – distributed using software agents [10] in which agents locations are optimised to minimise communication costs and consequently the workflow execution time.
- reliability and availability and runtime monitoring of resources similarly to the work on dynamic monitoring of service availability [6], the very same method can be used in the extended version using cloud services. In this case, though, the author proposes to extend monitoring with functions that will convert the measured metrics to a normalised [0,1] quality range where 1 denotes best quality. This will consider not only the metrics, but also their derivatives. As an example:
  – lower execution time of an HPC service results in better quality,
  – higher reliability results in better quality,
  – lower fluctuations of availability of a cloud results in better quality,
  – lower fluctuations of prices of a cloud results in better quality.

Figure 4.4 presents the layered architecture of the proposed solution. Figure 4.5 presents execution of a workflow superimposed on the architecture of the integrated solution. The BeesyCluster middleware allows users (U0, U1 and U2) to define and manage workflow applications through the embedded WfMS. Execution of the workflow is delegated to a group of software agents by passing the description of the workflow in BPEL and proper credentials to access external cluster, grid and cloud systems. Agents vote which one is responsible for execution of which parallel paths of the workflow. The WfMS can execute ready-to-use services installed on clusters, grids or clouds (SaaS) and find resources (IaaS) for executables available in clusters. In one workflow, human services can allow to control flows. Furthermore, human services are treated as all others including QoS parameters, runtime monitoring and reselection. If one human service fails (the person does not respond or has failed to respond in a given time frame), control is passed to an alternative human service as shown in Figure 4.5. Agents report back to the workflow execution module on statuses and gathered QoS information for cluster, grid, cloud services (including human) which are updated in a service directory to be used in further scheduling. The embedded auction module can use this information to allow bidding and winning services by BeesyCluster users.

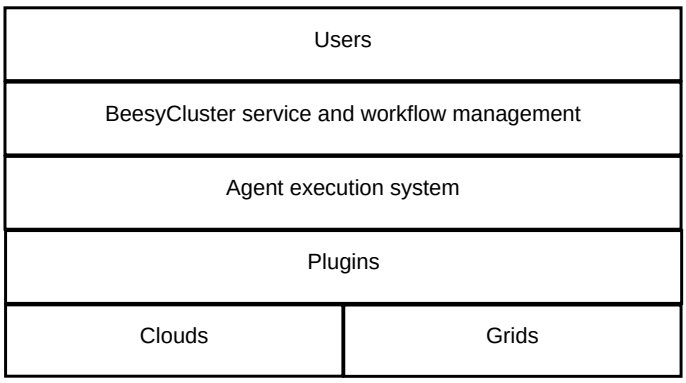

Fig. 4.4: Layers of the proposed architecture for running workflow applications on clouds and grids at the same time

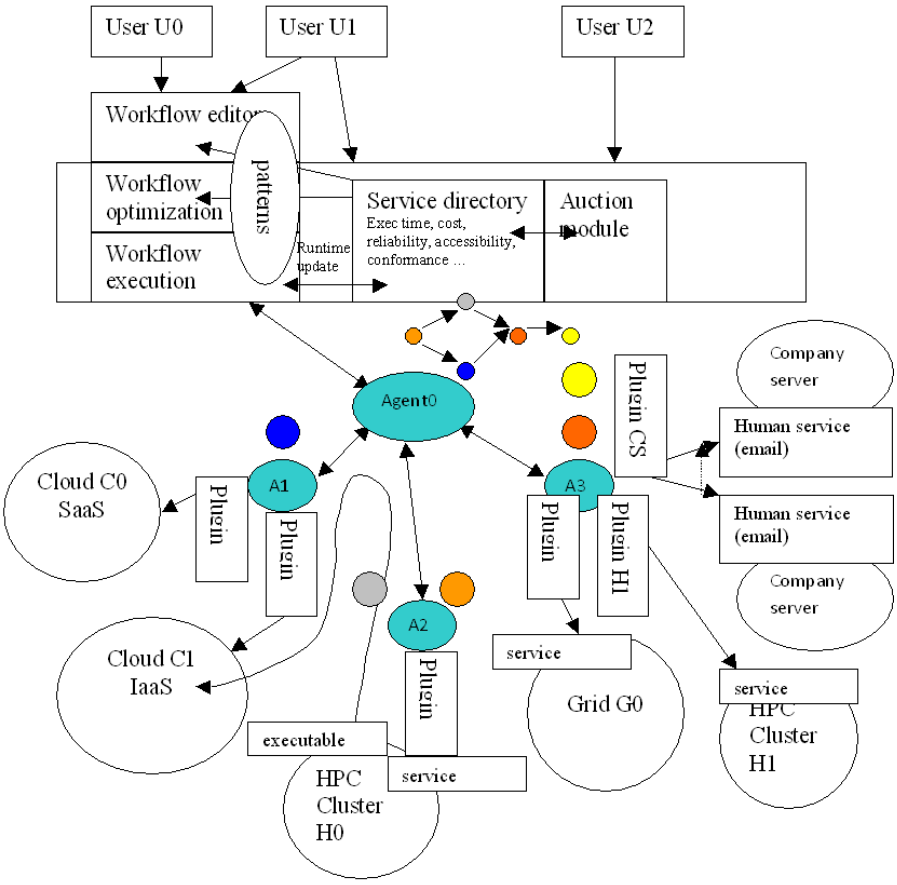

Fig. 4.5: BeesyCluster and agents executing a workflow application

5. Conclusions and Future Research Directions. The paper summarised the current developments in workflow management using on-site, grid and cloud computing with focus on challenges that appear in particular when engaging clouds in workflow management. The following challenges were discussed: preparation of using

a cloud and setting up (in case of private clouds), complex software stack for workflow management using a diverse set of resources (sky computing), data management issues, integration of features for both business and scientific services in one workflow, need for a directory of cluster, grid and cloud based services with monitoring of not only values of particular QoS metrics but also their changes in time (derivatives), reusable patterns for business and scientific uses, distributed execution of workflows composed out of distributed services, integration of semantic search into cloud-enabled workflow applications.

Secondly, suggestions and recommendations were provided on how the listed challenges can be implemented in the BeesyCluster middleware and workflow management system.

While workflow applications span more and more types of systems (local, embedded, cluster, grid and gradually cloud), the author predicts that further integration of workflow processing on these distinct types of systems will progress. For instance, dynamic ad-hoc discovery of services from mobile devices and automatic integration of services using semantic search for both business and compute-intensive tasks in one workflow is yet to follow on a global scale. This can be integrated with common uses (patterns) that should be developed for such integrated systems. A real world application could be activation of a mobile device when speaking to a foreigner in own language, uploading a translator application and voice to an IaaS cloud and conversion to a different language.

REFERENCES

[1] M. Abouelhoda, S. Issa, and M. Ghanem, *Tavaxy: Integrating taverna and galaxy workflows with cloud computing support*, BMC Bioinformatics, 13 (2012).

[2] A. Bala and I. Chana, *A survey of various workflow scheduling algorithms in cloud environment*, in IJCA Proceedings on 2nd National Conference on Information and Communication Technology NCICT, New York, USA, 2011, Foundation of Computer Science, pp. 26–30.

[3] S. Bharathi and A. Chervenak, *Scheduling data-intensive workflows on storage constrained resources*, in Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, WORKS 09, New York, NY, USA, 2009, pp. 1–10. DOI http://doi.acm.org/10.1145/1645164.1645167.

[4] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy, *Task scheduling strategies for workflow-based applications in grids*, in CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, vol. 2, May 2005, pp. 759–767.

[5] S. Chin, T. Suh, and H. Yu, *Adaptive service scheduling for workflow applications in service-oriented grid*, The Journal of Supercomputing, 52 (2010), pp. 253–283.

[6] P. Czarnul, *Modeling, run-time optimization and execution of distributed workflow applications in the JEE-based BeesyCluster environment*, The Journal of Supercomputing, (2010), pp. 1–26. 10.1007/s11227-010-0499-7, http://dx.doi.org/10.1007/s11227-010-0499-7.

[7] P. Czarnul, *Modelling, optimization and execution of workflow applications with data distribution, service selection and budget constraints in BeesyCluster*, in Proceedings of 6th Workshop on Large Scale Computations on Grids and 1st Workshop on Scalable Computing in Distributed Systems, International Multiconference on Computer Science and Information Technology, October 2010, p. 629 636. IEEE Catalog Number CFP0964E.

[8] P. Czarnul, M. Bajor, M. Fraczak, A. Banaszczyk, M. Fiszer, and K. Ramczykowska, *Remote task submission and publishing in beesycluster : Security and efficiency of web service interface*, in Proc. of PPAM 2005, Springer-Verlag, ed., vol. LNCS 3911, Poland, Sept. 2005.

[9] P. Czarnul and J. Kurylowicz, *Automatic conversion of legacy applications into services in beesycluster*, in Proceedings of 2nd International IEEE Conference on Information Technology ICIT'2010, Gdansk, Poland.

[10] P. Czarnul, M. R. Matuszek, M. Wójcik, and K. Zalewski, *Beesybees: A mobile agent-based middleware for a reliable and secure execution of service-based workflow applications in beesycluster*, Multiagent and Grid Systems, 7 (2011), pp. 219–241.

[11] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny, *Pegasus : Mapping Scientific Workflows onto the Grid*, in Across Grids Conference, Nicosia, Cyprus, 2004. `http://pegasus.isi.edu`.

[12] E. Deelman, G. Singha, M.-H. Sua, J. Blythea, Y. Gila, C. Kesselmana, G. Mehtaa, K. Vahia, G. B. Berrimanb, J. Goodb, A. Laityb, J. C. Jacob, and D. S. Katzc, *Pegasus: A framework for mapping complex scientific workflows onto distributed systems*, Scientific Programming, 13 (2005). IOS Press.

[13] A. Gao, D. Yang, S. Tang, and M. Zhang, *Web service composition using integer programming-based models*, in e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on, Sch. of Electr. Eng. & Comput. Sci., Peking Univ., Beijing, China, October 2005, pp. 603–606.

[14] S. K. Garg, R. Buyya, and H. J. Siegel, *Time and cost trade-off management for scheduling parallel applications on utility grids*, Future Generation Computer Systems, 26 (2010), pp. 1344 – 1355.

[15] S. Gogouvitis, K. Konstanteli, D. Kyriazis, G. Katsaros, T. Cucinotta, and M. Boniface, *Achieving Real-Time in Distributed Computing: From Grids to Clouds*, IGI Global, 2012, ch. Workflow Management Systems in Distributed Environments, pp. 115–132. 10.4018/978-1-60960-827-9.ch007.

[16] C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, and J. Good, *On the use of cloud computing for scientific workflows*, in IEEE Fourth International Conference on eScience '08, 2008, pp. 640–645. doi: 10.1109/e-Science.2008.167.

[17] G. Juve and E. Deelman, *Grids, Clouds and Virtualization*, Springer, 2010, ch. Scientific Workflows in the Cloud, pp. 71–91.
[18] N. Kaur, T. Aulakh, and R. Cheema, *Comparison of workflow scheduling algorithms in cloud computing*, International Journal of Advanced Computer Science and Applications, 2 (2011).
[19] K. Keahey, M. O. Tsugawa, A. M. Matsunaga, and J. A. B. Fortes, *Sky computing*, IEEE Internet Computing, 13 (2009), pp. 43–51.
[20] B. Linke, R. Giegerich, and A. Goesmann, *Conveyor: a workflow engine for bioinformatic analyses*, Bioinformatics, 27 (2011), pp. 903–11.
[21] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, and Y. Zhao, *Scientific Workflow Management and the Kepler System*, Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, (2005).
[22] S. Majithia, M. S. Shields, I. J. Taylor, , and I. Wang, *Triana: A Graphical Web Service Composition and Execution Toolkit*, in IEEE International Conference on Web Services (ICWS'04), IEEE Computer Society, 2004, pp. 512–524.
[23] S. Pandey, D. Karunamoorthy, and R. Buyya, *Workflow Engine for Clouds*, Wiley Press, New York, USA, 2011, ch. Cloud Computing: Principles and Paradigms. ISBN-13: 978-0470887998.
[24] C. Patel, K. Supekar, and Y. Lee, *A QoS Oriented Framework for Adaptive Management of Web Service based Workflows*, in Proceedings of the 14th International Database and Expert Systems Applications Conference (DEXA 2003), LNCS, Prague, Czech Republic, September 2003, pp. 826–835.
[25] R. Sakellariou, H. Zhao, E. Tsiakkouri, and M. D. Dikaiakos, *Scheduling workflows with budget constraints*, in in Integrated Research in Grid Computing, S. Gorlatch and M. Danelutto, Eds.: CoreGrid series, Springer-Verlag, 2007.
[26] M. Sonntag, D. Karastoyanova, and E. Deelman, *Bridging the gap between business and scientific workflows*, in Proceedings of e-Science 2010, Brisbane, Australia, 2010.
[27] M. Vizard, *Workflow management moves to the cloud.* IT Business Edge, Retrieved June 27, 2012, from http://www.itbusinessedge.com/cm/blogs/vizard/workflow-management-moves-to-the-cloud/?cs=42242, 2012.
[28] J. Vckler, G. Juve, E. Deelman, M. Rynge, and G. Berriman, *Experiences using cloud computing for a scientific workflow application*, in Proceedings of 2nd Workshop on Scientific Cloud Computing (ScienceCloud 2011), 2011.
[29] M. Wieczorek, A. Hoheisel, and R. Prodan, *Towards a general model of the multi-criteria workflow scheduling on the grid*, Future Generation Computer Systems, 25 (2009), pp. 237 – 256.
[30] M. Wieczorek, R. Prodan, and T. Fahringer, *Scheduling of scientific workflows in the askalon grid environment*, SIGMOD Rec., 34 (2005), pp. 56–62.
[31] Yingchun, X. Li, and C. Sun, *Cost-effective heuristics for workflow scheduling in grid computing economy*, in GCC '07: Proceedings of the Sixth International Conference on Grid and Cooperative Computing, Washington, DC, USA, 2007, IEEE Computer Society, pp. 322–329.
[32] J. Yu and R. Buyya, *A taxonomy of workflow management systems for grid computing*, Journal of Grid Computing, 3 (2005), pp. 171–200.
[33] J. Yu and R. Buyya, *A budget constrained scheduling of workflow applications on utility grids using genetic algorithms*, in Workshop on Workflows in Support of Large-Scale Science, Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing (HPDC 2006), Paris, France, June 2006.
[34] ———, *Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms*, Scientific Programming Journal, (2006). IOS Press, Amsterdam.
[35] J. Yu, R. Buyya, and K. Ramamohanarao, *Workflow Scheduling Algorithms for Grid Computing*, Springer, 2008, ch. Metaheuristics for Scheduling in Distributed Computing Environments. in Metaheuristics for Scheduling in Distributed Computing Environments, ISBN: 978-3-540-69260-7, Berlin, Germany, `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.144 .7107`.
[36] J. Yu, R. Buyya, and C.-K. Tham, *Cost-based scheduling of workflow applications on utility grids*, in Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), IEEE CS Press, Melbourne, Australia, December 2005.
[37] D. Yuan, Y. Yang, X. Liu, and J. Chen, *A cost-effective strategy for intermediate data storage in scientific cloud workflow systems*, in Proceedings of IPDPS 2010, 2010, pp. 1–12.
[38] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Sheng, *Quality driven web services composition*, in Proceedings of WWW 2003, Budapest, Hungary, May 2003.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 27, 2012
Accepted: Jan. 07, 2013

Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 339–350. http://www.scpe.org
ISSN 1895-1767

AN ALGORITHM FOR TRADING GRID RESOURCES IN A VIRTUAL MARKETPLACE

BENJAMIN AZIZ*

Abstract. This paper presents an algorithm for trading resources in Grids. Resource description includes main technical attributes of a resource, such as processing power, memory capacity, etc., as well as a price. Trading is performed in a marketplace where providers' resources are matched with consumers' demand by means of auction mechanisms. The matching algorithm follows a strategy where a consumer's demand is matched with providers meeting the technical requirement and the price closest to the one offered by the consumer.

Key words: Trading algorithms, Virtual Marketplace, Grid, Scheduling

1. Introduction. All the advantages provided by utility computing, grid computing and virtualization do not only enable the execution of computationally intensive, scientific applications but also allow commercial customers to use the power of such a Grid to exectute their applications quickly, effectively and efficiently. However, there are many different kinds of users such as SMEs, large enterprises, academia, etc., distinguishing themselves in the amount of budget, urgency of their application, and quality of service expectations. For example, users who require the termination of the execution of their application within specific period of time, are willing to pay a higher price than other users. There exists a wide variety of related market systems which are based on fixed prices, bartering, negotiations or auction models for leasing Grid contracts. These systems include GridEcon [1], SORMA [15], BREIN [7], BEinGRID [19], Edutain@Grid [6] and GRIA [21]. However, very few efforts have been made to fully specify the design of a market that is tailored for the Grid resources and services.

A Virtual Marketplace of Resources (VMR) is a marketplace, which comprises all the functionality for leasing of computational services for a time period so as to use Grid resources effectively and efficiently. A VMR allocates Grid resources according to their specifications to applications as a means of meeting performance goals described by the application provider to the available resources. The VMR system architecture we consider here was developed earlier in the context of project XtreemOS [23].

The main contribution of this paper is to introduce a marketplace trading algorithm that facilitates the commercialisation of Grid resources, where a provider is capable of listing the Grid resources, and buyers demand the required computing resources for their applications on the basis of utility (e.g. price/number of resources/time). The VMR algorithm offers a public Internet market that would be open to registered users to buy and sell computing resources. The ability to utilize remote Grid computing platforms frees both the provider and the buyer from the need to own or acquire the necessary computational infrastructure. Furthemore, the marketplace system will provide an infrastructure that will allow end-users not only to consume but also to sell services and resources on the Grid. Therefore, creating a new economy in which all users can actively participate. VMR offers a solution to both the high cost of ownership and the fluctuating usage patterns of computing capacity.

The VMR system facilitates the creation of self-governing collections of providers and buyers that make resource allocation decisions strictly based on current price/resource availability. Providers and buyers act autonomously to improve their own standing in a market. The price and resource specifications mentioned by the provider/buyer are their own choice and can adapt any strategy/technique as conditions change. Buyers usually want to pay the least amount possible for the resources needed to execute their application. Providers, on the other hand, wish to generate greater and greater revenue and larger profits from their offered resources.

VRM considers a fixed price approach instead of performance management approaches based on Service Level Agreements (SLAs) and utilization. Once the match has been allocated between a resource and and application, the published price has to be paid by the buyer to the provider after the execution of the application. Once the payment has been made the result of the execution is sent to the buyer.

This paper is organised as follows: Section 2 presents the computing resource exchange related work. In Section 3, we give an abstract view of the VMR solution. Section 4 mentions how the demands and offers are be described and when the trading algorithm is being activated. Section 5 describes the trading algorithm,

*School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom(benjamin.aziz@port.ac.uk).

and Section 6 explains how the trading algorithm is being executed with the help of an example, and finally, Section 7 concludes the paper giving directions for future work.

2. Related work. Several research systems [3] have explored the use of different economic models for trading resources to manage resources in different application domains: CPU cycles, storage space, database query processing, and distributed computing. Despite the fact that there are a few commercial providers of utility computing (e.g. Amazon [18], HP [10], IBM [9], Google [8], Sun [20]), these providers (being commercial) offer their resources at a relatively expensive cost. If compute resource users (providers and buyers) accept and trust the computing resource exchange for executing their trades, it will increase the supply of computing resources in the market. Consequently, computing resources price will decrease and become affordable to a low budget enterprises. Two open source systems SORMA [14] and GridEcon [1] have been developed in order to attract customers to computing resource exchanges.

SORMA [15] uses self-organizing resource management system to develop methods and tools for an efficient market-based allocation of resources. SORMA provides a flexible market infrastructure, which can access resources over different virtualisation platforms and enable different resource managers to plug in the market. SORMA follows the bottom up strategy, means it first define the market mechanisms for trading the resources and then design the appropriate middleware components for brokering, accounting, and charging. SORMA is being developed to offer the possibility of loosely integrating emerging Grid markets.

On the other hand, GridEcon [2] provides computing resource exchange for commoditised computing resoruces by offering a set of services that could help new users to accept the computing resource exchange concept. GridEcon is a computational resource auctioning system built upon a bid matching algorithm. Offers submitted by the resoruce buyers and providers are matched to execute the application on the cheapest available resource. GridEcon follows the top down strategy, means it first identify the kind of higher-level goods that applications would like to obtain in a commercial Grid environment and then defines the appropriate business model.

While SORMA focuses on the openness of decentralised complex service markets and GridEcon addresses an exchange for basic Computing resources, VMR is a compliment to these systems as VMR is an auctioning system that provides decentralised computing resource exchange to access resources over different platforms. VMR is independent of any underlying Grid middleware of the platform. VMR matches demands and offers and executes them against each other only after verifying the terms and conditions defined by the users.

Other more recent works have also addressed the problem of establishing a marketplace of Grid/Cloud resources. For example, in the Polish Agents in Grid (AiG) project [22], software agents are used to provide a meta-level Grid middleware where economic models can be established based on Service Level Agreement (SLA) representations of the Grid resources and clients. In this middleware, owners can make their resources available and clients can commission those resources for the execution of jobs, after SLAs of both sides are negotiated.

In [24], the authors propose also a multi-agent system for carrying out automated negotiations in any service-oriented environment including Grids. Similarly, in [11], the authors use the cost of electrical power as the unit of cost in a model of scheduling they propose for environments of distributed servers. This approach provides a more concrete realisation of the concept of cost than in our case and the case of other models.

Business-oriented large-scale systems, such as Clouds, have also adopted SLA-based trades. These include for example OPTIMIS [16], mOSAIC [13] and Cloud4SOA [4]. In this paper however, we stay within the scope of marketplace research carried out in the context of Grid system.

3. The Concept of a Virtual Market Place. At its heart, a VMR facilitates the commercialization of Grid resources on-demand through a virtual marketplace of computational resources, where a seller is capable of listing the Grid resources, and buyers can ask/bid dynamically for required computing resources for their applications. VMRs assume that resources are available in Grids based on various technologies (e.g. Globus-based, gLite-based Grids etc.). One such VMR was developed within the scope of project XtreemOS [5, 12]. XtreemOS aimed at building a Grid-based distributed operating system that provided a single abstraction of physical hardware and software services offered by a collection of standalone Linux operating systems to users within a Grid.

Such VMR system aims at providing a computational resources auctioning system built upon a dynamic bid matching algorithm tailored specifically for the trading of computing power. It helps both consumers and providers of computational resources to use the resources efficiently so as to maximize the economical benefits and minimize the idle time for them. The XtreemOS VMR was developed to integrate into a single framework three key features:

- *Interoperability* is achieved by using a standard programmable interface, the Simple API for Grid Application (SAGA) [17], to bridge the gap between existing Grid middleware and application level needs. The same system could run on any Grid platform (e.g. XtreemOS, gLite etc.), or interoperate on Grids using resources from other platforms.
- *Cost saving for end users* is guaranteed by allocating the applications to the more economical resource(s), following policies defined by the end users.
- *Dynamic scheduling* is achieved through the virtual marketplace, which implements scheduling and trading algorithms that allocates applications following classical performance parameter as well as the cost of resource usage.

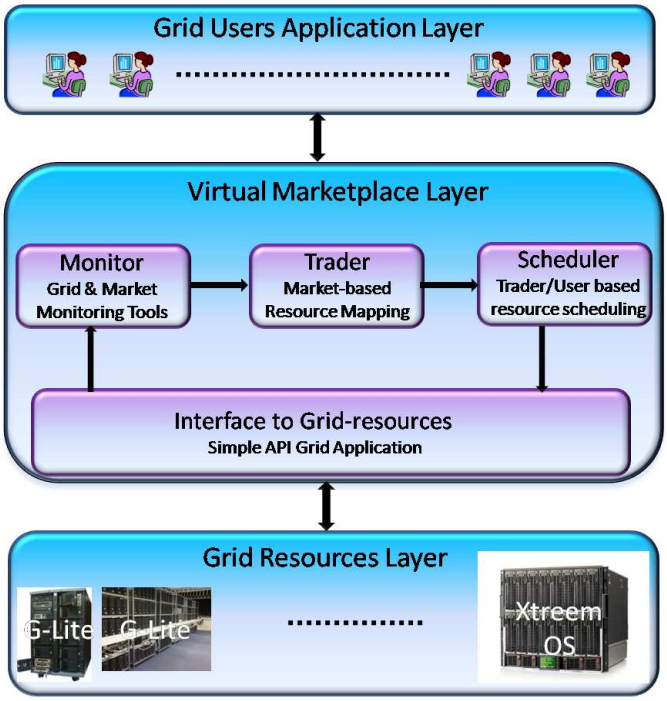

Fig. 3.1: Abstract view of VMR [23].

Figure 3.1 (originally from [23]) illustrates a logical layered architecture of the virtual marketplace defined in XtreemOS. This architecture represents the entities and their dependency to other entities, where the flow of information or control is depicted by arrows. An arrow from an entity X to an entity Y means that X sends information to Y or passes control to Y. Our system offers the mechanisms for deploying and executing the application (e.g. automatic deployment, execution monitoring, and hardware resource discovery) for the business processes to purchase the resources on the Grid. Implementing such a business model requires at least the following basic roles, which belong to three layers: the Grid users application layer, the Virtual marketplace layer, and the Grid resources layer.

Grid users application layer: This layer allows end-users (scientist, chemist, physician etc.) to submit applications to the deployed resources. We consider Grid application to be a collection of work items to solve a certain problem or to achieve desired results using the Grid infrastructure. Grid applications can be scientific, mathematical, academic problems or the simulation of business scenarios, like stock market development, that require a large amount of data as well as a high demand for computing resources in order to calculate and handle the large number of variables and their effects.

Virtual marketplace layer: This layer implements monitoring, trading and scheduling services so as to utilize the available Grid resources efficiently and exploit the benefits of the interoperability and scalability of the Grid platform. This layer consists of four main components:

- *Monitor* implements the monitoring/reporting techniques, which monitors resources and reports

changes such as dynamic re-allocation of resources, according to changes generated from evolution in the resource market,execution status of submitted applications, etc. Monitoring is achieved by either direct or indirect capture of resource status and pre-defined events. The indirect interface uses logs generated at run-time by the Grid infrastructure. The direct interface is a portal collecting dynamically events generated by monitoring services associated to the Grid infrastructure.

- *Trader* implements the trading algorithms that depends on criteria such as cost, processing power, execution time or resource availability. It is also responsible for sending notifications to users about the status of their request. For example, inform a bidder whether the bid is winning or not.
- *Scheduler* schedule the application on to the selected Grid resource. Scheduling of the end users application is done on to the selected resource by following the analysis provided by the trader.
- *Interface to Grid resources* provides simple access for distributed systems and abstractions for applications and thereby address the fundamental application design objectives of interoperability across different infrastructure. It also supports job submission and data management (efficient data access, data replication, streaming of data, etc.).

Grid resources layer: This layer consist of the resources (server, storage and network) used to execute the end users applications. The submitted application is executed on the selected Grid resources and result is sent back.

4. VMR Marketplace Mechanism and Matching module. In our marketplace, Buyers and providers interact through the marketplace by means of the broker, in order to lease/offer Grid resources. Providers are indifferent regarding how their machines will be consumed in the market, i.e. what kind of needs the consumers want to accommodate by leasing the providers' resources. The providers solely offer their resources and it is the responsibility of the marketplace to match them with the consumers' demand by means of a market mechanism and a matching algorithm.

A buyer's order is specified by means of the total number of resources (with its required specification) that must be made available up to a specific time interval, so that a certain computationally-intensive task is executed in time. All parties publicly announce the maximum price they are willing to buy for and the minimum price they are willing to sell for. All buyers should mention the maximum price they are willing to buy for and all the providers should mention the minimum price they are willing to sell for the leasing of resources in a specified time interval. These prices, resource information, participants' information are recorded and put in a database.

Below we define the demand (resp. the offer) that the buyers (resp. the providers) submit to the VMR in order to express their services, which comprises of computational elements that are made available in a time interval whose start and end time are specified upon submission and are not flexible.

A demand describes the buyer's requirements. Demand is specified as:

- (1) R_b - The resource specification (processing power, hard disk, RAM, OS, etc.),
- (2) Q_b - The number of resources that are demanded,
- (3) S_b - The start time of the interval for using the resources,
- (4) D_b - The time duration, for which resources are needed,
- (5) P_b - The price expressed in £for use of one resource/min, and
- (6) Et_b - The expiration time of the demand.

An offer describes the resources posted by the providers. Offer is specified as:

- (1) R_p - The resource specification (processing power, hard disk, RAM, OS, etc.),
- (2) Q_p - The number of resources that are available,
- (3) S_pStp - The start time of the interval when the resources are available,
- (4) D_p - The time duration, for which resources are available,
- (5) P_p - The price expressed in £for use of one resource/min, and
- (6) Et_p - The expiration time of the offer.

When more than one offer/demand is added to the queue at same time than sort according to submission expiry time if two same then sort according to the start time.

Buyers/providers orders will not be immediately fulfilled unless there is a previously posted compatible reciprocal demand/offer. All the compatible trades i.e., when the buyers demand price exceeds the providers offer price for a match between an application requirements and resoruce specification, are immediately executed. If no compatible reciprocal offers/demands are available, the offer/demands remain in the respective queue until they are matched in the future or expire.

The matching module is invoked whenever a demand or an offer is submitted to the VMR. The rationale behind the matching module is summarized as follows:

- a) Demands are completely satisfied, i.e. there are never remainder demands, pieces of the same demand that are still pending. This is not the case for offers, which can be partly matched in order to serve demands.
- b) Each demand is served by one provider.
- c) Matching solution ensures that the demanded resources are allocated throughout the service time interval (application's demanded duration), so that the resources switching is avoided over time.

Matching module activates in the following two events:

- 1) A new demand is submitted by the buyer: Matches candidates for a demand (whose price is P_b) only those offers (whose price P_p) where $P_b \geq P_p$ holds. Therefore, we omit examining higher price offers and try combining them with lower price offers, even if such combinations could in fact serve the demand.
- 2) A new offer is submitted by the provider: Matches candidates for an offer (whose price is P_p) only those demands (whose price P_b) where $P_b \geq P_p$ holds. Therefore, we omit examining lower price demands and try combining them with higher price demands.

The rationale of the matching procedure is to provide the required coverage of a) the demand with the cheapest matching offer and/or b) the offer with the equal or higher matching demand by means of a matching algorithm. If a demand is matched fully then reservation of resources, accounting and computation of remainder offer that replace the original offer in the offer queue are performed; and the demand is removed from the demand queue and subsequently serviced. On the other hand, if an offer is matched fully then reservation of resource and accounting are performed; and the offer is removed from the offer queue and the demand is removed from the demand queue, which is subsequently serviced.

As a demand is always fully matched, this is not the case for an offer. Therefore, in general a fraction of an offer may be used to (partly) match and serve a demand, thus generating remainder offers. Thus, when an offer is matched partially, the reservation of resources, accounting and computation of remainder offer that replaces the original offer in the offer queue is performed; and the matched demand is removed from the demand queue and subsequently serviced.

It is the responsibility of the matching module to be invoked periodically, in order to compute matches and remove expired offers and demands from the offer/demand queue. The results of the matching procedure are subsequently passed to the scheduler and the accounting system of the market place.

5. Trading algorithm. The algorithm defines how demands (submitted by the buyers) and offers (submitted by the providers) are matched. VMR trading algorithm is based on market mode because it offers a control strategy that is computationally efficient, flexible in the face of emergent behavior, and makes visible to IT personnel mission-critical price-performance statistics that directly reflect the marketplace's ability to deliver infrastructure tailored to real business value. The market-model trader used in VMR is capable of trading any kind of resources (compute resources, storage resources etc.), as long as the resource's consumption requirements can be translated into the trader's key-value format.

Currently, we assume that demands should be fully served by resources of a single provider, however in the future, buyers may be allocated resources of multiple providers, as long as each of these is reserved for the buyers throughout the specified time interval. This assumption is imposed due to technological constraints, since there may be significant switching costs when shifting unfinished computing jobs between virtual machines of different providers within the service time interval.

Psuedo codes 5.0.1 and 5.0.2 next present the VMR trading logic that is executed when a new demand/offer is submitted. As a first prototype, a meaningful matching procedure for the VMR is to try matching a demand with the cheapest matching offers. In future other matching procedures can be plugged-in to do matching according to the company preferences/constraints such as buyer can specify the list of providers they would like to submit their application for execution.

The trading algorithm runs from scratch, whenever a new demand arrives or a new offer arrives to perform a matching between the offers and demands respectively. When an offer arrives that match a demand, three things have to be decided a) how much of it to use, b) where to place it and c) what to do if offer is not completely used. The solution we adopt is a) order the matching offers according to the demand's time constraints i.e., use till demand is completely fulfilled, b) place it meet the demand's deadline such that offer is divided in minimum

Algorithm 5.0.1 when a new demand is submitted by a buyer.

```
if offer_queue ≠ null then
  select the offer where R_b == R_p
  store the selected offers in ascending order of offer prices P_p
  Select the offers where P_b ≥ P_p
  if selected_offer_queue≠null then
    i ← 0
    while i < sizeof(selected_offer_queue[]) do
      if ((Q_p[i] ≥ Q_b) &&(D_p[i] ≥ D_b)) then
        begin_time= max(St_p[i], St_b)
        end_time=min(Et_p[i], Et_b)
        if (end_time-begin_time) ≥ D_b then
          if begin_time == St_p[i] then
            Computation start time S_t=begin_time
          else
            Computation start time S_t=end_time-D_b
          end if
        end if
        pass information (selected_offer_queue[i], S_t, D_b)to the scheduler
        if (D_p > D_b[i])||(Q_p > Q_b[i]) then
          calculate the remaining offer using Pseudo code 5.0.3.
          resubmit the new created offers
        else
          remove the offer from the offer queue
        end if
      else
        i++
      end if
    end while
  else
    put the demand in the demand queue
  end if
else
  put the demand in the demand queue
end if
```

chunks and c) if offer left with services/time to be used, remaining offer is calculated according to pseudo code 5.0.3 and resubmitted.

Trading is performed by means of an auction mechanism. The submitted demands and offers are placed in the demand queue and the offer queue respectively. If two or more orders at the same price appear in an allocated queue, then they are entered by time with older orders placed above the newer orders. Trader collects orders from buyers and providers and executes trades (makes a call) periodically to clear the market by matching buyers with providers. A demand/offer remains in the queue until it is allocated, removed due to its expiration time or removed by the submitted user. Resources are allocated for a specified amount of time that is required by the application and defined by the buyer.

The trading algorithm initially computes the candidate matches to demand by means of creating a matrix as shown in Figure 5.1. Each column of the matrix corresponds to a time slot (i.e. the time interval in which service can be provided). Each row corresponds to a provider that can offer service now, with the cheapest being on the top row. A cell of the matrix is marked if the provider can offer computing resources during this specific time slot, as shown in Figure 5.2.

Complexity of buyers and providers trading algorithm: Counting the total number of basic operations, those which take a constant amount of time in Psuedo code 5.0.1 followed by the while loop where the value of i changes every time through the loop (N + (N - 1) + $\cdot$ + 2 + 1 = $N(N+1)/2$), the total number

Algorithm 5.0.2 when a new offer is submitted by the provider.

if *demand_queue* $\neq$ *null* **then**
  select the demands where $R_b == R_p$
  store the selected demands in descending order of offer prices P_p
  Select the demands where $P_b \geq P_p$
  $i \leftarrow 0 j \leftarrow 0$
  while selected_demand_queue$\neq$null **do**
    if $((Q_p \geq Q_b[i])$ &&$(D_p \geq D_b[i]))$ **then**
      begin_time= max$(St_p, St_b[i])$
      end_time=min$(Et_p, Et_b[i])$
      if (end_time-begin_time) $\geq D_b[i]$ **then**
        selected_demands[j]=selected_demand_queue[i]
        calculate sorting_condition[j] = $D_b[i] * Q_b[i] * P_b[i]$
        j++
      else
        i++
      end if
    end if
  end while
  if selected_demands$\neq$null **then**
    Sort selected_demands in descending order of sorting_condition.
    Select the selected_demands[0]
    if $St_{selected\_demands[0]} > St_p$ **then**
      Computation start time S_t=$St_{selected\_demands[0]}$
    else
      Computation start time
      S_t=min$(Et_p, Et_{selected_demands[0]})$-$D_{selected\_demands[0]}$
    end if
    pass information (selected_offer_queue[i], S_t, D_b)to the scheduler
    Remove the demand from demand queue
  end if
  if $(D_p > D_b$[selected_demands[0]])||($Q_p > Q_b$[selected_demands[0]]) **then**
    calculate the remaining offer using Psuedo Code 5.0.3
    resubmit the new created offers
  end if
else
  put offer in offer queue
end if

of operations is equivalent to $O(N^2)$. As the runtime complexity of Psuedo code 5.0.2 is less than Psuedo code 5.0.1, we can say that Psuedo code 5.0.2 is more efficient than Psuedo code 5.0.1. However, both falls into $O(N^2)$ complexity class.

6. Implementation of Virtual Marketplace of Resoruces. To explain the trading algorithm, we assume that VMR has a demand queue as shown in Figure 6.1 and an offer queue as shown in Figure 6.2 with respect to the matching martix of Figure 5.1. For simplicity, the time in demand/offer queue is considered to be of same day and even the resource specifications is limited to the OS only. However, VMR uses the timestamp datatype for describing the time which enables the provider/buyer to specify the time from future dates. Similarly resoruce specification is not only limited to the OS, a provider/buyer can speciy the resoruce's hardware and software description along with the versions.

First case is when a new offer is submitted by a provider. For example, Provider P offers 8 XtreemOS resources for 5 hrs, starting at time 11:00, with offer price £0.03, and time limit 23:30. Following the Psuedo Code 5.0.2 buyers B_1, B_3, B_5 are selected and sorted in descending order of price from the demand queue, i.e., B_5, B_3, B_1. After comparing the price $P_{B_5}, P_{B_3}, P_{B_1} \geq P_p$ only demands from B_5 and B_3 are added to the

Algorithm 5.0.3 calculate the remainging offer.

if $S_t == St_p$ **then**
  if $Q_b < Q_p$ **then**
    Submit new offers for partial used resources as $(S_t + D_b, Et_p, P_p, D = D_p - D_b, Q_b)$ and totally unused resoruces as $(St_p, Et_p, P_p, D = D_p, Q = Q_p - Q_b)$
    (*Case of Fig. 5.2(1)*)
  else
    Submit new offer as $(S_t + D_b, Et_p, P_p, D = D_p - D_b, Q_p)$
    (*Case of Fig. 5.2(2)*)
  end if
else
  if $S_t + D_b == Et_p$ **then**
    if $Q_b < Q_p$ **then**
      Submit new offers for partial used resources as $(St_p, Et_p - D_b, P_p, D = D_p - D_b, Q_b)$ and for unused resoruces as $(St_p, Et_p, P_p, D = D_p, Q = Q_p - Q_b)$
      (*Case of Fig. 5.2(3)*)
    else
      Submit new offer as $(St_p, Et_p - D_b, P_p, D = D_p - D_b, Q_p)$
      (*Case of Fig. 5.2(4)*)
    end if
  else
    if $Q_b < Q_p$ **then**
      Submit new offers for partial used resources before use as $(St_p, St, P_p, D = D_p - D_b, Q_b)$, after use as $(S_t + D_b, Et_p, P_p, D = D_p - D_b, Q_b)$, and unused resoruces as $(St_p, Et_p, P_p, D = D_p, Q = Q_p - Q_b)$
      (*Case of Fig. 5.2(5)*)
    else
      Submit new offers for only partial used resources before use as $(St_p, St, P_p, D = D_p - D_b, Q_p)$ and after use as $(St + D_b, Et_p, P_p, D = D_p - D_b, Q_p)$
      (*Case of Fig. 5.2(6)*)
    end if
  end if
end if

selected_demand_queue. First quantity and duration required by the buyer B_5 is checked. Possible available duration is calculated by matching the start time and expiry time of buyer B_5 and provider P, which is similar to the Figure 6.3(6). As available duration is more than the required duration by B_5, the demand is added to the slected_demands queue and its sorting_condition is calculated (3*3*0.06) as 0.54. Then next demand from the selected_demand_queue is selected i.e., demand from buyer B_3 and same procedure the repeated as mentioned for the demand by buyer B_5. Now the selected demands B_5 and B_3 are sorted in descending order of the sorting condition, which conclude B_3's demand to be the matching demand for the P's offer. Time slot for the matching matrix is calculated (shown in Figure 5.1). Demand is removed from the demand queue.As provider P has offered quantity and duration is more than used by the demand of B_3, remaining offer is calculated using Pseudo code 5.0.3. As starting time S_t is calculated to be 19:00 which is neither St_p (11:00) nor Et_p (23:30), however Q_{B_3} (5) is less than the Q_P (8), the remaining offers (Figure 5.2(6)) are resubmitted to the VMR.

Second case is when a new demand is submitted by a buyer. For example, Buyer B demands for 3 Linux resources to be used for 2 hrs, starting at time 13:00, with demand price £0.05/resource/min, and time limit 23:00. Following the Pseudo Code 5.0.1, providers P_2 and P_5 offers the similar resources as required by the buyer B. P_2 and P_5 are sorted in ascedning order of price and placed in selected_offer_queue as both has $(P_{P_2}, P_{P_5}) \geq P_B$. First, the quantity and duration offered by P_5 is checked and possible available duration is calculated (which is similar to Figure 6.3(7)). As available duration is greater than the demanded duration, the match between P_5 and B is added to the matching matrix. Provider P_5 has offered more quantity and duration than used by the demand, remaining offers are calculated using the Pseudo code 5.0.3. As end time of the execution is equal to Et_{p_5}, the remaining offers (Figure 5.2(1)) are resubmitted to the VMR.

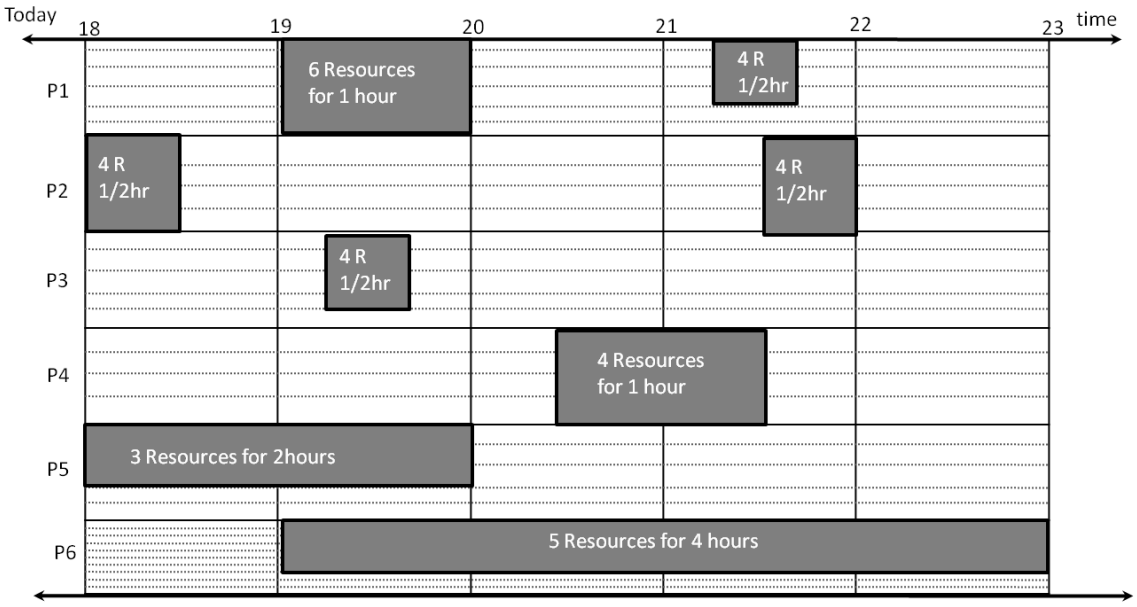

Fig. 5.1: Matching Matrix.

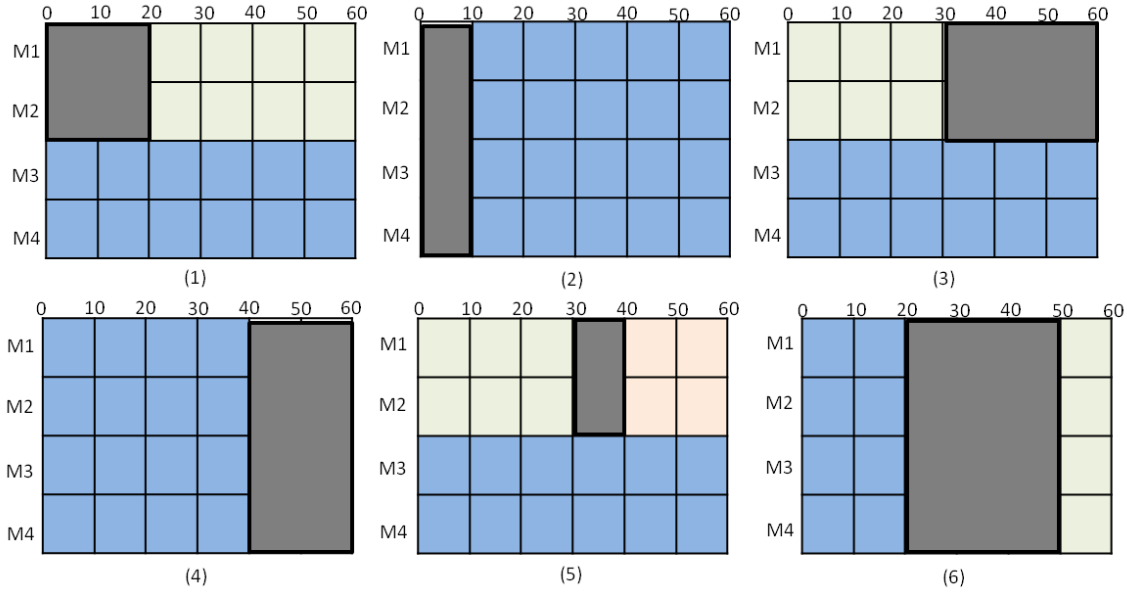

Fig. 5.2: Grey: Matching (demand and offer) possibilities, other colors: recalculated offers.

VMR GUI is a visualisation tool that provides an online marketplace for provider and buyers to publish their products and needs. Matches that have been made by the trading algorithm are also displayed.

The VMR GUI provides a portal for new users to register and check the current status of the market. Currently, the GUI displays the demands/ offers of buyers/providers and matches that has been made by the trading algorithm. GUI Development Language is English and database is MySQL. Resources from two different Grids (gLite and XtreemOS) are added to the database. To achieve interoperability when using resources from different platforms SAGA [17] is considered.

Buyers	Requirement	Qunatity	Duration	Price	Start time	Expiry time
B_1	XtreemOs	4	3	0.02	12:00	20:00
B_2	gLite	9	3	0.04	10:00	18:00
B_3	XtreemOs	5	4	0.03	15:00	23:00
B_4	Windows XP	3	2	0.05	18:00	23:00
B_5	XtreemOs	3	3	0.06	10:00	20:00

Fig. 6.1: A demand queue.

Providers	Requirement	Qunatity	Duration	Price	Start time	Expiry time
P_1	gLite	5	3	0.04	8:00	16:00
P_2	Linux	4	4	0.05	10:00	20:00
P_3	Windows XP	2	5	0.04	12:00	23:00
P_4	gLite	10	5	0.03	9:00	18:00
P_5	Linux	5	3	0.04	18:00	23:00

Fig. 6.2: An offer queue.

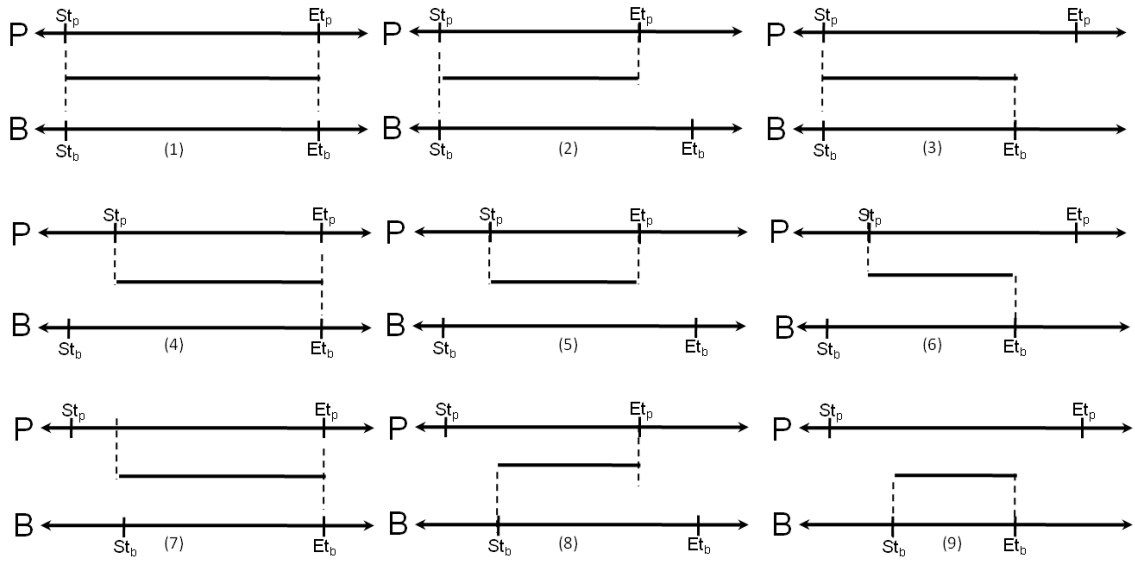

Fig. 6.3: Possibilities of demand and offer start and expiry time combinations.

Each resource allocation requires:

- *Resource specification*: This specification contains the application's requirement list. The list is used to specify which operating system, how much memory, which software, etc. are needed. Time Frame specification: This specification defines a window of time (in seconds) within which the allocation is to be considered valid. Applications execution start time, duration and deadline time should be specified by the buyer and similarly, the resource availability start time, duration and deadline time should be specified by the providers.
- *Periodic matching*: This time in seconds/minutes defines the interval at which VMR will execute the trading algorithm. A periodic cycle consists of the steps followed to process and execute trades. At the end of the trading cycle, compatible demands and offers are paired up. Demands and offers that are not converted into successful trades at the end of the trading cycle will wait in the queue till the next trading cycle, until cancelled or matched. The length of the cycle depends upon the type (CPU,

storage) and purpose or sector (a user community like research, bank) or can be fixed as in our first prototype to be 120seconds.

- *VMR Catalog*: It is an organized and searchable repository of resources, providers and buyers information. The information is composed of a variable length amount of metadata in the form of name-value pairs that describes performance requirements or capabilities. Each entry in the catalog represents a single entity (resource, provider, and buyer) and contains metadata that describes the entities' attributes, properties, performance requirements and functionality. The catalog can be searched by specifying a query composed of a set of metadata that must match the metadata of one or more entries to be included in the result set.

7. Conclusion. This paper presents an algorithm for trading resources in Grids. Resource description includes main technical attributes of a resource, such as processing power, memory capacity, etc., as well as a price. Trading is performed in a marketplace where providers resources are matched with consumers demand. The VMR solution presented in this paper answers questions such as "which Grid resource should be used that will minimize cost along with achieving efficient applications' execution time?", "how end-user can select Grid resources according to pre-defined policies, including cost policies?" and "how to achieve interoperability when using resources from different platforms?".

For future work, we would like to extend the algorithms to allow them to become more flexible incorporating other criteria that may affect the decision for selecting offers and demands. The presence of several factors, not just cost, would imply a trade-off-based algorithm, which will be able to provide compromises among the different criteria. For example, one such criterion could be matching the security requirements (privacy, assurance, or risk-based) as expressed by the policies of the buyers and providers. Usually, security comes at an increased cost. So, depending on the level of assurance provided by the provider, the buyer may be willing to pay more for better security.

The current form of the algorithm assumes that the buyer or the provider is a single entity or organisation. In the future, we would like also to consider federations of entities (buyers/providers) or virtual organisations. This would have implications on the underlying architecture, as it would imply some form of synchronisation or consensus among the different entities comprising the federation as to how the price of the offer or demand is reached.

Currently trading algorithm consider that each demand is served by one provider, it will be useful if more provides together can serve the demand. For example, if a demand of 10 resources arrives and no one provider can serve the demand, however, 3 providers together can do so than demand should be matched, instead of adding to the demand queue. Even, once the match has been made, the application is executed on the selected resoruces even if any cheaper resoruces become available during the course. On the fly switching of application to cheaper suitable resource is also under consideration.

REFERENCES

[1] J. Altmann, C. Courcoubetis, J. Darlington, and J. Cohen, *Gridecon - the economic-enhanced next-generation internet*, in GECON'07: Proceedings of the 4th international conference on Grid economics and business models, Berlin, Heidelberg, 2007, Springer-Verlag, pp. 188–193.

[2] J. Altmann and S. Routzounis, *Economic modeling of grid services*, 2006.

[3] R. Buyya and S. Venugopal, *MARKET-ORIENTED COMPUTING AND GLOBAL GRIDS: An Introduction*, in Market-oriented Grid and Utility Computing, John Wiley and Sons, Hoboken, NJ, USA, 2010, ch. 1, pp. 3–27.

[4] Cloud4SOA, *Cloud4soa*. World Wide Web electronic publication. *http : //www.cloud4soa.eu/* accessed 04-Jan-2013.

[5] T. Cortes, C. Franke, Y. Jégou, T. Kielmann, D. Laforenza, B. Matthews, C. Morin, L. P. Prieto, and A. Reinefeld, *XtreemOS: a Vision for a Grid Operating System*. XtreemOS Technical Report # 4, May 2008.

[6] T. Fahringer, C. Anthes, A. Arragon, A. Lipaj, J. Müller-Iden, C. Rawlings, R. Prodan, and M. Surridge, *The edutain@grid project*, in 4th International Workshop on Grid Economics and Business Models, D. J. Veit and J. Altmann, eds., vol. 4685 of LNCS, Springer, August 2007, pp. 182–187.

[7] H. M. Frutos and I. Kotsiopoulos, *Brein: Business objective driven reliable and intelligent grids for real business*, International Journal of Interoperability in Business Information Systems, Issue3 (1) (2009).

[8] Google, *Run your web apps on google's infrastructure*. World Wide Web electronic publication. *http : //code.google.com/appengine* accessed 07-Oct-2010.

[9] IBM, *e-business on demand: A developer's roadmap*. World Wide Web electronic publication. *http : //www.ibm.com/developerworks/ibm/library/ i − ebodov/index.html* accessed 07-Oct-2010.

[10] H. Labs, *Utility computing services*. World Wide Web electronic publication. *http : //www.hpl.hp.com/research/about/ utility_services.html* accessed 07-Oct-2010.

[11] S. Mani and S. Rao, *Operating cost aware scheduling model for distributed servers based on global power pricing policies*, in Proceedings of the Fourth Annual ACM Bangalore Conference, COMPUTE '11, New York, NY, USA, 2011, ACM, pp. 12:1–12:8.
[12] C. Morin, *XtreemOS: A Grid Operating System Making your Computer Ready for Participating in Virtual Organizations*, in Proceedings of the Tenth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2007), IEEE Computer Society, 2007, pp. 393–402.
[13] mOSAIC, *mosaic cloud*. World Wide Web electronic publication. *http* : *//www.mosaic − cloud.eu/* accessed 04-Jan-2013.
[14] D. Neumann, J. Stoesser, A. Anandasivam, and N. Borissov, *Sorma - building an open grid market for grid resource allocation*, in GECON'07: Proceedings of the 4th international conference on Grid economics and business models, Berlin, Heidelberg, 2007, Springer-Verlag, pp. 194–200.
[15] J. Nimis, A. Anandasivam, N. Borissov, G. Smith, D. Neumann, N. Wirström, E. Rosenberg, and M. Villa, *Sorma — business cases for an open grid market: Concept and implementation*, in GECON '08: Proceedings of the 5th international workshop on Grid Economics and Business Models, Berlin, Heidelberg, 2008, Springer-Verlag, pp. 173–184.
[16] Optimis, *Optmised infrastructure services*. World Wide Web electronic publication. *http* : *//www.optimis − project.eu/* accessed 04-Jan-2013.
[17] S. Sehgal, M. Erdélyi, A. Merzky, and S. Jha, *Understanding application-level interoperability: Scaling-out mapreduce over high-performance grids and clouds*, Future Generation Comp. Syst., 27 (2011), pp. 590–599.
[18] A. W. Services, *Amazon elastic compute cloud (amazon ec2)*. World Wide Web electronic publication. *http* : *//aws.amazon.com/ec*2 accessed 07-Oct-2010.
[19] K. Stanoevska-Slabeva, D. M. Parrilli, and G. Thanos, *Beingrid: Development of business models for the grid industry*, in GECON '08: Proceedings of the 5th international workshop on Grid Economics and Business Models, Berlin, Heidelberg, 2008, Springer-Verlag, pp. 140–151.
[20] Sun, *Sun grid engine*. World Wide Web electronic publication. *http* : *//wikis.sun.com/display/GridEngine/Home* accessed 07-Oct-2010.
[21] M. Surridge, S. Taylor, and D. D. Roure, *Experiences with gria - industrial applications on a web services grid*, in In E-SCIENCE 05: Proceedings of the First International Conference on e-Science and Grid Computing, IEEE Computer Society, 2005, pp. 98–105.
[22] K. Wasielewska, M. Ganzha, M. Paprzycki, M. Drozdowicz, D. Petcu, C. Badica, N. Attaoui, I. Lirkov, and R. Olejnik, *Negotiations in an Agent-based Grid Resource Brokering System*, Saxe - Coburg Publications, 2012, ch. 16, pp. 355 – 374.
[23] XtreemOS Consortium, *Methodology and design alternatives for federation and interoperability*, in XtreemOS public deliverables - D3.5.15, A. Arenas, ed., Work Package 3.5, March 2010.
[24] J. Zhang and J. Luo, *Agent based automated negotiation for grid*, in Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on, april 2008, pp. 330 –336.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 27, 2012
Accepted: Jan. 04, 2013

AIMS AND SCOPE

The area of scalable computing has matured and reached a point where new issues and trends require a professional forum. SCPE will provide this avenue by publishing original refereed papers that address the present as well as the future of parallel and distributed computing. The journal will focus on algorithm development, implementation and execution on real-world parallel architectures, and application of parallel and distributed computing to the solution of real-life problems. Of particular interest are:

Expressiveness:

- high level languages,
- object oriented techniques,
- compiler technology for parallel computing,
- implementation techniques and their efficiency.

System engineering:

- programming environments,
- debugging tools,
- software libraries.

Performance:

- performance measurement: metrics, evaluation, visualization,
- performance improvement: resource allocation and scheduling, I/O, network throughput.

Applications:

- database,
- control systems,
- embedded systems,
- fault tolerance,
- industrial and business,
- real-time,
- scientific computing,
- visualization.

Future:

- limitations of current approaches,
- engineering trends and their consequences,
- novel parallel architectures.

Taking into account the extremely rapid pace of changes in the field SCPE is committed to fast turnaround of papers and a short publication time of accepted papers.

INSTRUCTIONS FOR CONTRIBUTORS

Proposals of Special Issues should be submitted to the editor-in-chief.

The language of the journal is English. SCPE publishes three categories of papers: overview papers, research papers and short communications. Electronic submissions are preferred. Overview papers and short communications should be submitted to the editor-in-chief. Research papers should be submitted to the editor whose research interests match the subject of the paper most closely. The list of editors' research interests can be found at the journal WWW site (`http://www.scpe.org`). Each paper appropriate to the journal will be refereed by a minimum of two referees.

There is no a priori limit on the length of overview papers. Research papers should be limited to approximately 20 pages, while short communications should not exceed 5 pages. A 50–100 word abstract should be included.

Upon acceptance the authors will be asked to transfer copyright of the article to the publisher. The authors will be required to prepare the text in LaTeX 2$_\varepsilon$ using the journal document class file (based on the SIAM's `siamltex.clo` document class, available at the journal WWW site). Figures must be prepared in encapsulated PostScript and appropriately incorporated into the text. The bibliography should be formatted using the SIAM convention. Detailed instructions for the Authors are available on the SCPE WWW site at `http://www.scpe.org`.

Contributions are accepted for review on the understanding that the same work has not been published and that it is not being considered for publication elsewhere. Technical reports can be submitted. Substantially revised versions of papers published in not easily accessible conference proceedings can also be submitted. The editor-in-chief should be notified at the time of submission and the author is responsible for obtaining the necessary copyright releases for all copyrighted material.