RIGA TECHNICAL UNIVERSITY

Antons MISLEVICS

MOBILE AGENTS FOR BUSINESS PROCESS MANAGEMENT SUPPORT IN CLOUD COMPUTING ENVIRONMENTS

Summary of Doctoral Thesis

Riga 2013

RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science and Information Technology
Institute of Applied Computer Systems

Antons MISLEVICS
Student of the Doctoral study program “Computer Systems”

MOBILE AGENTS FOR BUSINESS PROCESS MANAGEMENT SUPPORT IN CLOUD COMPUTING ENVIRONMENTS

Summary of Doctoral Thesis

Scientific supervisor
Dr.habil.sc.ing., Professor
J. GRUNDSPENKIS

Riga 2013

UDK 004.415(043.2)
Mi 826 u

Mislevics A. Using Mobile Agents to Support Business Process Management in Cloud Computing Environments. Summary of Doctoral Thesis.-R.:RTU, 2013.-50 p.

Printed in accordance with the decision of the board of Institute of Applied Computer Systems, Faculty of Computer Science and Information Technology, Riga Technical University No 12-32 of December 10, 2012.

This work has been supported by the European Social Fund within the project “Support for the implementation of doctoral studies at Riga Technical University”.

EIROPAS SOCIĀLAIS FONDS

EIROPAS SAVIENĪBA

DOCTORAL THESIS
SUBMITTED FOR THE DOCTORAL DEGREE IN ENGINEERING

The defence of the thesis submitted for doctoral degree of computer systems will take place at an open session on March 4, 2013 in Meza Street 1/3, auditorium 202, Riga Technical University Faculty of Computer Science and Information Technology.

OFFICIAL OPPONENTS

Professor, Dr.sc.ing. Uldis Sukovskis
Riga Technical University, Latvia

Professor, Dr.sc.comp. Janis Bicevskis
University of Latvia, Latvia

Professor, PhD Kuldar Taveter
Tallinn University of Technology, Estonia

CONFIRMATION

I confirm that I have developed this thesis submitted for the doctoral degree at Riga Technical University. This thesis has not been submitted for the doctoral degree in any other university.

Antons Mislevics …………………………….(signature)

Date: ………………………

The doctoral thesis is written in Latvian and includes introduction, 4 sections, conclusions, bibliography, 4 appendixes, 62 figures and 4 tables in the main text, 219 pages. The bibliography contains 219 references.

TABLE OF CONTENTS

DEFINITIONS

Business process – a set of activities that help to achieve specific goal of the organization;

Business process model – formal description of business process in graphical way;

Business process management – concepts, methods and techniques that support analysis, design, implementation and management of business processes in the organization;

Business process management system (BPMS) – information system that supports business process management in the organization;

Cloud-BPMS – business process management system in cloud computing environment. This approach allows organizations to use BPMS technology as a service, thus reducing initial costs for implementing BPMS;

Enterprise information system – information system that is designed to work with large amounts of corporate data, or to support and manage business processes of the company;

Process aware information system – information system that supports business processes in organizations, taking into account specifics of business processes and context of the company;

Agent – anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors;

Software agent – computer system that is situated in some environment and is capable of autonomous action in order to meet its design objectives;

Mobile agent (MA) – software agent that can move within the network while executing;

Mobile agent system – platform that provides capabilities to create, execute, transfer and terminate MAs;

Strong mobility – agent mobility implementation approach, when MA system captures and moves execution state with the agent. In MA systems that support strong mobility it is guaranteed that all variables will have identical values after moving MA to another host;

Weak mobility – agent mobility implementation approach, when MA system captures most of a program’s data, but restarts the program from a predefined program point and thus requires some programmer involvement at each migration.

INTRODUCTION

Business Process Management (BPM) is becoming more popular in recent years. As a result BPM is implemented not only in pilot projects, but also introduced on enterprise level [13]. BPM is a set of concepts, methods and techniques that support analysis, implementation and management of business processes in organization. Historically business processes more often are implemented using traditional approach: employees are familiarized with business processes that they are implementing and which are described in internal policies and procedures [113]. However, recently BPM related technologies are gaining popularity [13, 95]. Companies more often start using specialized information systems that support BPM implementation [21]. Such systems are called Business Process Management Systems (BPMS).

Analyzing future of BPM researchers agree that cloud computing environments will be used to adopt BPM applications [13, 54, 62, 88, 95]. Cloud environments allow organizations to reduce initial implementation costs and use BPMS as a service (*Software as a Service* – SaaS). As any successful BPMS implementation requires establishing a solid integration of business processes with existing line of business (LOB) applications and organizational data sources, moving BPMS to the cloud puts the focus on “cloud to on-premises” integration questions. Similar questions are addressed in many studies around integrating SaaS applications with on-premises systems and data sources [31, 55, 56, 75, 87, 102, 103, 114, 118]. Nevertheless, integrating cloud-BPMS with on-premises LOB applications has some specific scenarios and requirements that cannot be effectively addressed by using existing technologies and patterns [62]. The thesis describes how cloud-BPMS integration challenges are solved using mobile agents (MA) technology.

Motivation of the research

Analyzing BPMS evolution researchers foresee that cloud computing environments will be used to implement BPMS [54, 88]. Market analysts observe the same trend [13, 95].

In 2011 Gartner market analysts have performed market research to analyze modern BPM technologies, related knowledge areas and IT services [13]. The goal of the research was to evaluate how soon specific modern technology, knowledge area or IT service will become widely adopted in enterprises, and how significant the effect of the implementation will be.

Analysts pay special attention to cloud BPM offerings that allow organizations to use technology as a service, thus reducing initial implementation costs [13]. Analysts are expecting that cloud-BPMS will achieve mainstream adoption within two to five years. This gets also reflected in numbers for some BPMS vendors which are observing that more revenue is coming from cloud-BPMS offerings in recent quarters [2].

While implementing BPMS it is important to ensure integration with existing enterprise information systems. In case of cloud-BPMS SaaS application must be integrated with existing on-premises applications. As none of existing technical solutions for integrating SaaS and on-premises applications fully addresses specific cloud-BPMS integration requirements new integration mechanism should be proposed.

The goal of the thesis

The goal of the thesis is to develop mobile agent based mechanism that allows integrating cloud-BPMS with existing enterprise information systems and devices, implement this mechanism in software system and perform evaluation of developed system.

The goal is based on the following **hypotheses**:

- it is possible to ensure integration between cloud-BPMS and existing enterprise information systems and devices by using mobile agents;
- it is possible to create processes and tools that would allow people having no experience in programming to develop mobile agents;
- it is possible to define universal format of mobile agent packages that addresses common requirements of mobile agent systems.

The tasks of the thesis

In order to achieve the goal of the thesis the following tasks are specified:

- analysis of development tendencies and actual research directions in business process management area;
- analysis of problems that are solved by using business process management systems;
- analysis of implementation aspects of business process management systems;
- analysis of specific integration requirements of cloud-BPMS and how these are addressed by existing integration solutions;
- analysis of existing mobile agent systems and evaluation of suitability for this integration scenario;
- development of architecture of mobile agent system that enables integration of cloud-BPMS with on-premises systems and devices;
- development of prototype that implements all components of proposed architecture;
- evaluation of developed prototype in approbation scenarios.

Research objects

Research objects of the thesis are cloud-BPMS and mobile agent systems.

Research subject

Research subject of the thesis is integration scenarios in cloud-BPMS and how these can be implemented by using mobile agents.

Research methods

The thesis is based on incompleteness and drawbacks of known systems acquired as a result of the analysis of available information sources. Methods of software engineering were used to implement prototype of the system. The prototype was approbated in six test scenarios.

Scientific novelty of the thesis:

- the approach for integrating cloud-BPMS with on-premises systems and devices has been developed, that is based on the architecture of mobile agent system proposed in the thesis;
- the algorithm that allows to determine possible approach to integrate mobile agents with business process management system has been developed;
- mobile agent system that is based on using workflows to design and execute agents has been developed;
- common MA package format has been developed.

Practical value of the thesis

Practical value of the thesis is related to developed prototype of mobile agent system that can be used:

- to integrate cloud-BPMS with on-premises systems and devices to address requirements of specific organization;
- to validate if proposed approach and architecture of mobile agent system can be used in other scenarios;
- as sample implementation of proposed approach and architecture of mobile agent system, that will help to implement similar solutions for other platforms or by using different technologies.

Approbation of the obtained results

Four presentations on the main results of the research were made in international scientific conferences (two of them in foreign countries, i.e. in Lithuania, and two in Latvia):

- Mislevics, A. An Introduction to Mobile Agent-Supported Business Process Management. 4th International Conference: Information Society and Modern Business, The Role of Regional Centers in Business Development, May 14-16, 2009, Ventspils, Latvia;
- Mislevics, A. Mobile Agents in Business Process Management. 50th RTU International Scientific Conference, October 14-15, 2009, Riga, Latvia;
- Mislevics, A., Grundspenkis, J. Mobile Agents for Integrating Cloud-Based Business Processes with On-Premises Systems and Devices. Baltic DB & IS 2012, Tenth

International Baltic Conference on Databases and Information Systems, July 8-11, 2012, Vilnius, Lithuania;

- Mislevics, A., Grundspenkis, J. Workflow Based Approach for Designing and Executing Mobile Agents. The Second International Conference on Digital Information Processing and Communications (ICDIPC12), July 10-12, 2012, Klaipeda, Lithuania.

The main results of the thesis are reflected in five publications:

- Grundspeņķis, J., Mislēvičs, A., Intelligent Agents for Business Process Management Systems. Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology. Malgorzata Pankowska (ed.), (Karol Adamiecki University of Economics in Katowice, Poland), pp. 97-131, 2010. Citation indexed: IGI Global, InfoSci, Google, Amazon, ACM Digital Library, Safari Books Online;
- Mislēvičs, A., An Introduction to Mobile Agent-Supported Business Process Management. Proceedings of The 4th International Conference "Information Society and Modern Business", Ventspils, Latvia, May 14-16, 2009, pp. 49-57, 2009;
- Mislēvičs, A., Grundspeņķis, J., Mobile Agents for Integrating Cloud-Based Business Processes with On-Premises Systems and Devices, Frontiers in Artificial Intelligence and Applications, vol.249: Databases and Information Systems VII, IOS Press, ISBN 978-1-61499-161-8, pp. 191-203, 2013. Citation indexed: ACM Digital Library, Web of Science (Conference Proceedings Citation Index and Book Citation Index – Science), DBLP, Google Scholar, SciVerse Scopus, Zentralblatt MATH;
- Mislēvičs, A., Grundspeņķis, J., Workflow Based Approach for Designing and Executing Mobile Agents, Proceedings of The Second International Conference on Digital Information Processing and Communications (ICDIPC12), ISBN 978-1-4673-1105-2, pp. 97-102, 2012. Citation indexed: IEEE Xplore, Elsevier, Thomson Reuters, ProQuest, IET, NLM, Google;
- Mislēvičs, A., Grundspeņķis, J., Integrating Workflow-Based Mobile Agents with Cloud Business Process Management Systems, International Journal on New Computer Architectures and Their Applications (IJNCAA) vol. 2(4), The Society of Digital Information and Wireless Communications, ISSN: 2220-9085, pp. 511-530, 2012. Citation indexed: Ulrichsweb, INSPEC, EBSCOhost, Academickeys, MediaFinder, Google Scholar, Microsoft Academic Research, ResearchGate, Research Bible, Scirus, Copernicus Journals Master List, Cabell Directory.

The prototype developed in the thesis is approbated in six scenarios:

- Designing and executing MAs;
- Protecting sensitive MA configuration information by using encryption;
- Integrating MA system with cloud-BPMS using *SharePoint Online* system as an example;
- Signing MA packages using Latvian identification card (eID);
- Integrating cloud-BPMS with on-premises application: using data from *Tildes Jumis* in *SharePoint Online* workflow;
- Integrating cloud-BPMS with devices in local network: using local printer in *SharePoint Online* workflow.

Structure of the thesis

The thesis includes introduction, 4 chapters, conclusions, bibliography and 4 appendixes.

In the introduction the motivation of the thesis, research goals and tasks are defined. Applied scientific methods, novelty, practical value of the thesis and approbation of the main results are described as well.

Analysis of business process management area is performed in Chapter 1 in order to determine key concepts and elements of BPM, development tendencies and actual research directions.

Chapter 2 is devoted to analysis of enterprise information systems and business process management systems, and covers scenarios of how intelligent agents are used in business process management systems.

Analysis of intelligent agents and mobile agents is performed in Chapter 3. It allows to identify opened issues in existing implementation of mobile agent systems that make these systems inappropriate for cloud-BPMS integration scenarios. To address these issues a newly proposed workflow based approach for designing and executing mobile agents and common MA package format are introduced.

Proposed MA system architecture, *AgentWF* prototype implementation details, approbation scenarios and results are covered in Chapter 4.

Main results of research and conclusions are presented in the last part of the thesis.

1. BUSINESS PROCESS MANAGEMENT

Business process management (BPM) area has evolved gradually in 20th century. Nowadays term BPM has different meanings for different audiences. It is treated as a direction of scientific researches, business tool, market category, area of consultancy, methodology, or management discipline.

The goal of this chapter is to define key concepts and elements of BPM, and analyze development tendencies and actual research directions.

1.1. Key Concepts

Business process management (BPM) is based on assumption that any product (or service) offered by a company is a result of executing a set of activities [113]. Business processes allow to organize such sets of activities and analyze dependencies between these. Every business process helps to achieve specific goal of the company.

BPM includes concepts, methods and techniques that support analysis, design, implementation and management of business processes in organization. The main goal of BPM is to create formal description of business process that includes all activities, dependencies between these and existing limitations. Such formal descriptions allow to analyze and optimize existing business processes, and implement new ones.

To describe business process in a formal way, graphical business process model is created. Models of business processes are designed using special graphical notations. Notations define graphical elements that may be used to construct process model, their properties and principles how these elements can be combined. Several notations are used to design models of business processes, however their base elements and principles are similar [113]. Business Process Modeling Notation (BPMN) is one of the most commonly used notations and a standard language to describe business process models [74]. To implement designed process models and to perform analysis of business processes to identify potential improvements, complete BPM process must be implemented in the organization.

BPM is iterative process that consists of four stages: analysis and design, implementation, execution and diagnostics (Figure 1.1) [96, 113]. The process is started by describing business process in formal way. This is crucial stage that could be complicated for organizations where processes are executed chaotically and every new instance of the process is treated as entirely new case. This approach usually does not allow achieving predictable results when executing business processes [34]. For every process appropriate performance indicators and their expected values must be defined. These are defined by analyzing specifics of the process, expectations of customers or offerings of competitors: for example, execution time of the process or specific stages. When the process is implemented it should be managed and values of appropriate performance indicators must be collected while executing process instances. If collected values are worse than expected values defined during analysis phase the process must be analyzed to understand what causes this issue: inaccurate process model or inappropriate execution quality. When cause of the issue is identified appropriate resolution plan is developed and implemented and BPM iteration is repeated.

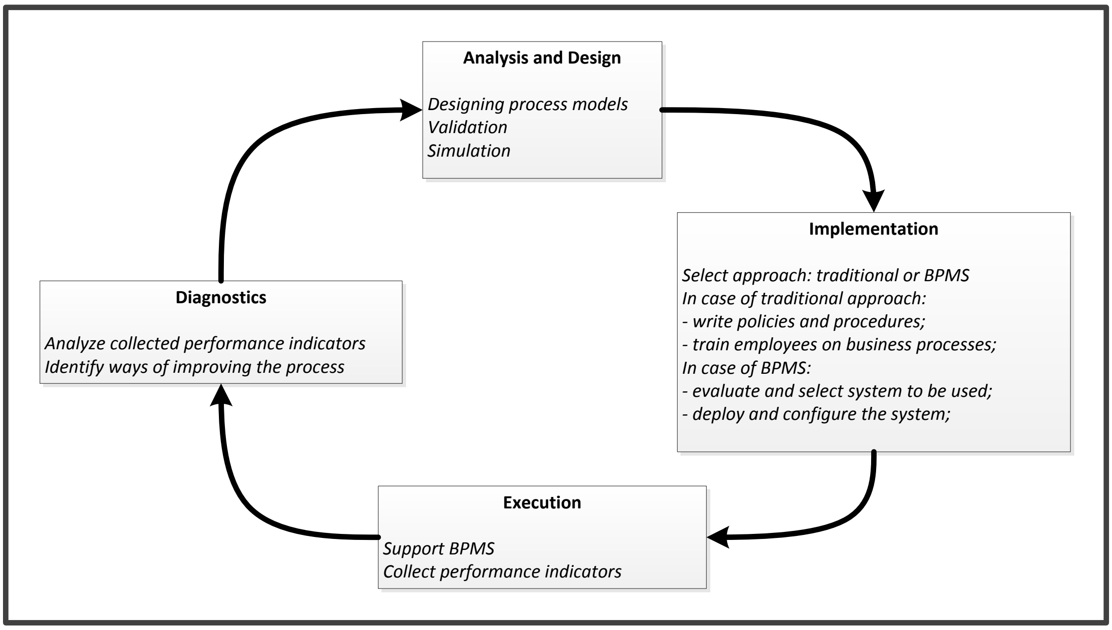

Figure 1.1. Business process management cycle

Analysis of business processes is performed gradually at five levels: from high level business strategies to implementation details of specific business processes (Figure 1.2) [113]. The highest level defines enterprise strategy, describing concepts that ensure long-term development of the company. Next, organization strategy is split into specific goals. Each goal can be further split into several sub-goals if needed. Third level defines organizational business processes – high-level business processes that are documented in plain text, specifying inputs, outputs, expected results and dependencies with other organizational business processes. Next, for each organizational business process operational business processes are defined that provide formal description of organizational business process or separate parts of it. At this stage formal process models are created. Operational business processes contain information about activities and dependencies between these, however do not describe implementation details. Implementation details are described at last level when implemented business processes are created. Implemented business process is a specification that describes how specific operational business process is implemented (by using organizational or technical means).

Figure 1.2. Analysis of business processes

Organizations usually implement business processes by using traditional approach when management educates employees on business processes that are documented in internal policies and procedures [113]. However, recently more companies start using special software systems that support BPM implementation: analysis, design, implementation, execution and diagnostics of business processes [21]. These systems are named Business Process Management Systems (BPMS). BPMS ensure execution of business processes according to defined process models.

1.2. Evolution of Business Process Management

Modern BPM concepts are the result of evolution of three major research directions in business processes field [36]. These are: quality management, business management and information technology approaches. Historically all three approaches were initiated by industrial revolution in the end of 19th century. Quality management approach is focusing on ideas that must be followed by good managers in order to improve efficiency of work and ensure quality of products: simplification of tasks, controlling and analyzing work time, ongoing experimenting to find more efficient ways of performing specific task and implementing control systems that allow to measure results of processes and reward best-performing employees. As opposed to quality management, business management approach was driven not by engineers and quality control experts, but by business managers who were searching for ways of how to make work in organizations more efficient. The main focus is on enterprise strategy and ways to organize and manage employees in order to achieve goals of the company. Information technology approach is based on using computers and specialized software applications to automate and improve business processes.

Modern BPM trend combines ideas of all three approaches in single BPM approach [36]. Research directions are organized in three levels: organizational, process and implementation. **Organizational level** has initiatives that are related to strategy, architecture, process management and performance indicators in context of entire organization. Significant research directions are Enterprise Architecture [35, 124], Value Chains [5, 8, 32, 34, 79], Business Process Frameworks or Operation Reference Frameworks [4, 36, 80], Process Maturity Models [10, 73, 78, 84, 85], Integrated Process Measurement Systems [4, 29, 80, 97] and managing transformations in organizations [36]. **Process level** groups methodologies and tools used to analyze and improve business processes. Important research directions are Innovations Management [33, 68, 83, 93] and analysis and design of complex business processes [36, 37]. **Implementation level** contains methodologies and technologies used to implement and execute business processes. Significant directions are business process management systems (BPMS) and standardization and certification in BPM field [36, 52, 113].

1.3. Analysis of Market Situation in Business Process Management Field

In 2011 Gartner market analysts have performed market research to analyze modern BPM technologies, related knowledge areas and IT services [13]. The goal of the research was to evaluate how soon specific modern technology, knowledge area or IT service will become widely adopted in enterprises, and how significant the effect of the implementation will be. Market analysts observe [13] that during the last years BPM is becoming more popular. As a result organizations are implementing BPM not only in pilot projects, but also on enterprise level. However, BPM implementation requires not only specific knowledge and skills, but also changes in organizational culture and work processes [71]. To ensure knowledge and skillset required to implement BPM organizations are building internal Business Process Competency Centers (BPCC) [13]. BPCC is a department that supports BPM implementation in organization and consults other employees, it is involved in all BPM projects, defines internal BPM guidelines and makes decisions on BPM standards and tools that are used in the organization. It is noted that BPCC is critically important to cultivate BPM knowledge and skills in the organization [70, 72].

Recently BPM technologies are also becoming more popular [13, 95]. Analysts [13] foresee that the following technologies will become widely adopted in enterprises in upcoming two years: Business Activity Monitoring (BAM) [22, 23], Business Rule Management Systems (BRMS) [11], Case Management Systems [3], Optimization and Simulation Solutions [94] and Business Process Management Systems (BPMS) [95].

BPMS is BPM technology that provides broad set of features for analysis, design, implementation, execution, monitoring and improvement of business processes. Modern BPMS products support all phases of BPM iteration.

Analysts pay special attention to cloud BPM technologies that allow organizations to use technology as a service, reducing costs associated with initial implementation [13].

1.4. Conclusions

Analysis of business process management area has led to the following conclusions:

- BPM includes concepts, methods and techniques that support analysis, design, implementation and management of business processes in organization;
- Modern BPM research directions are organized in three levels: organizational, process and implementation;
- BPM is iterative process that consists of four stages: analysis and design, implementation, execution and diagnostics;
- Market analysis shows that during the last years BPM is becoming more popular. More companies start using special software systems that support BPM implementation – BPMS. Analysts pay special attention to cloud BPM technologies that allow organizations to use technology as a service, reducing costs associated with initial implementation.

2. BUSINESS PROCESS MANAGEMENT SYSTEMS

In the first chapter by analyzing business process management (BPM) area key concepts and elements of BPM are defined and development tendencies and actual research directions are analyzed. It is concluded that more companies start using business process management systems (BPMS). BPMSs are modern enterprise information systems that are oriented on business processes and support BPM implementation in organizations.

This chapter covers actual trends in enterprise information systems, architectures of process aware information systems and possible usages of intelligent agents in business process management systems.

2.1. Enterprise Information Systems: Evolution and Recent Trends

Enterprise information systems are systems that are designed to work with large amounts of corporate data, or to support and manage business processes of the company [15]. Enterprise information systems for managing and automating business processes were introduced with growing popularity of computers, and were evolving over last 40 years [96]. Analyzing development tendencies of enterprise information systems researchers note several significant trends [109, 111].

Nowadays there are many information systems available that match specific functional needs of organizations: for example, operating systems support various devices, database management systems allow to work with different amounts and types of information, various information systems address specific needs of separate departments (vehicles management, issue tracking, human resources management, accounting) [109]. As a result the main goal of developers when implementing enterprise information systems is not to develop specific functionality, but to integrate functionality already provided by available systems.

Another significant trend is that implementation and evolution of enterprise information systems had become more dynamic [111]. In the past development of any system had started with detailed analysis of requirements and planning of all functions that were provided by the system. Nowadays developers must foresee that the system will naturally evolve over time to adapt to changes in organizational environment and business processes. As a result it is necessary to build integrations with enterprise information systems that already exist in the organization [96].

The next trend is that nowadays designing enterprise information systems developers are focusing on business processes instead of data [96]. Earlier in 1970th and 1980th the main goal of information systems was to ensure storing and managing data. Developers were not analyzing business processes in which data was used. As a result business processes were usually later adapted to specifics of information systems. This approach had changed significantly in 1990th when Business Process Reengineering (BPR) methodology was introduced. Nowadays process oriented approach prevails [96, 111]. The idea of process oriented approach is that the goal of developers when implementing information system is to ensure execution of entire business process by building integrations with existing specific systems that are used to work with necessary data [7, 96]. As a result new category of information systems was introduced named “process aware” information systems [86, 110]. These systems support business processes in organizations, taking into account specifics of business processes and context of the company.

Researchers are also noting that cloud computing environments are becoming more popular for implementing enterprise information systems. Cloud environments allow organizations to use enterprise information systems as services [13, 54, 62, 88, 95].

2.2. Process Aware Information Systems

The core of process aware information system is a workflow. Workflow is automation of a business process [123]. First systems of this kind were introduced in 1990th and were named Workflow Management Systems (WfMS) [24, 40, 51, 53, 96, 108, 113, 125]. Later software vendors started using a name Business Process Management Systems (BPMS) [96, 113]. Comparing to WfMS, BPMS supports the entire BPM iteration. WfMS provides functionality for analysis and design, implementation and execution of business processes. In addition to this, BPMS has diagnostics capabilities, for example, tools for simulation, or collection and analysis of performance indicators [111].

Analyzing user requirements analysts conclude that modern BPMS must support the following key elements of BPM in organizations [95]:

- Optimizing the performance of end-to-end business processes, as well as processes that might extend beyond the enterprise to include partners, suppliers and customers;
- Making the business process visible to business and IT representatives through business process modeling, monitoring and optimization;
- Keeping the business process model in sync with process instances during execution;
- Empowering business users and analysts to manipulate a business process model to modify instances of the process;
- Enabling the rapid integration of processes and underlying systems for continuous process improvement and optimization.

It is obvious that to support these key elements establishing solid integration with existing organization's information systems and data sources is essential [95, 96]. Figure 2.1 shows components of modern BPMS.

The most common architecture for implementing BPMS is Workflow-Oriented BPMS (WBPMS) [28, 113]. BPMS of this kind are also called centralized, because these systems have a single process which ensures execution of the entire business process. WBPMS are straightforward to implement and that's why popular among developers. However, several studies have identified a set of issues with which WBPMS cannot deal so well: for example, insufficient automation and flexibility, lack of resource management and non-trivial error handling mechanism [27, 28, 57, 67, 77, 107, 113, 123]. As a result various alternative BPMS implementation approaches are proposed, like Subject-Oriented BPM, Social BPM, Dynamic BPM or by using intelligent agents [6, 12, 13, 16, 41, 42, 43, 44, 58, 89, 101, 123].

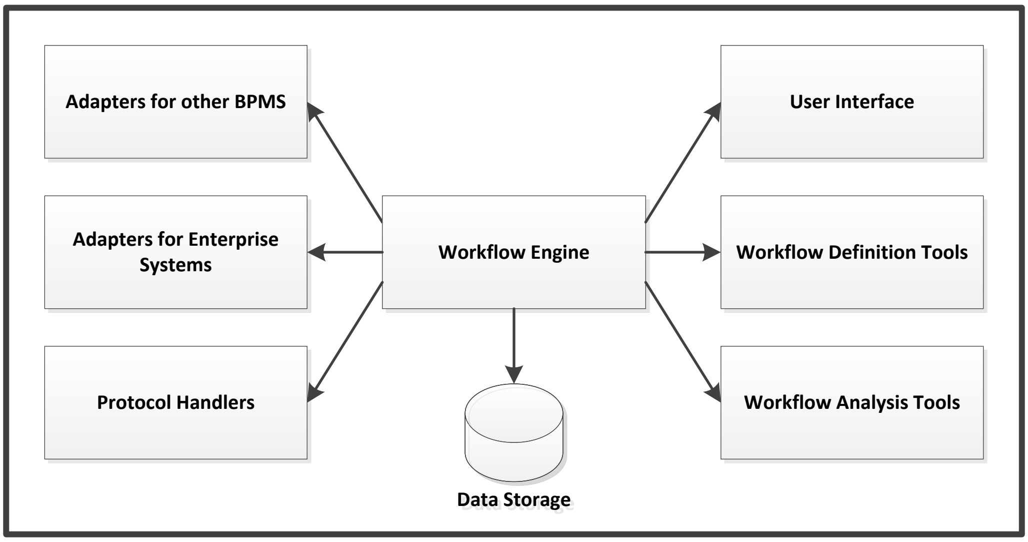

Figure 2.1. Components of business process management system (adopted from [96])

Analyzing future of BPMS researchers agree that cloud computing environments will be used to adopt BPMS applications [54, 88]. Market analysts observe the same trend and pay special attention to cloud BPM offerings, expecting that these will achieve mainstream adoption within two to five years [13, 95]. Cloud environments allow organizations to reduce initial implementation costs and use BPMS as a service (*Software as a Service* – SaaS). As any successful BPMS implementation requires establishing a solid integration of business processes with existing line of business (LOB) applications and organizational data sources, moving BPMS to the cloud puts the focus on "cloud to on-premises" integration questions. From security perspective the main integration challenge in accessing on-premises applications is that these are typically isolated from external network by using a firewall. That's why direct calls from SaaS application to on-premises applications are not possible (Figure 2.2). To avoid this issue SaaS and on-premises applications can be integrated the way that all requests are sent from local network only (by on-premises applications). The main disadvantage of this approach is that it is not scalable and may significantly increase the load on network infrastructure, especially if information in on-premises applications is modified frequently [55]. In addition, this approach cannot be used in integration scenarios where communication is initiated by an event in SaaS application.

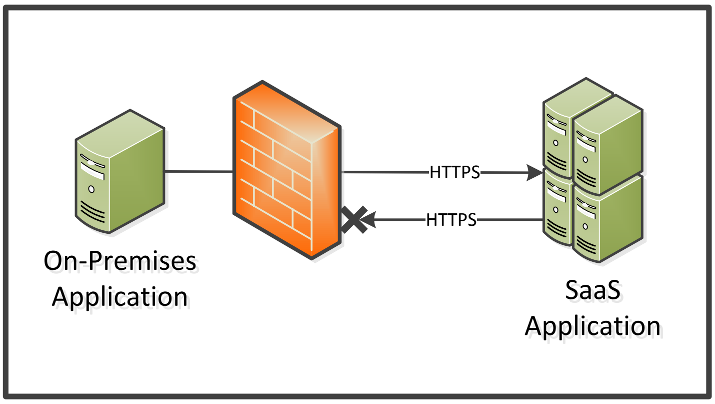

Figure 2.2. Communication between SaaS and on-premises applications (adopted from [55])

There are several known technical solutions that allow SaaS applications to send requests to on-premises applications:

1) Publishing services provided by on-premises applications as web services to external network (by using web services provided by the system, or developing custom ones). In case of complex network infrastructures publishing web services could require significant effort. In addition, this approach requires constant involvement of IT department to publish or unpublish web services. It also has security risks as in case of incorrect configuration of web services external parties can get access to information stored in on-premises applications (for example, if addresses from which connecting to

these services is allowed are not limited, or authentication mechanism is not configured) [55];

2) Configuring Virtual Private Network (VPN) channel between SaaS application and local network. As a result SaaS application is connecting to on-premises applications directly and requests are not sent through the firewall [87, 103, 118]. This approach is implementable only if Infrastructure as a Service (IaaS) cloud option is used, as in this case organization controls servers in cloud computing environment at infrastructure level. In case of SaaS applications, when organization is using information system as a service, VPN cannot be configured;
3) Using communication proxy that is located in external network and is available to SaaS and on-premises applications (relayed messaging) [56, 114]. This approach allows SaaS and on-premises applications to communicate the same way, as when both systems are deployed in the same network without the need to reconfigure the firewall. However, to build integration corresponding web services must be configured in special way that all requests are sent through communication proxy;
4) Deploying special communication agent in local network, that handles requests from SaaS application [31, 55, 75, 102]. The agent opens communication channel to SaaS application and waits for incoming requests. After receiving a request the agent connects to on-premises applications by using web services or database interfaces. This approach does not require any configuration to on-premises applications or the firewall. It also allows to reduce security risks as all communication between SaaS and on-premises applications is controlled by communication agent.

Analyzing specifics of BPMS integration scenarios the following additional SaaS integration requirements were identified during the research [62]:

1) Performing complex computations close to data sources in internal network (dealing with large amounts of data);
2) Performing complex transformations and computations with data stored in on-premises application in internal network (for security and privacy reasons);
3) Implementing rapid changes in integrations (to adapt business process to changes in the environment);
4) Accessing legacy systems and specific devices that are deployed on-premises and have no web services or database interfaces.

Table 2.1 shows how these specific integration requirements are addressed by existing technical solutions. VPN option is not included as it cannot be used in SaaS integration scenarios, when organization is using information system as a service and cannot control servers in cloud computing environment at infrastructure level.

Table 2.1. BPMS specific integration requirements in existing technical solutions

Requirement	Technical solution			
	Publishing existing web services	**Implementing specific web services**	**Communication proxy**	**Communication agent**
1) Complex computations close to data sources	Not possible	Partly. Computations and transformations in local network are possible only on servers where implemented web service is deployed	Not possible	Partly. Computations and transformations in local network are possible only on servers, where communication agent is deployed
2) Transformations and computations in local network	Not possible	Possible	Not possible	Possible
3) Rapid changes in integrations	Not possible, as publishing or unblishing web services requires IT involvement	Not possible, as publishing or unblishing web services requires IT involvement	Not possible, as web services must be reconfigured	Partly. Depends on implementation. Usually part of integration configuration information is stored in local network and is tightly coupled with web services and database interfaces of on-premises applications
4) Accessing specific systems and devices	Not possible	Possible	Not possible	Usually is not possible. Depends on implementation of communication agent. Systems and devices must be available for direct requests from communication agent, what is typically not possible in distributed and dynamic environments

As seen from the table above, none of existing solutions fully addresses specific cloud-BPMS integration requirements. In the thesis these integration problem is solved by using mobile agents [62]. Implementation details and approbation results of proposed solution are covered in Chapter 4

2.3. Intelligent Agents in Business Process Management Systems

There are two scenarios, how agents can be used in business process management: agent-supported BPM and agent-driven BPM [12, 123]. In the first scenario intelligent agents support business processes, which are operated by BPMS, for example, agents may help users to complete tasks assigned during a business process, handle work items automatically or ensure communication between BPMS and external applications. In the second – agents are driving the process encapsulating the whole logic.

Analyzing implementation of agents in both scenarios [12, 123] the following conclusions are made:

1) Agent-supported BPM scenario can be used with any BPMS that provides integration protocols for external systems, for example, web services;

2) To enable agent-driven BPM scenario BPMS must be implemented in specific way, named Agent-BPMS architecture. Agent-BPMS is multi agent system where Each agent is responsible for one or multiple workflow steps, and communicating with other agents decides which step should be executed next [27, 42];
3) Combining both scenarios allows to ensure agent-driven execution for some part of business process that is operated by WBPMS.

In cloud-BPMS scenarios developers cannot affect architecture and implementation of BPMS system, thus cannot implement full agent-driven BPM scenario. However, cloud-BPMS usually provide various integration interfaces [55], which may be used to implement agent-supported BPM scenarios. It also allows to combine agent-supported and agent-driven BPM approaches to ensure that on some stage business process is executed in agent-driven mode.

In the thesis MAs are used to address specific cloud-BPMS integration requirements. Developed integration approach is shown on Figure 2.3. Organization is using a cloud-BPMS through an existing internet connection. Organization's internal environment consists of employees, various devices, information systems and data storages. To integrate this internal environment with cloud-BPMS a MA infrastructure should be deployed in organization. On client side this requires installing agent execution environment software on devices. In addition a communication proxy (agency) should be established. This may be deployed as part of cloud-BPMS, part of organization's internal infrastructure, or a separate cloud service. The communication proxy should provide agent queue that receives agents and routes them to appropriate destination (BPMS or agent host). Execution of MAs is initiated in BPMS. MAs are traveling in the network and perform operations on different hosts. While transporting agents are packaged to special container (agent package). Details of implementation and approbation of the system are covered in Chapter 4.

There are several benefits in using mobile agents in cloud-BPMS integration scenarios [62]. First, integration logic is stored in the agent itself (on BPMS side) what makes implementing rapid changes easier. Agents may also perform computation and transformations close to data sources as they may move between hosts in internal network. Deploying additional hosts (agent execution environments) close to data sources is also trivial what is especially important in highly dynamic environments. Finally, by implementing custom agent activities it is possible to integrate agents with legacy systems and specific devices in internal network. This may be achieved by modifying the agent definition only and without the need to deploy any specific components in client infrastructure. As a result proposed MA based approach addresses all specific cloud-BPMS integration requirements that were identified during the research.

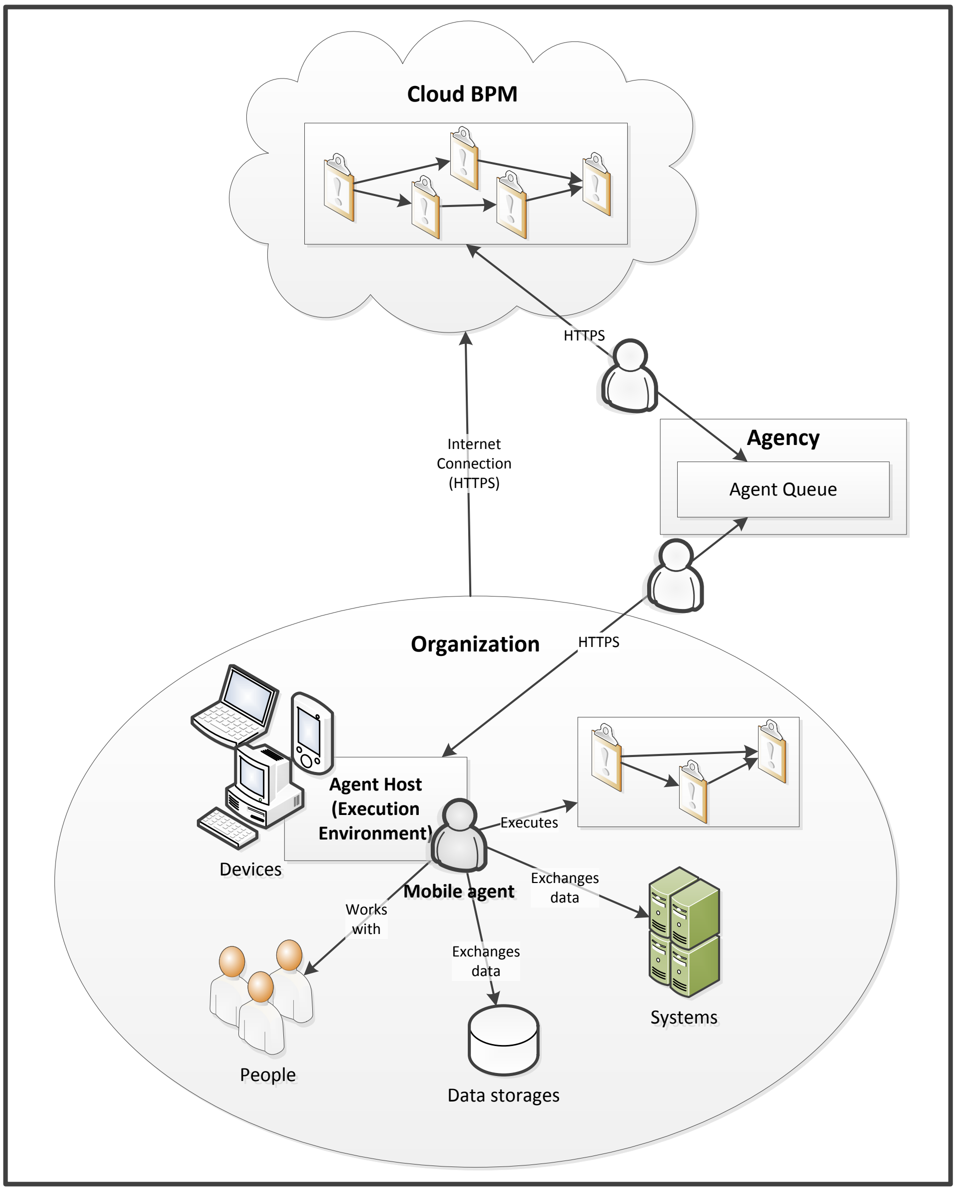

Figure 2.3. Mobile agents in cloud-BPMS integration scenario [62]

2.4. Conclusions

Analysis of business process management systems led to the following conclusions:

- Nowadays when implementing enterprise information systems the goal of developers is to ensure execution of entire business process by building integrations with existing specific systems that are used to work with necessary data;
- When implementing enterprise information systems in cloud computing environments it is important to ensure integration with information systems that are deployed in local network of the organization;
- Cloud-BPMS has additional integration requirements that cannot be fully addressed by any existing technical solution;
- In the thesis new integration solution is proposed that is based on mobile agents and addresses all specific cloud-BPMS integration requirements.

The following theoretical results are achieved in this chapter:

- Four specific cloud-BPMS integration requirements are identified;
- Analysis of how these specific integration requirements are addressed by using existing technical solutions is performed;
- As a result of analysis necessity to use MAs is justified;
- The approach for integrating cloud-BPMS with on-premises systems and devices is developed.

3. INTELLIGENT AGENTS

In the second chapter by analyzing business process management systems (BPMS) it is concluded that when implementing enterprise information system in cloud computing environment it is important to ensure integration with information systems that are deployed in local network of the organization. Cloud-BPMS has additional integration requirements that cannot be fully addressed by using any existing technical solution. These requirements can be addressed by using mobile agents (MA).

In this chapter the concept of intelligent and mobile agents (MA) is covered, architectures and implementation aspects of MA systems are analyzed and newly proposed workflow based approach for designing and executing mobile agents and common MA package format are introduced.

3.1. Intelligent Agent Systems

Agent technologies are one of the most actively researched areas in the last two decades. As is to be expected from a rather young area of research, even such fundamental concept as “an agent” has not been defined in a single unified and widely accepted definition. The agent metaphor subsumes both natural and artificial systems. Several approaches were made attempting to define what may be considered as an agent [65]. If the most general approach is used then “an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors” [86]. An agent may be considered as a computer system that is situated in some environment and is capable of autonomous action in order to meet its design objectives [119].

From the software point of view agents are a natural progression from objects. There is an ontological distinction between agents and objects [112]. Only agents are active entities that can perceive events, perform actions, solve problems, communicate and make commitments. Objects are passive entities with no such capabilities.

From the structural point of view an agent is a program and architecture. The initial phase for an agent program is to understand and describe percepts, actions, goals and environment [26].

Agent architecture specifies the decomposition of an agent to a set of modules and relationships between these modules. Several simple agent architectures are described in [86]. The most interesting architectures from the BPM point of view are goal-based, utility-based, learning, Belief-Desire-Intension (BDI) [120] and mobile agents (MA) [48, 49].

MAs are covered in more details as these are the key element of solution proposed in the doctoral thesis.

3.2. Mobile Agent Systems

Mobile agents (MA) are self-contained and identifiable computer programs that can move within the network and act on behalf of the user or another entity. During the move entire program code and execution state is transferred. When MA reaches destination host execution of the program is continued from the last saved state [49, 50, 66]. MA systems are platforms that provide capabilities to create, execute, transfer and terminate MAs. Not all MA systems provide support for state mobility [20, 104]. The term strong mobility is used to describe systems that can capture and move execution state with the agent. In agents that support strong mobility it is guaranteed that all variables will have identical values after moving to another host. Weakly MA systems usually support the capture of most of a program's data, but restart the program from a predefined program point and thus require some programmer involvement at each migration. The advantage of strong mobility is that the result of migrating is well defined and easier to understand.

There are two main scenarios that must be addressed by MA systems [59]: creating an agent and transferring an agent. To create an agent MA system creates an instance of the agent class on a default host, or a host the client specifies. The agent class specifies the implementation of the agent. During the execution MA may request the source agent system for a transfer to another host or agent system. To start this process host or agency on a sender's side needs to initiate agent's transfer: suspend the agent, identify transferable agent's state, serialize the agent class and state, encode it for the chosen transport protocol, provide authentication information to the server, and transfer the agent. To receive an agent destination host needs to authenticate client, decode the agent, deserialize the agent class and state, instantiate the agent, restore the agent state and resume agent execution. Serialization is the process of storing an agent in a serialized form, sufficient to reconstruct the agent. Deserialization is the inverted process [59].

There are two general approaches of transferring MAs: transferring via central agency and direct transferring between hosts [101]. In the first scenario source host sends MA to central agency, and then MA is transferred to target host. In the second scenario source host sends MA directly to target host, what allows to reduce the load on networks as MA is transferred only once. In this case source and destination hosts must be connected to single network. If network connection between hosts is not available only the first approach can be used.

A security aspect of MA as a technology is raised as the biggest concern by various authors [9, 19, 25, 64]. It is important to protect agent hosts during MA execution and agents itself. Designing MA system developers may use the following techniques to protect agent hosts: authenticating credentials, access-level monitoring and control, code verification, limitation techniques (time, range, and duplication limits) and audit logging. In addition, the following techniques may be used to protect agents itself: fault tolerance and encryption (code obfuscation, encrypted data manipulation).

There is no reference architecture for MA systems defined. While implementing common requirements of MA systems developers had created different architectures of MA systems. Analyzing implementation and architectures of known MA systems several base components can be identified (Figure 3.1) [90, 91, 104]. These components allow to ensure the entire lifecycle of MA.

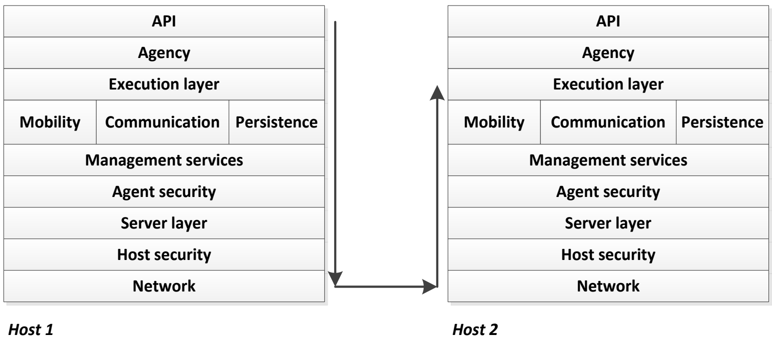

Figure 3.1. Components of mobile agent system (adapted from [91])

MA lifecycle consists of multiple stages. First user initiates agent execution by using system API. API forwards the request to agency that provides core capabilities of MA system, and the system initiates MA execution. Mobility and communication components ensure core mobility and communication capabilities of MAs. Persistence component ensures fault tolerance while executing MAs. Management services implement serialization and conversion operations that are used to prepare agent for transferring. Agent security component encrypts the entire MA or separate parts to ensure that these are not modified while transferring MA over the network. Server layer provides capabilities to generate unique names for agents, what allows to track agents during execution. Host security component assigns credentials that are required for authentication in MA systems or hosts. Network component encrypts MAs according to requirements of transport protocol that is used. After transferring is completed, network component decrypts received request. Host security component validates authentication credentials. Then unique name of MA is registered at server layer, agent security components decrypt MA, but management services perform deserialization and conversion operations. Finally execution of MA is continued at destination host.

There are many MA platforms available, with over 100 such systems implemented in the last five years. However, according to researchers the following may be considered as the most popular: *Telescript, NOMADS, SafeTCL, D'Agents, JavaSeal, Mole, Aglets, Lime, Messenger, JADE, Voyager, TACOMA, Grasshopper, SPRINGS, MAPNET* and *EtherYatri.NET* [30, 81, 104]. Analyzing existing systems researchers conclude that weak mobility is by far the predominant approach [104]. Java is still the most popular agent implementation language, mainly due to both the popularity of the language and its support for dynamic loading and advanced security features [30, 104]. Microsoft .NET is more recent technology, which is also based on a virtual execution environment and supports the dynamic loading of programs, and is becoming more popular in recent MA implementations [81, 104].

Analysis of existing MA platforms allows to conclude the following [63]:

1) Known MA systems have similar requirements regarding the information that must be included in transferred MA package, however there is no common approach for packaging MAs. Every implementation uses specific form of serializing agent state and data, leveraging capabilities of technical platform used to implement MA system. In the doctoral thesis common MA package format is developed, which addresses generic requirements of MA systems (which are, for example, support for compression, digital signatures and encryption) and is extendable, what allows it to be adjusted to additional requirements of specific scenario;
2) Analyzing agent development experience it should be noted that also some platforms are providing graphical user interface, MAs are developed by using some programming language. It is expected that user creating MA has at least basic programming skills. In MA systems implementing weak mobility approach details of saving agent’s state and restarting execution are also not obvious for end-users with no technical knowledge. As a result users having no programming experience cannot develop MAs. This makes existing systems inappropriate for cloud-BPMS integration scenarios, where users having no programming skills must be able to implement rapid changes in MA behaviors. To solve this problem new workflow based approach for designing and executing mobile agents is developed in the doctoral thesis.

3.3. Workflow Based Mobile Agent System

3.3.1. Agent Development and Execution Process

Workflow based MA system developed in the thesis relies on using workflows for designing and executing MAs. MA design and initiation process is organized in three steps (Figure 3.2) [62]:

1) Business analyst defines agent behavior using special application proposed in the thesis that allows defining sequential agent behaviors as steps of a workflow. As a result Agent Definition package is created (.awfd file);
2) Agent Definition package is then uploaded to BPMS;
3) BPMS initiates execution of an agent. At this point Agent Instance package is created (.awfx file).

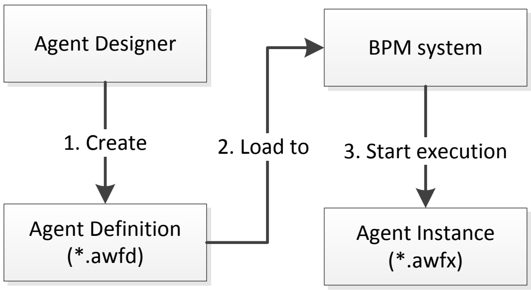

Figure 3.2. Agent development process [62]

Figure 3.3 shows example of Agent Designer application that was developed as part of *AgentWF* prototype. While designing MA analyst defines agent workflow (central part of the tool) by adding steps from the set of available activities (left tool pane). Agent migration to another host is initiated by using special workflow activity called *MoveAgentToHost*, providing the name of target host as an attribute.

Figure 3.4 shows MA execution process that contains the following steps:

1) External system or user is initiating the process, by sending MA package and specifying the name of the host, where agent must be executed: *Host1*. Agency handles this request and saves MA package in agent queue;
2) Hosts are regularly sending requests to the agency to check if there are any agents registered in agent queue that are waiting for execution on this host. After sending this request *Host1* detects that there is MA that is waiting for execution and downloads MA package;
3) *Host1* deserializes MA package and starts execution of agent workflow;
4) When the workflow reaches *MoveAgentToHost* activity with attribute *Host2*, executing host receives transfer request from MA;

5) *Host1* stops execution of MA, saves execution state and information about target host (*Host2*) into MA package, and sends MA package to the agency. The agency handles the request and saves MA package into agent queue;
6) *Host2* detects that there is an agent waiting for execution and downloads MA package;
7) *Host2* deserializes MA packages and continues execution of agent workflow from saved state;
8) When all workflow activities are executed, *Host2* receives notification that execution of MA is completed;
9) *Host2* saves final state of MA into MA package and sends MA package to the agency. The agency handles the request and saves MA package into the queue of completed MAs;
10) The agency sends MA package from the queue of completed MAs to the system that has requested MA execution.

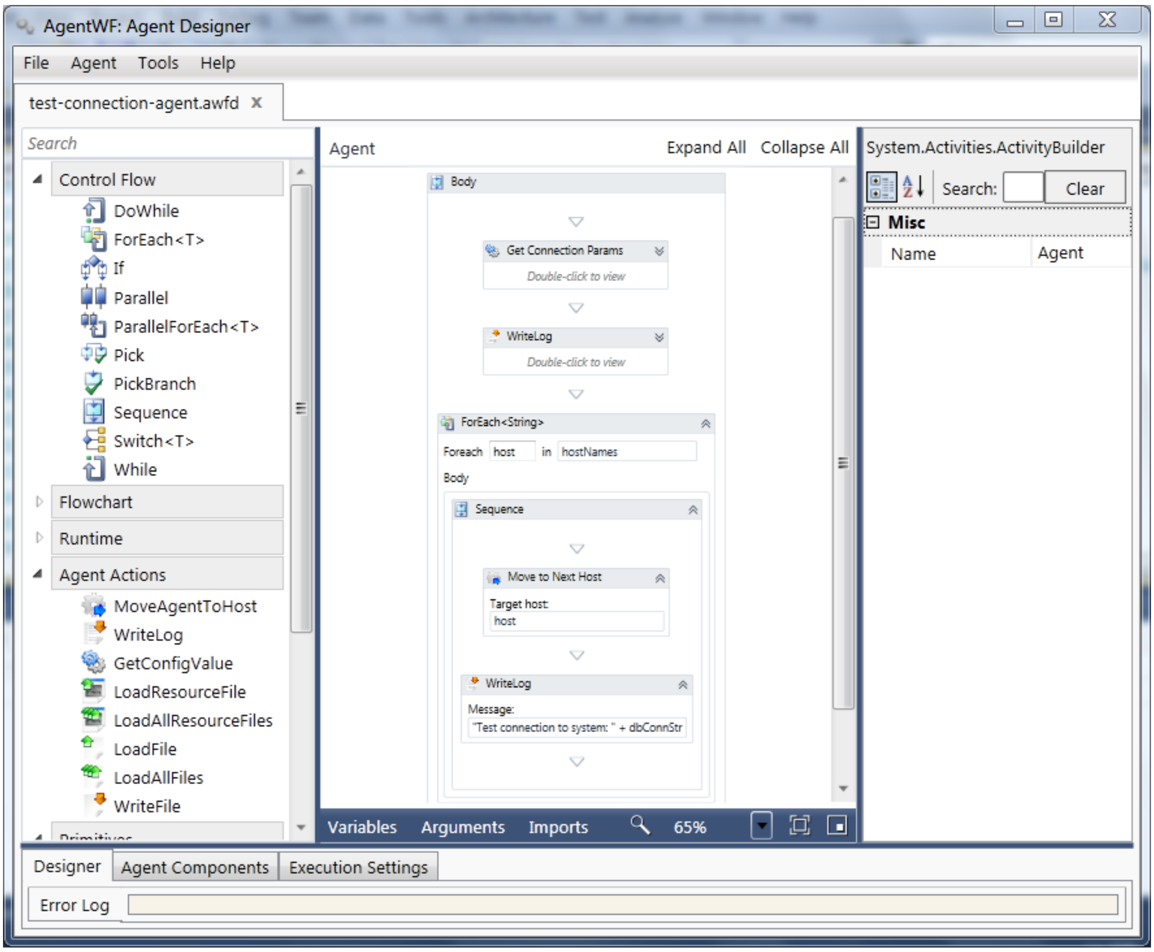

Figure 3.3. *AgentWF* Agent Designer application

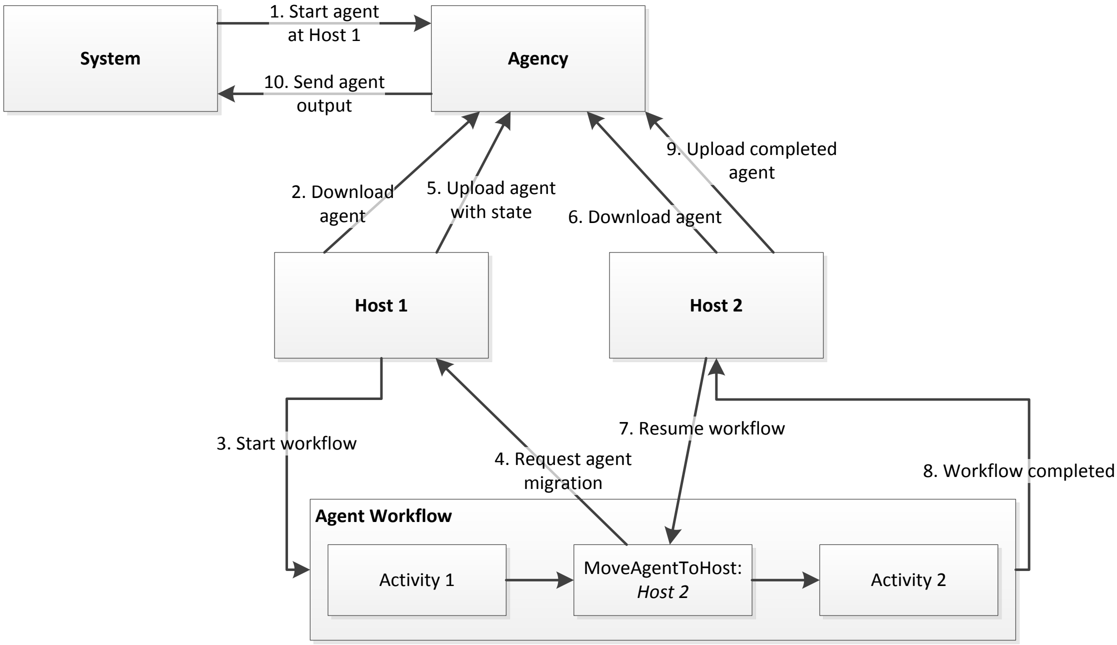

Figure 3.4. Agent execution process

Agent transferring is implemented by using transferring via central agency approach. This allows to transfer MAs in scenarios when hosts are not deployed in single network or are dynamically moving in the network.

Workflow based approach for designing and executing mobile agents allows to ensure strong mobility. While transferring MA the package contains complete workflow execution state. After transferring to target host is completed workflow execution is continued from the last saved state. The process of transferring MAs does not require programmer involvement.

3.3.2. Agent Package Format

Introducing common MA package format could allow not only to establish interoperability between different MA systems (what is already addressed in other existing researches and is driven by introducing common MA standards [59, 76]), but also to use common tools for authoring and monitoring MAs [63].

MA package is hierarchical container built according to Open Packaging Conventions (OPC) specification. OPC describes a way how to pack any information. The package is constructed from parts and relationships between them. OPC is based on open technologies (XML and ZIP file formats) and is documented as part of ECMA specification for Open XML formats [100].

OPC is applicable for creating MA packages because of the following features [63]:

- OPC is open standard;
- MA package may contain any data (also encrypted);
- format is extendable what allows adapting MA package to specific requirements later;

- format supports digital signatures;
- MA package is archived using ZIP algorithm what allows to reduce amount of information that is transferred over the network.

Different MA systems have similar requirements regarding the information that must be included in MA packages. After analyzing these requirements the following MA package objects were identified [63]: package manifest, agent properties, agent body (agent program), agent state, input data, output data, supporting files, resource files, supporting assemblies, agent execution log and configuration settings (Figure 3.5).

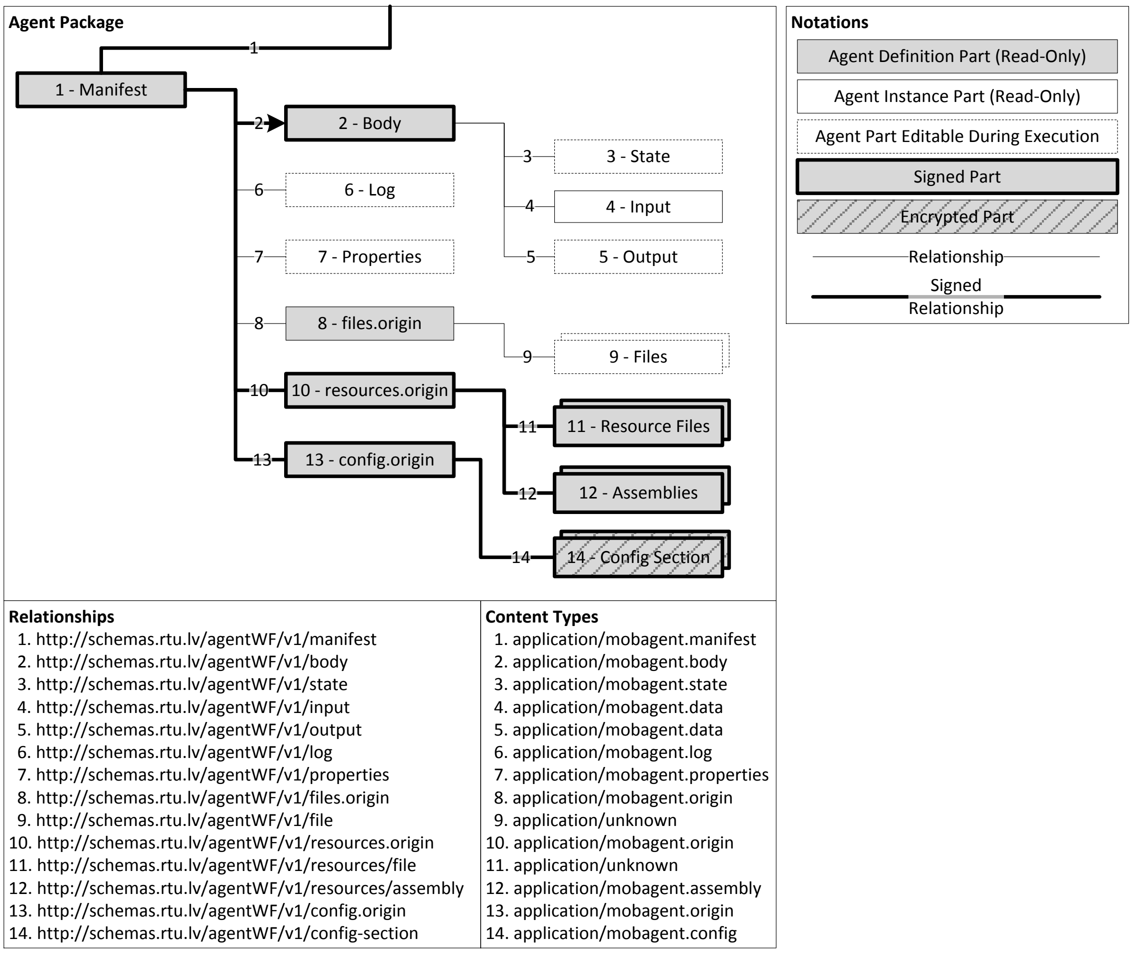

Figure 3.5. Agent package [63]

MA package objects and relationships are split into three categories:

- agent definition components are digitally signed, to ensure that these are not modified while creating agent instance or executing MA;
- agent instance components can be created or modified when creating MA instance, but cannot be modified during MA execution;
- components that can be modified during MA execution.

Figure 3.5 shows package objects and relationships. Relationships and objects are numbered to describe types for relationships and content types for objects. It also shows three categories of package components described above, as well as components that are digitally signed and objects that are encrypted.

MA package format was implemented as part of *AgentWF* prototype.

3.4. Conclusions

The main conclusions are the following:

- Mobile agents (MA) are self-contained and identifiable computer programs, that can move within the network and act on behalf of the user or another entity. During the move entire program code and execution state is transferred;
- Not all MA systems provide support for state mobility. The term strong mobility is used to describe systems that can capture and move execution state with the agent. Weakly MA systems usually support the capture of most of a program's data, but restart the program from a predefined program point and thus require some programmer involvement at each migration. The advantage of strong mobility is that the result of migrating is well defined and easier to understand;
- There are two general approaches of transferring MAs: transferring via central agency and direct transferring between hosts. If network connection between hosts is not available only the first approach can be used;
- When designing MA system it is important to protect agent hosts during MA execution and agents itself during execution and transfer;
- Known MA systems have similar requirements regarding the information that must be included in transferred MA package, however there is no common approach for packaging MAs defined;
- In the doctoral thesis common MA package format is developed, which addresses generic requirements of MA systems and is extendable, what allows it to be adjusted to additional requirements of specific scenario;
- Analyzing agent development experience in known MA systems, it is noted that also some platforms are providing graphical user interface, MAs are developed by using some programming language. It is expected that user creating MA has at least basic programming skills, what makes existing systems inappropriate for cloud-BPMS integration scenarios;
- In the doctoral thesis new workflow based approach for designing and executing mobile agents is developed. MAs are designed by using special graphical user

application proposed in the thesis that allows defining sequential agent behaviors as steps of a workflow. Proposed approach allows to ensure strong mobility.

The following theoretical results are achieved in this chapter:

- Analyzing known MA systems two aspects are identified, that must be improved to use MAs in cloud-BPMS integration scenarios:
    - Users having no programming skills cannot develop MAs;
    - No common format for packaging MAs is defined;
- New workflow based approach for designing and executing mobile agents is developed that allows users having no programming skills to develop MAs;
- Common MA package format is developed, which addresses generic requirements of MA systems and is extendable, what allows it to be adjusted to additional requirements of specific scenario.

4. MOBILE AGENT SYSTEM FOR CLOUD-BPMS INTEGRATION

In the third chapter proposed workflow based approach for designing and executing MAs and MA package format are introduced.

This chapter covers proposed MA system architecture, *AgentWF* prototype implementation details, approbation scenarios and results.

4.1. Design and Architecture of Mobile Agent System

4.1.1. System Architecture

In the thesis as a result of analysis of business process management systems (BPMS) and mobile agent (MA) systems several requirements where identified that must be addressed while implementing MA system to integrate cloud-BPMS with on-premises systems and devices:

1) Ensure integration with cloud-BPMS without the need to reconfigure firewalls or network infrastructure;
2) Address specific integration requirements of cloud-BPMS;
3) Ensure scalability at all levels of the system and mechanisms to add new components dynamically to support evolution of the system over time;
4) Ensure core requirements of MA systems;
5) Address common security aspects while handling enterprise data;
6) Support transferring MAs in complex network infrastructures – when agent hosts are not connected to single network or are dynamically moving in the network;
7) Allow users having no programming skills to develop MAs.

To address these requirements new MA system architecture is developed in the doctoral thesis. The system implements agent-supported BPM approach. Design and execution of MAs is performed by using workflow based approach developed in the thesis. MAs are stored and transferred in packages that are created by using common MA package format developed in the thesis. Figure 4.1 shows architecture of the system.

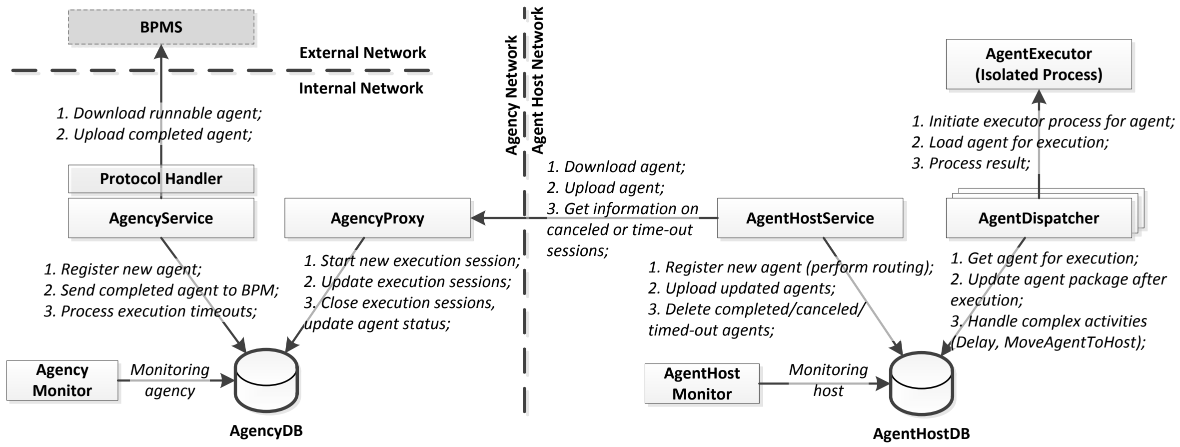

Figure 4.1. System architecture [62]

Components of MA system are split into three categories: MA design tools, agency components and agent host components.

Agency consists of the following five components:

- Protocol Handler that ensures communication with specific BPMS;
- Agency Service that ensures core agency operations and communication with BPMS;
- Agency Database that stores information about agent hosts, MAs and MA execution sessions;
- Agency Proxy that handles requests from agent hosts;
- Agency Monitor that is used to monitor current state of the agency.

Agency scalability is achieved by deploying additional instances of Agency Services and Agency Proxies, and by increasing performance of database servers hosting Agency Database.

Agent host consists of the following five components:

- Agent Host Service that ensures core agent host operations and communication with agency;
- Agent Host Database that stores information on MAs and their execution states;
- Agent Dispatcher service that manages execution of MAs;
- Agent Executor that is used to execute MAs in isolated process;
- Agent Host Monitor that is used to monitor current state of the agent host.

Scalability of agent hosts is achieved by deploying additional instances of Agent Host Services and Agent Dispatchers, and by increasing performance of database servers hosting Agent Host Database.

System architecture is designed the way that all communication is initiated from client-side only: from agency to BPMS and from agent hosts to agency. Communication is performed through widely adopted protocols HTTP and HTTPS. Proposed architecture is based on the following standards and open specifications: OPC [100], XML [18], XML encryption [122], XAML [17, 121], JSON [45], X509 certificates (digital signing and encryption) [38, 39]. MA execution process is organized the following way:

- MA execution is initiated in external system. External system defines from which agent definition MA instance must be created. It also provides MA instance initialization parameters: input data, supporting files and the name of the host where MA execution must be started;
- Agency Service is regularly sending requests to external system in order to check if there are MAs that are awaiting execution in this agency. To start execution of MA, Agency Service downloads MA definition and instance initialization parameters, creates new MA instance and assigns unique identified (ID) to it. MA instance is registered in Agency Database. In addition Agency Service can validate digital signature that is used to sign MA definition, and accept for execution only agents that are signed with specific certificates;
- Agent Host Service is regularly sending requests to Agency Proxy in order to check if there are MAs that are awaiting execution on this host. If such MA exists Agency Proxy registers new agent execution session in Agency Database and sends MA to the host. Agent Host Service downloads MA and registers it in Agent Host Database. By analyzing certificate used to sign MA package Agent Host Service determines which Agent Dispatcher will handle execution of MA. This allows executing different MAs using different security settings, as every Agent Dispatcher is running as isolated process for which specific permissions to access resources of local system may be granted. From this moment Agent Host Service must regularly send notifications to Agency Proxy that processing of specific agent execution session is in progress. If Agency Proxy does not receive such notifications it assumes that processing of agent execution session was aborted (for example, as a result of hardware failure on the host). In this case Agency Service deletes registered agent execution session and updates execution state of MA back to “waiting for execution”;
- Agent Dispatcher is regularly querying Agent Host Database in order to get information on MAs that must be executed in this process. MAs are executed in isolated Agent

Executor process. For every MA Agent Dispatcher creates new isolated process and loads workflow execution environment and global assemblies (assemblies that are available to all MAs which are executed by this Agent Dispatcher). Next, information from MA package is loaded into this process: MA assemblies, MA workflow and last saved state. When information from MA package is loaded, Agent Dispatcher starts MA execution and waits for completion. MA execution time is limited – if MA execution is not completed during this time interval, Agent Dispatcher aborts MA execution and assumes that MA execution is completed with an error. When MA execution is completed Agent Dispatcher updates MA package and saves it to Agent Host Database. After this MA is processed by Agent Host Service and sent back to Agency Proxy. Finally Agency Proxy updates MA package in Agency Database. Agent Dispatcher also handles complex MA workflow activities – transfer (*MoveAgentToHost*) and delay (*Delay*) requests. After receiving transfer request Agent Dispatcher saves updated MA package in Agent Host Database specifying the name of target host. In case of delay request Agent Dispatcher saves updated MA package in Agent Host Database specifying time when MA execution must be continued;

- Agency Service is regularly querying Agency Database in order to get information on completed MAs and uploads these to external system that has initiated MA execution.

4.1.2. Cloud-BPMS Integration Approach

The following four components are required to integrate MA system with cloud-BPMS [61]:

- Protocol Handler – ensures communication with specific BPMS;
- Agent Definitions Store – stores agent definitions;
- Agent Instance Store – stores agent instances;
- Workflow Actions – allows to use MA specific activities while defining workflow models (for example, initiate MA execution, or handle MA execution result).

Figure 4.2 shows these components and their relations with agency and agent designer application.

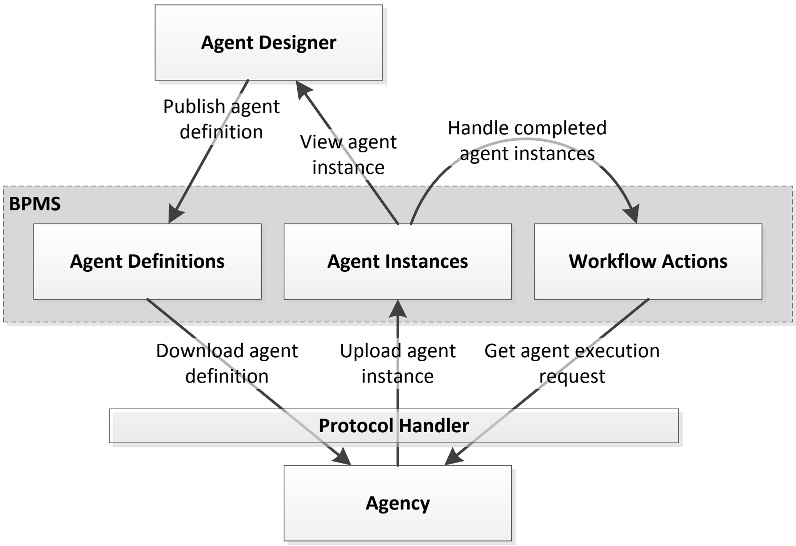

Figure 4.2. Integration with cloud-BPMS [61]

Protocol Handler is mandatory component that is required to integrate MA system with BPMS. To implement Protocol Handler developers use integration interfaces provided by BPMS. If BPMS is not providing such interfaces, implementing Protocol Handler is not possible. All communication between agency and BPMS happens through Protocol Handler.

Agent Definitions and Instances Stores are mandatory components that should be implemented as extensions to BPMS. In this case workflow models and definitions for MAs used in these are stored in single location (cloud-BPMS). This also allows to avoid developing separate mechanism to transfer MA execution results back to BPMS, as this is a part of MA instance stored in BPMS. Agent definitions are stored as MA package files (.awfd). Agent instances may be stored as MA package file (.awfx) or as unpacked collection of files (this simplifies analyzing MA execution results, as raw files are available and handling MA package is not required).

If storing Agent Definitions and Instances in BPMS is not possible, separate storage must be implemented. This should address the following requirements [61]:

- It allows to store MA package files (.awfd and .awfx);
- It allows to publish MA definitions developed in Agent Designer application;
- It provides communication protocols through which agency may download MA definition packages and upload MA instances.

In this scenario it is also required to implement components that ensure information exchange between agency and Agent Definition and Instances Stores. In addition, it is also required to implement mechanism that transfers MA execution result into BPMS.

Agent Workflow Actions allow using MA specific activities while defining workflow models. Implementing the following activities is recommended [61]:

- MA initialization, specifying agent definition and initiation parameters;

- Starting MA execution;
- Handling MA execution results.

If BPMS does not provide mechanisms to extend workflow models with custom activities, MA integration should be implemented using existing workflow activities. For example, integration may be achieved by using standard workflow tasks [61]:

- To initiate MA execution new task is created, including agent initiation parameters in task description;
- When MA execution is completed, task status is changed to completed, and agent execution results are included in metadata of the task.

In the thesis a special algorithm is proposed that allows to determine how BPMS integration components should be implemented based on technical specification of cloud-BPMS (Figure 4.3).

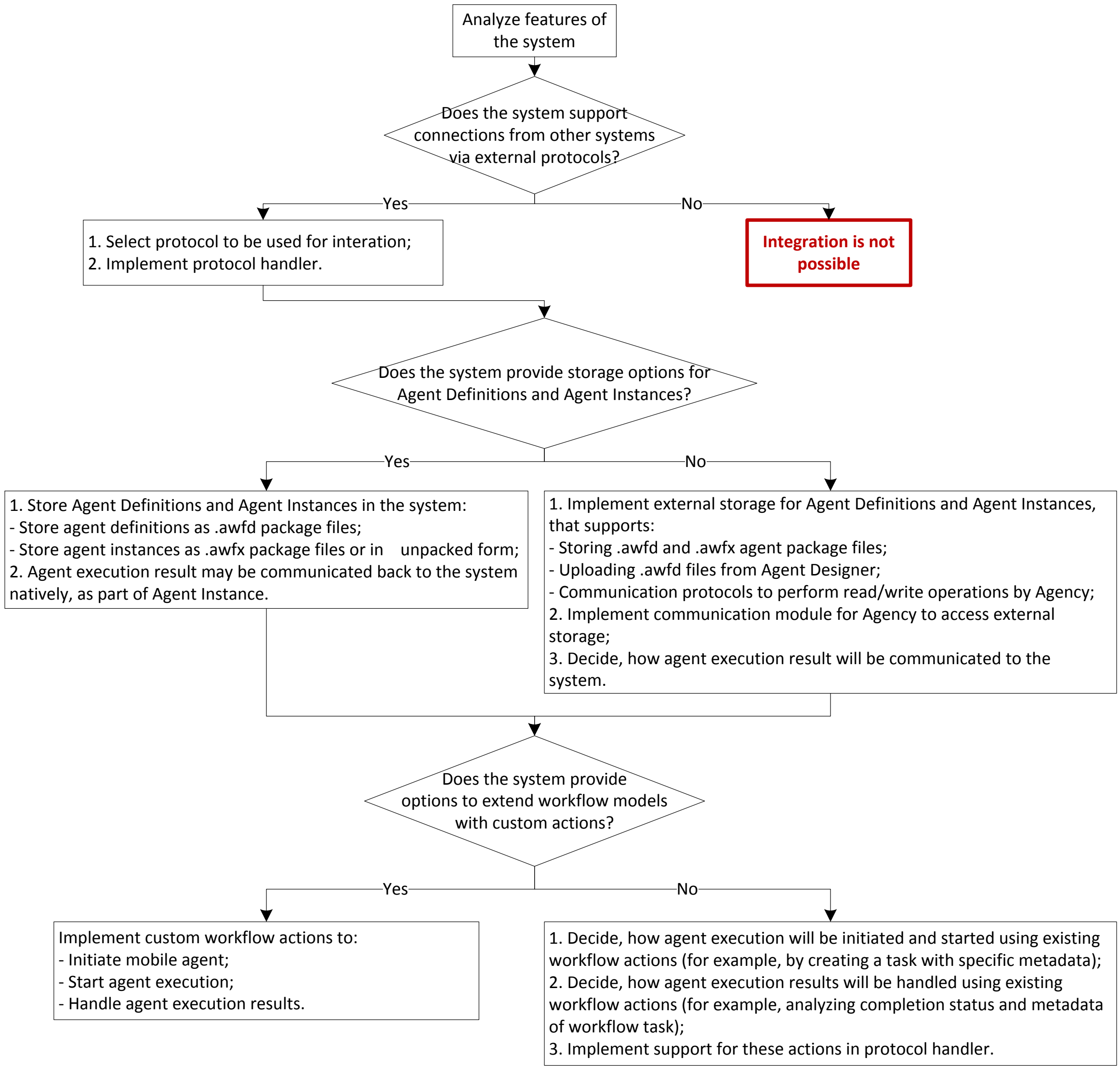

Figure 4.3. Algorithm used to analyze cloud-BPMS integration options [61]

4.2. Overview of AgentWF Prototype

To validate proposed architecture MA system prototype called *AgentWF* is developed. It implements all components of proposed architecture. *AgentWF* is built on Microsoft .NET Framework 4.0 technology what allows to use the following built-in features of the platform:

- *Windows Workflow Foundation (WF)* [117] – designing and executing workflows;
- *Windows Communication Foundation (WCF)* [115] – implementing communication between components of the system;
- *Windows Presentation Foundation (WPF)* [116] – implementing graphical user interfaces;
- *Packaging API* [105] – handling OPC [100] packages;
- *AddIn Framework* [1] – executing MAs in isolated environment;
- *Reflection* [82] – dynamic loading of assemblies and classes;
- *Serialization* [92] – serializing and deserializing information in JSON, XAML and XML formats.

SQL Server Compact edition [98] is used as database management system to operate databases of agencies and hosts. This allows to simplify deployment of new agencies and hosts as installing and configuring database server is not required. *AgentWF* is not using any specific features of *SQL Server Compa*ct, thus solution can be used with other *SQL Server* editions as well [99].

To protect MAs and hosts the following security techniques are implemented in the prototype:

- Authenticating Credentials – separate parts of MA package are signed using X509 certificates to ensure that these are not modified while executing or transferring MAs;
- Access-Level Monitoring and Control – MA is executed in isolated environment. Depending on certificate that is used to sign agent package the agent is executed with different level of access privileges;
- Limitation Techniques – execution of agents is limited by various time metrics: for example, maximum execution time in the system, or maximum execution time for session on agent host;
- Audit Logging – information on all MA activities is recorded in agent log;
- Fault Tolerance – the state of MA workflow is persisted, execution is continued from last saved state;
- Encryption – sensitive configuration information in agent package gets encrypted with X509 certificates. It can be decrypted and accessed only on agent hosts where specific certificate with private key is installed.

On Figure 4.4 MA system architecture (Figure 4.1) is supplemented with *AgentWF* implementation details.

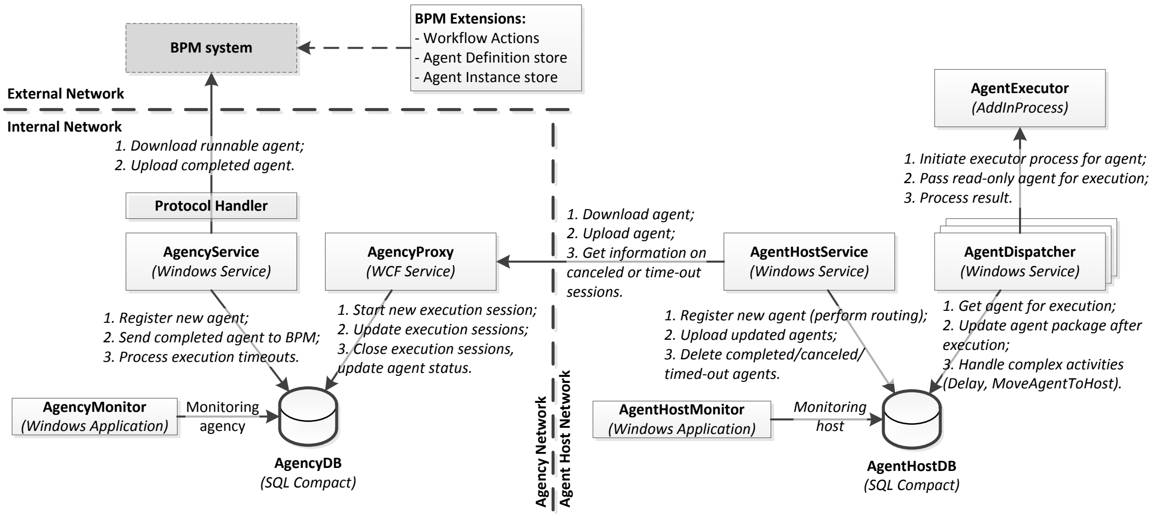

Figure 4.4. *AgentWF* architecture [62]

4.3. Approbation

To validate proposed BPMS integration approach, MA design and execution process and MA system architecture, several approbation scenarios are implemented using *AgentWF* prototype. Approbation scenarios were adapted to local specifics by selecting technologies that are available or widely adopted in Latvia:

- *SharePoint Online* (part of *Office 365*) is used as cloud-BPMS [69];
- *Tildes Jumis* accounting system is used as example of existing enterprise system deployed in local network [106];
- Latvian identification card (eID) is used to sign MAs [14].

The following six approbation scenarios are implemented:

1) Designing and executing MAs: designing MAs using workflow based tool; ensuring strong mobility while transferring MAs; aborting MA execution in case of incorrect behavior;
2) Protecting sensitive MA configuration information by using encryption;
3) Integrating MA system with cloud-BPMS using *SharePoint Online* system as an example;
4) Signing MA packages using Latvian identification card (eID);
5) Integrating cloud-BPMS with on-premises application: using data from *Tildes Jumis* in *SharePoint Online* workflow;
6) Integrating cloud-BPMS with devices in local network: using local printer in *SharePoint Online* workflow.

The following conclusions are made as a result of approbation:

1) Proposed architecture of MA system enables integration of cloud-BPMS with information systems and devices in local network of an organization;
2) Graphical workflow based MA design tool allows developing MAs without programming;
3) The system ensures strong mobility:
   - while transferring the entire workflow state is included in MA package;
   - MA execution is continued from last saved workflow state;
   - the process of transferring MAs does not require to involve programmers;
4) Involving programmers is required in order to develop:
   - additional agent workflow actions that are required, for example, to integrate MAs with specific information systems or devices;
   - protocol handlers to integrate MA system with BPMS;
5) The process of installation and configuration of agencies and agent hosts requires specific technical knowledge. To simplify this process it is recommended to extend the system with special deployment wizards and graphical configuration tools for agencies and agent hosts in the future;
6) Proposed BPMS and MA integration approaches analysis algorithm allows to determine possible integration scenario and components required for integration. To enable integration BPMS must provide interfaces for integrating with external systems. Analyzing other BPMS technical features developers should decide where to store agent definitions and instances, how to pass MA execution results to BPMS, and how to integrate MA execution initiation and completion operations with BPMS workflows and workflow design tools. In case of *SharePoint Online* all identified components were implemented by using extensibility options provided by BPMS;
7) Approbation scenarios have shown that proposed integration solution addresses common security aspects while handling enterprise data:
   - Communication with BPMS is performed using secure channel (HTTPS);
   - Special service account is configured in BPMS. Minimum required access privileges are granted to this account: read-only access to agent definitions library and read/write access to agent instances library;
   - MA system is using credentials of this service account when connecting to BPMS;
   - MA configuration parameters that contain sensitive information, for example, connection strings, are encrypted;
   - MAs are digitally signed. Before executing MA the signature is validated to ensure that MA package was not modified. Agency configuration has a setting allowing

to specify which agents are approved for execution by the agency, based on certificate used for signing MA package;

8) X509 certificates are used for signing and encryption. This allows to use certificates issued by existing certification authorities, for example, signing certificates on Latvian identification card;
9) MA based integration solution allows to implement asynchronous integration scenarios. Proposed solution is not applicable in scenarios, when it is required to ensure:
    - real time synchronous integration (BPMS sends request and is waiting for immediate response from enterprise information system);
    - minimal possible request processing time (as transferring MAs significantly increases request processing time).

4.4. Conclusions

The main conclusions are the following:

- MA system architecture developed in the doctoral thesis implements agent-supported BPM approach. Components of MA system are split into three categories: MA design tools, agency components and agent host components. Design and execution of MAs is performed by using workflow based approach developed in the thesis. MAs are stored and transferred in packages that are created by using common MA package format developed in the thesis. Proposed architecture is based on the following standards and open specifications: OPC [100], XML [18], XML encryption [122], XAML [17, 121], JSON [45], X509 certificates (digital signing and encryption) [38, 39];
- The following four components are required to integrate MA system with cloud-BPMS: protocol handler, agent definitions store, agent instances store and agent workflow actions. There are several ways to implement these components. In the thesis a special algorithm is proposed that allows to determine how BPMS integration components should be implemented based on technical specification of cloud-BPMS. It is recommended to implement stores for agent definitions and instances as extensions to BPMS. If BPMS provides mechanisms to extend workflow models with custom activities, it is recommended to implement the following MA specific workflow actions: MA initialization, starting MA execution and handling MA execution results;
- To validate proposed architecture MA system prototype called *AgentWF* is developed. It implements all components of proposed architecture;

- To validate proposed BPMS integration approach, MA design and execution process and MA system architecture, six approbation scenarios are implemented using AgentWF prototype;
- Approbation results prove that developed MA system architecture implements all requirements that have been defined during analysis;
- Proposed MA based integration solution allows implementing asynchronous integration scenarios. It is not applicable in scenarios, when it is required to ensure real time synchronous integration or minimal possible request processing time.

The following theoretical results are achieved in this chapter:

- the algorithm that allows to determine possible approach to integrate MAs with BPMS has been developed. It allows to identify components required for integration;
- the architecture of MA system for integrating cloud-BPMS with on-premises systems and devices has been developed.

The following practical results are achieved in this chapter:

- The prototype *AgentWF* that implements all components of proposed architecture is developed;
- The prototype is approbated to prove that it implements all defined functions.

MAIN RESULTS AND CONCLUSIONS

The goal of the thesis is to develop mobile agent based mechanism that allows integrating cloud-BPMS with existing enterprise information systems and devices, implement this mechanism in software system and perform evaluation of developed system. In order to achieve the goal the following tasks have been performed:

- Analysis of development tendencies and actual research directions in business process management area, and which of existing problems are solved by using business process management systems;
- Analysis of implementation aspects of business process management systems;
- Analysis of specific integration requirements of cloud-BPMS and how these are addressed by existing integration solutions;
- Analysis of existing mobile agent systems and evaluation of suitability for this integration scenario;
- Development of architecture of mobile agent system that enables integration of cloud-BPMS with on-premises systems and devices;
- Development of prototype that implements all components of proposed architecture;
- Evaluation of developed prototype in approbation scenarios.

The research has led to the following conclusions:

- Organizations are interested in BPM area. Companies more often start using specialized enterprise information systems that support BPM implementation – BPMS. Special attention is given to cloud-BPMS technologies that allow organizations to reduce initial costs for implementing BPMS;
- Nowadays when implementing enterprise information systems the goal of developers is to ensure execution of entire business process by building integrations with existing specific systems that are used to work with necessary data. When implementing enterprise information systems in cloud computing environments it is important to ensure integration with information systems that are deployed in local network of the organization;
- Cloud-BPMS have additional integration requirements that cannot be fully addressed by any existing technical solution:
    - Performing complex computations close to data sources in internal network (dealing with large amounts of data);
    - Performing complex transformations and computations with data stored in on-premises application in internal network (for security and privacy reasons);
    - Implementing rapid changes in integrations (to adapt business process to changes in the environment);
    - Accessing legacy systems and specific devices that are deployed on-premises and have no web services or database interface;

  In the thesis new integration solution is proposed that is based on mobile agent (MA) technology and addresses all of these requirements;
- Analysis of existing MA platforms allowed to conclude that there are two unresolved tasks:
    - Known MA systems have similar requirements regarding the information that must be included in transferred MA package, however there is no common approach for packaging MAs defined. Every implementation uses specific form of serializing agent state and data, leveraging capabilities of technical platform used to implement MA system;
    - Analyzing agent development experience, it is noted that also some platforms are providing graphical user interface, MAs are developed by using some programming language. It is expected that user creating MAs has at least basic programming skills. As a result users having no programming experience cannot develop MAs. This makes existing systems inappropriate for cloud-BPMS

integration scenarios, where users having no programming skills must be able to implement rapid changes in behaviors of MAs.

In the thesis all of identified issues are addressed and the following main theoretical results are achieved:

- The approach for integrating cloud-BPMS with on-premises systems and devices has been developed, that is based on the architecture of mobile agent system proposed in the thesis;
- The algorithm that allows to determine possible approach to integrate MAs with BPMS has been developed. It also allows to identify components required for integration;
- MA system that is based on using workflows to design and execute agents has been developed. It allows users having no programming skills to design MAs;
- Common MA package format is developed. It addresses generic requirements of MA systems and is extendable, what allows it to be adjusted to additional requirements of specific scenario.

To validate theoretical results the prototype called *AgentWF* is developed and approbated in six scenarios:

- Designing and executing MAs;
- Protecting sensitive MA configuration information by using encryption;
- Integrating MA system with cloud-BPMS using *SharePoint Online* system as an example;
- Signing MA packages using Latvian identification card (eID);
- Integrating cloud-BPMS with on-premises application: using data from *Tildes Jumis* in *SharePoint Online* workflow;
- Integrating cloud-BPMS with devices in local network: using local printer in *SharePoint Online* workflow.

Approbation of the prototype proves that it implements all defined functions.

Results achieved in the thesis prove correctness of raised hypotheses. The goal of the thesis is achieved and all tasks are completed.

The thesis is accompanied by a CD that contains installation files and source code of *AgentWF* prototype. The prototype can be used:

- to integrate cloud-BPMS with on-premises systems and devices to address requirements of specific organization;
- to validate if proposed approach and architecture of MA system can be used in other scenarios;

- as a sample implementation of proposed approach and architecture of MA system, that will help to implement similar solutions for other platforms or by using different technologies.

Possible directions of future research are the following:

- Analyze possible improvements in proposed integration solution, to make it applicable in scenarios, when it is required to ensure real time synchronous integration or minimal possible request processing time;
- Extend proposed architecture of MA system to implement MASIF standard;
- Analyze possibilities to use proposed MA package format in other MA systems and scenarios.

BIBLIOGRAPHY

1. Add-ins and Extensibility. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/bb384200.aspx (cited: 01.08.2012).
2. Appian Announces Strong Growth in Q1 2011. Appian. Available from: http://www.appian.com/bpm-company/news/press/appian-announces-strong-growth-in-q1-2011.jsp (cited: 01.08.2012).
3. Bell T., Critical Capabilities for Composite Content Applications: Case Management, Gartner Research, 2010.
4. Bolstorff, P., Rosenbaum, R., Supply chain excellence: a handbook for dramatic improvement using the SCOR model, 2nd edn., Amacom, New York, 2007.
5. Bovet, D., Martha, J., Value nets: breaking the supply chain to unlock hidden profits. Wiley, New York, 2000.
6. Buchwald, H., Fleischmann, A., Seese, D., Stary, C., S-BPM One: Setting the Stage for Subject-Oriented Business Process Management, Springer-Verlag, 2010.
7. Chang, J. F., Business Process Management Systems, Strategy and Implementation. Auerbach Publications, 2005.
8. Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rockham, G., Impact of service orientation at the business level. IBM System Journal 44(1):81–107, 2005.
9. Chess, D., Harrison, C., Kershenbaum, A., Mobile Agents: Are They a Good Idea? IBM Research Division, 1995.
10. Chrissis, M. B., Konrad, M., Shrum, S., CMMI second edition: guidelines for process integration and product improvement. Person Education, Addison-Wesley, New York, 2007.
11. Deitert, E., McCoy, D. W., The Anatomy of a Business Rule Management System, Gartner Research, 2007.
12. Delias, P., Doulamis, A., Matsatsinis, N., What agents can do in workflow management systems, Artif Intell Rev, 35:155–189, 2011.
13. Dixon, J., Jones, T., Hype Cycle for Business Process Management, Gartner Research, 2011.
14. eID karte. VAS Latvijas Valsts radio un televīzijas centrs. Available from: https://www.eparaksts.lv/lv/eid-lietotajiem/kas-ir-eid/ (cited: 01.08.2012).
15. Enterprise Information Systems: Concepts, Methodologies, Tools and Applications. Idea Group Inc (IGI), ISBN 978-1-61692-852-0, 2010.
16. Erol, S., Granitzer, M., Happ, S., Jantunen, S., Jennings, B., Johannesson, P., Koschmider, A., Nurcan, S., Rossi, D., Schmidt, R., Combining BPM and social software: contradiction or chance?. J. Softw. Maint. Evol.: Res. Pract., 22: 449–476, 2010.
17. Extensible Application Markup Language (XAML). Microsoft. Available from: http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=19600 (cited: 01.08.2012).
18. Extensible Markup Language (XML). W3C. Available from: http://www.w3.org/XML/ (cited: 01.08.2012).

19. Farmer, W. M., Guttman, J. D., Swarup, V., Security for Mobile Agents: Issues and Requirements, The MITRE Corporation, 1996.
20. Fuggetta, A., Picco, G. P., Vigna, G., Understanding code mobility, IEEE Transactions on Software Engineering, 1998
21. Gartner Survey Shows Spending on Business Process Management to Grow Significantly in 2011, Gartner. Available from: http://www.gartner.com/it/page.jsp?id=1740414 (cited: 01.08.2012).
22. Gassman, B., Business Activity Monitoring Architecture Evolution, Gartner Research, 2008.
23. Gassman, B., Business Activity Monitoring Helps Execute a Pattern-Based Strategy, Gartner Research, 2009.
24. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3, 119–153, 1995.
25. Greenberg, M., Byington, J., Harper, D., Mobile Agents and Security, IEEE Communications Magazine, 1998.
26. Grundspenkis, J., Kirikova, M., Impact of the intelligent agent paradigm on knowledge management. Intelligent knowledge-based systems: Business and technology in the new millennium (C.T. Leondes, Ed.), Vol. 1. Boston, Dordrecht, London: Kluwer Academic Publishers, 164-206, 2005.
27. Grundspeņķis, J., Mislēvičs, A., Intelligent Agents for Business Process Management Systems. Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology, Malgorzata Pankowska (Karol Adamiecki University of Economics in Katowice, Poland), 2010, 97-131.
28. Grundspenkis, J., Pozdnyakov, D., An overview of the agent based systems for the business process management. Proceedings of the international conference on computer systems and technologies (CompSysTech'06), June 15-16, 2006, Veliko Tarnovo, Bulgaria, II.13-1 - II.13-6, 2006.
29. Gupta, P., Six Sigma business scorecard, 2nd edn. McGraw-Hill, New York, NY, 2006.
30. Gupta, R., Kansal, G., A Survey on Comparative Study of Mobile Agent Platforms, International Journal of Engineering Science and Technology, Vol. 3 No. 3, pp. 1943-1948, 2011.
31. Hai, H., Sakoda, S., SaaS and Integration Best Practices, Fujitsu Scientific Technology Journal, Vol. 45, No. 3, 257-264, 2009.
32. Hammer, M., Champy, J., Reengineering the corporation: a manifesto for business revolution. Harper Business, New York, NY, 1993.
33. Hammer, M., Deep change: how operational innovation can transform your company. Harv Bus Rev 82(4):84–96, 2004.
34. Hammer, M., What is Business Process Management? In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg, 2010.
35. Harmon, P., Business process change. A guide for business managers and BPM and Six Sigma professionals, 2nd edn. Morghan Kaufman, San Francisco, CA, 2007.
36. Harmon, P., The Scope and Evolution of Business Process Management. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg, 2010.
37. Harrison-Broninski, K., Dealing with human-driven processes. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 2. Springer, Heidelberg, 2010.
38. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF. Available from: http://tools.ietf.org/html/rfc5280 (cited: 01.08.2012).
39. Internet X.509 Public Key Infrastructure: Certification Path Building. IETF. Available from: http://tools.ietf.org/html/rfc4158 (cited: 01.08.2012).
40. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London, 1996.
41. Jennings, N. R. An agent-based approach for building complex software systems. Communications of the ACM, 44(4), 35-41, 2001.
42. Jennings, N. R., et al., Autonomous agents for business process management. Applied Artificial Intelligence, 14, 145-189, 2000.
43. Jennings, N. R., Norman, T. J., Faratin, P., ADEPT: An agent-based approach to business process management. ACM SIGMOD Record, 27, 32-39, 1998.
44. Jennings, N. R., Sycara, K., Wooldridge, M., A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1(1):7, 7-38, 1998.

45. JSON. Available from: http://json.org/ (cited: 01.08.2012).
46. Kaplan, R., S., Norton, D., P., Strategy maps. Harvard Business School Press, Boston, MA, 2004.
47. Khan, R., N., Business process management: a practical guide. Meghan-Kiffer, Tampa, FL, 2004.
48. Kotz, D., Gray, R., Mobile agents and the future of the Internet. ACM Operating Systems Review, 33(3), 7-13, 1999.
49. Kotz, D., Gray, R., Rus, D., Future directions for mobile agent research. IEEE Distributed Systems Online, 3(8), 1-6, 2002.
50. Lange, D., & Oshima, M., Seven good reasons for mobile agents. Communications of the ACM, 42(3), 88-89, 1999.
51. Lawrence, P., Workflow Handbook 1997, Workflow Management Coalition. John Wiley and Sons, New York, 1997.
52. Leymann, F. et al., Business process management standards. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg, 2010.
53. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice-Hall PTR, Upper Saddle River, 1999.
54. Liu, C., Chang, J., Yang, A., Analysis on Development Tendency of Business Process Management, ICICA 2011, Part II, CCIS 244, 629–636, 2011.
55. Liu, F., Guo, W., Zhao, Z. Q., Chou, W., SaaS Integration for Software Cloud, IEEE, 2010.
56. Liu, F., Wang, G., Chou, W., Li, L., TARGET: Two-way Web Service Router Gateway, Proc. IEEE International Conference on Web Services, 2006.
57. Ma, C., Caldera, A., Petridis, M., Bacon, L., Windall, G., Integration of BPM Systems. In: Process Management, Book edited by: Mária Pomffyová, ISBN 978-953-307-085-8, INTECH, Croatia, pp. 197-222, 2010.
58. Metasonic. Available from: http://www.metasonic.de/ (cited: 01.08.2012).
59. Milojicic, D., Breugst, M., Busse, I., Campbell, J., et al., MASIF: The OMG Mobile Agent System Interoperability Facility, Personal Technologies, vol. 2, Nr. 2, pp. 117-129, 1998.
60. Mislēvičs, A., An Introduction to Mobile Agent-Supported Business Process Management. Proceedings of The 4th International Conference "Information Society and Modern Business", Ventspils, Latvia, May 14-16, 2009, pp. 49-57, 2009.
61. Mislevics, A., Grundspenkis, J., Integrating Workflow-Based Mobile Agents with Cloud Business Process Management Systems, International Journal on New Computer Architectures and Their Applications (IJNCAA) vol. 2(4), The Society of Digital Information and Wireless Communications, ISSN: 2220-9085, pp. 511-530, 2012.
62. Mislēvičs, A., Grundspeņķis, J., Mobile Agents for Integrating Cloud-Based Business Processes with On-Premises Systems and Devices, Frontiers in Artificial Intelligence and Applications, vol.249: Databases and Information Systems VII, IOS Press, ISBN 978-1-61499-161-8, pp. 191-203, 2013.
63. Mislēvičs, A., Grundspeņķis, J., Workflow Based Approach for Designing and Executing Mobile Agents, Proceedings of The Second International Conference on Digital Information Processing and Communications (ICDIPC12), ISBN 978-1-4673-1105-2, pp. 97-102, 2012.
64. Mitrovic, N., Arribalzaga, U. A., Mobile Agent security using Proxy-agents and Trusted domains. Second International Workshop on Security of Mobile Agents, 2002.
65. Murch, R., Johnson, T., Intelligent software agents. New Jersey: Prentice Hall PTR, 1999.
66. Nwana, H., Software agents: An overview. Knowledge Engineering Review, 11(3), 1-40, 1996. Available from: http://agents.umbc.edu/introduction/ao/ (cited: 01.08.2012).
67. O'Brien, P. D., Wiegand, M. E., Agent based process management: Applying intelligent agents to workflow. The Knowledge Engineering Review, 13(2), 161-174, 1998.
68. O'Reilly, C., A., Tushman, M., L., The ambidextrous organization. Harv Bus Rev 82(4):74–82, 2004.
69. Office 365. Microsoft. Available from: http://www.microsoft.com/lv-lv/office365/online-software.aspx (cited: 01.08.2012).
70. Olding, E., Best Practices and Pitfalls for Business Process Competency Center Success, Gartner Research, 2010.
71. Olding, E., Rosser, B., Laying the Groundwork for Your BPM Initiative, Gartner Research, 2008.
72. Olding, E., Starting Up the Business Process Competency Center, Gartner Research, 2007.
73. OMG, Business process maturity model (BPMM), Version 1.0, Standard document, 2008. Available from: http://www.omg.org/spec/BPMM/1.0/PDF (cited: 01.08.2012).
74. OMG, Business Process Model and Notation (BPMN), Version 2.0, Standard document, 2010. Available from: http://www.omg.org/spec/BPMN/2.0/PDF (cited: 01.08.2012).

75. Ottinger, J., Software as a Service Integration via Mule. MuleSoft. Available from: http://mule.mulesource.org/ (cited: 01.08.2012).
76. Overeinder, B. J., de Groot, D. R. A., Wijngaards, N. J. E., Brazier, F. M. T., Generative Mobile Agent Migration in Heterogeneous Environments, Scalable Computing: Practice and Experience, Vol 7, No 4, 2006.
77. Pang, G., Implementation of an agent-based business process (Diploma thesis). Institut für Informatik der Universität Zürich, 2000.
78. Paulk, M., C. et al., The capability maturity model: guidelines for improving the software process. Addison-Wesley, Boston, MA, 1995.
79. Pohle, G., Korsten, P., Ramamurthy, S., Component business models: making specialization real. Whitepaper issued by IBM institute for business value. IBM Global Services, New York, 2005.
80. Poluha, R., G., Application of the SCOR model in supply chain management. Cambria Press, Amherst, NY, 2007.
81. Rama, J., Bishop, J., Towards a mobile agent framework for Nomad using .NET, Proceedings of SAICSIT, pp. 111–113, 2004.
82. Reflection in the .NET Framework. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/f7ykdhsy(v=vs.100).aspx (cited: 01.08.2012).
83. Ries, E., The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Corwn Business, ISBN: 978-0-307-88789-4, 2011.
84. Rosemann, M., de Bruin, T., Power, B., BPM maturity. In: Jeston J, Nelis J (eds) Business process management. Practical guidelines to successful implementations. Elsevier, NY, pp. 299–315, 2006.
85. Rosemann, M., vom Brocke, J., The Six Core Elements of Business Process Management. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg, 2010.
86. Russell, S., Norvig, P., Artificial intelligence. A modern approach (3rd ed). New Jersey: Prentice Hall, 2009.
87. Ruth, P., Rhee, J., Xu, D., Kennell, R., Goasguen., S., Autonomic live adaptation of virtual computational environments in a multi-domain infrastructure. In ICAC ’06: Proceedings of the 2006 IEEE International Conference on Autonomic Computing, Washington, DC, USA, 2006.
88. Scheer, A.-W., Klueckmann, J., BPM 3.0, BPM 2009, LNCS 5701, 15–27, 2009.
89. Schmidt, R., Nurcan, S., BPM and Social Software, Lecture Notes in Business Information Processing, 1, vol. 17, Business Process Management Workshops, part 9, pp. 649-658, 2009.
90. Schoeman, M. A., Architectural guidelines for Mobile Agent Systems. Technical Report: TR-UNISA-2003-02, School of Computing, University of South Africa, 2003.
91. Schoeman, M., Cloete, E., Architectural Components for the Efficient Design of Mobile Agent Systems, Proceedings of SAICSIT 2003, pp. 48–58, 2003.
92. Serialization. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/7ay27kt9(v=vs.100).aspx (cited: 01.08.2012).
93. Silverstein, D., DeCarlo, N., Slocum, M., INsourcing innovation. How to achieve competitive excellence using TRIZ. Auerbach Publications, New York, NY, 2005.
94. Sinur, J., A Vision of Advanced Business Process Simulation and Optimization, Gartner Research, 2010.
95. Sinur, J., Hill, J. B., Magic Quadrant for Business Process Management Suites, Gartner Research, 2010.
96. Smith, H., Fingar, P., Business Process Management The Third Wave. Meghan-Kiffer Press, ISBN: 0-929652-33-9, 2003.
97. Smith, R. F., Business process management and the balanced scorecard. Using processes as strategic drivers. Wiley, New York, NY, 2007.
98. SQL Server Compact. Microsoft. Available from: http://msdn.microsoft.com/en-us/data/ff687142.aspx (cited: 01.08.2012).
99. SQL Server Editions. Microsoft. Available from: http://www.microsoft.com/sqlserver/en/us/editions.aspx (cited: 01.08.2012).
100. Standard ECMA-376, Office Open XML File Formats, 1st edition, Part 2: Open Packaging Conventions, 2006. Available from: http://www.ecmainternational.org (cited: 01.08.2012).
101. Suh, Y.-H., Namgoong, H., Design of a Mobile Agent-Based Workflow Management System, S. Pierre and R. Glitho (Eds.): MATA 2001, LNCS 2164, pp. 93-102, 2001.
102. Sun, W., Zhang, K., Chen, S. K., Zhang, X., Liang, H., Software as a Service: An Integration Perspective. Proceedings of the ICSOC 2007, LNCS 4749, 558–569, 2007.

103. Sundararaj, A., Dinda., P., Towards virtual networks for virtual machine grid computing. In VM'04: Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium, 2004.
104. Suri, N., Vitek, J., Mobile Agents, Computational Complexity, pp. 1880-1893, 2012.
105. System.IO.Packaging Namespace. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/system.io.packaging.aspx (cited: 01.08.2012).
106. Tildes Jumis. Tilde. Available from: http://www.tilde.lv/tildes-jumis (cited: 01.08.2012).
107. Trammel, K., Workflow without fear. Byte, 1996.
108. van der Aalst, W., van Hee, K., Workflow management. Models, methods and systems. MIT Press, Cambridge, MA, 2004.
109. van der Aalst, W.M.P., Making Work Flow: On the Application of Petri nets to Business Process Management. In: J. Esparza and C. Lakos, editors, Application and Theory of Petri Nets 2002, vol. 2360 of Lecture Notes in Computer Science, pp. 1–22. Springer-Verlag, Berlin, 2002.
110. van der Aalst, W.M.P., Process-Aware Information Systems: Lessons to Be Learned from Process Mining. In: K. Jensen and W. van der Aalst (Eds.): ToPNoC II, LNCS 5460, pp. 1–26. Springer-Verlag Berlin, 2009.
111. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M., Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg, 2003.
112. Wagner, G., Agent-oriented analysis and design of organizational information systems. Databases and information systems (J. Barzdins & A. Caplinskas, Eds.). Kluwer Academic Publishers, 111-124, 2011.
113. Weske, M., Business Process Management Concepts, Languages, Architectures. Springer Berlin Heidelberg, ISBN: 978-3-540-73521-2, 2007.
114. Windows Azure Service Bus. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/windowsazure/ee732537 (cited: 01.08.2012).
115. Windows Communication Foundation. Microsoft. Available from: http://msdn.microsoft.com/en-us/netframework/aa663324.aspx (cited: 01.08.2012).
116. Windows Presentation Foundation. Microsoft. Available from: http://windowsclient.net/wpf/default.aspx (cited: 01.08.2012).
117. Windows Workflow Foundation. Microsoft. Available from: http://msdn.microsoft.com/en-us/netframework/aa663328 (cited: 01.08.2012).
118. Wood, T., Shenoy, P., Gerber, A., Ramarkrishnan, K., Merwe, J., "The case for enterprise-ready virtual private clouds". Workshop on Hot Topics in Cloud Computing, HotCloud'09, 2009.
119. Wooldridge, M., An introduction to multi agent systems (2nd ed). Chichester, UK: John Wiley & Sons, 2009.
120. Wooldridge, M., Intelligent agents. The MIT Press, 1-51, 1999. Available from: http://www.csc.liv.ac.uk/~mjw/pubs/mas99.pdf (cited: 01.08.2012).
121. XAML Syntax In Detail. Microsoft. Available from: http://msdn.microsoft.com/en-us/library/ms788723.aspx (cited: 01.08.2012).
122. XML Encryption Syntax and Processing. W3C. Available from: http://www.w3.org/TR/xmlenc-core/ (cited: 01.08.2012).
123. Yuhong, Y., Zakaria, M., Weiming, S. Integration of Workflow and Agent Technology for Business Process Management. Proceedings of The Sixth International Conference on CSCW in Design, 2001.
124. Zachman, J. A., A framework for information systems architecture. IBM System Journal 26(3):276–292, 1987.
125. zur Muehlen, M.: Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems, Logos, Berlin, 2004.