Information Systems Integration

XML Schema for Configuration of Project Management Information Systems

Solvita Bērziša

Riga Technical University, Faculty of Computer Science and Information Technology, 1 Kalku, Riga, Latvia, berzisa@gmail.com

Abstract. Project management information systems support the complex project management process. This process is regulated by project management methodologies, standards and other requirements. To support this process is necessary that project management information system could be configured according the requirements of chosen methodology. To facilitate the configuration process project management requirements are described and defined in a standardized manner using XML. The objective of this paper is to elaborate structure of the aforementioned schema, which is referred as XML schema for Configuration of Project Management information systems (XCPM). Development of the XML schema is based on the comprehensive project management concept model. Each entity form project management concept model can be described using XCPM schema. This paper also surveys project management information system configuration approach and project management concept model. Project management information system configuration process is provided as example of schema usage.

Keywords: XCPM, project management information systems, PMIS configuration, XML schema

1 Introduction

Project management information systems (PMIS) are one of the key elements for project management (PM) and project success [1]. However, similarly as for other types of information systems, application of PMIS would be successful if it is implemented according to requirements of a particular project management situation or project management methodology. Methodologies that describe PM can be divided into two groups: general (e.g. PMBOK [2] and PRINCE2 [3]) and domain specific (e.g. RUP[4] and MSF[5]). Ideally, PMIS would adhere to all requirements of PM methodology used for a particular project. But in reality multiple projects are governed by different methodologies and regulatory requirements what makes PMIS configuration a complex task. Therefore, an approach for implementation and configuration of PMIS was developed [6]. It uses principles of model and templates driven configuration of packaged applications to reduce efforts of implementing PMIS. PM methodologies and other project requirements are specified according to a

standardized definition, which is subsequently used to configure a packaged project management application by means of package specific transformations. The standardized definition can be represented as a project management domain definition XML schema. This schema must be sufficiently comprehensive so that different PM methodologies can be described in terms of this definition.

An objective of this paper is to elaborate the aforementioned XML schema, which can be used for configuration of PMIS. This schema is referred as to XML for Configuration of Project Management information systems (XCPM). It is developed on the basis of the PM concept model (PMCM) that has been obtained after performing conceptual modelling of the project management domain [7]. The XCPM schema is established using information from the PMCM and by applying the concept model to XML schema transformation rules. Each entity of the PMCM model is represented in the XCPM schema. The paper describes the PMIS configuration approach, the PMCM, XCPM schema elaboration rules and structure. A PMIS configuration prototype is designed to illustrate XCPM schema usage.

The contribution of this research is development of the XCPM schema suitable for configuration of PMIS. The new XML schema is developed because existing PM schemas (PMXML [8] and Microsoft Project XML schema [9]) focus on operational data such as tasks, resource, and assignment rather than on specification of structural properties. The XCPM schema also includes descriptive information about PM and PM processes. Abstract elements are included in the schema what allows to describe specific configuration information and can be used for special data storage. The main distinctive feature of the proposed schema is joint representation of data, processes and knowledge.

The paper is structured as follows. Section 2 describes the PMIS configuration approach and reviews related approaches for configuration of enterprise systems. Section 3 gives overview of the PMCM and it generation process. Section 4 briefly describes structure of XCPM schema and more detail description of process and knowledge area definition. Prototype of PMIS configuration is described in Section 5. Section 6 concludes and discusses future work.

2 PMIS configuration

Development of XCPM is a part of ongoing research on development of the model and template driven approach for configuration of PMIS according to requirements of project management requirements and regulatory requirements [6].

2.1 Overview of approach

The PMIS configuration and implementation process is shown in Fig. 1. It starts with definition of requirements from the PM methodology or regulatory guidelines, PM process and other PM relevant information. These requirements are necessary to represent in a standardized manner. The PM requirements are transformed into the appropriate standardized representation and formalized specification of PM

requirements is obtained. The formalized specification of PM requirements can be automatically transformed to configure a PMIS by using PM application specific transformation scripts. The standardized representation enables using the same transformation scripts for different project management methodologies. During the transformation processes, the methodology-specific content also could be loaded into the PM system. That includes predefined data values, document templates, and process guides. These are retrieved from the PM knowledge database.

The standardized representation is developed using XCPM. Therefore, XCPM should be comprehensive enough to enable definition of different PM methodologies. The analysis of the PM domain shows that XCPM should resemble data, process and knowledge related aspects of PM [7].

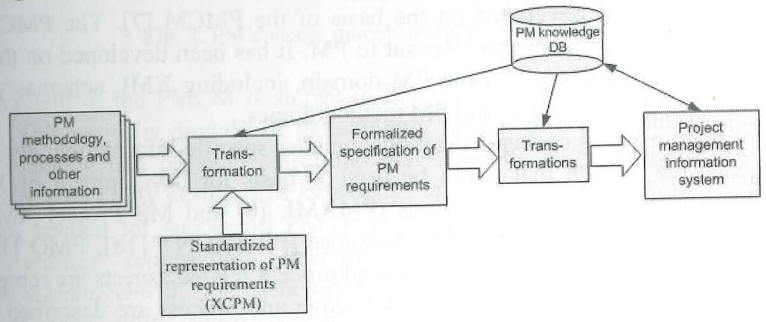

Fig. 1. An approach of configuration of PMIS

2.2 Enterprise and workflow system configuration

To our knowledge, there are no papers addressing configuration of PMIS according to requirements of specific PM methodologies, but there are different approaches for configuration of workflow systems and enterprise applications such as ERP systems. Both systems focus on the automation of business process and information sharing, but workflow systems are process-centric and ERP systems are data-centric [10]. PMIS is similar to workflow system, but it is concentrated to a specific domain – project management. The PM domain has a fairly well-defined scope what makes it possible to elaborate an all-inclusive domain definition.

Approaches of enterprise information systems and ERP implementation and configuration are also based on standardized configuration description that mostly is business process description with models. For example, model-driven implementation approach uses model driven architecture models transformation [11]. Object-process based approach uses ERP system model and enterprise requirement model alignment [12]. The model-driven configuration approach uses configurable event-driven process model [13]. The holistic approach uses three perspectives: management, business process analyst and technical analyst [14]. Each of these perspectives requires a model at different level of elaboration, for example, a business process analyst uses the configurable event-driven process model, but a technical analyst the workflow model [14].

A second opportunity of configuration description is XML which is widely used in workflow systems. For example, MS SharePoint workflows are stored in XML workflow markup file (XOML) [15]. This format is used by Windows Workflow Foundation. XML process definition language (XPDL) is a widely used format for process definition exchange between workflow systems [16]. The configurable workflow model is developed for YAWL (Yet Another Workflow Language) and XML document is used for configuration description [17].

3 Project Management Concept Model

The XCPM schema is developed on the basis of the PMCM [7]. The PMCM is a composition of different concepts relevant to PM. It has been developed on the basis of existing conceptualizations of the PM domain, including XML schemas used in PM applications, PM ontologies and PM methodologies.

Each of these existing conceptualizations includes specific information, but no one is comprehensive enough that it can be used as basis for development of XCPM. XML schemas used in PM applications (PMXML [8] and MS Project XML [9]) define only operational PM data. PM ontologies (PROMONT [18], PMO [19] and others) focus on PM knowledge while data and process related aspects are represented only at very high level of abstraction. Knowledge and process are described in PM methodologies (e.g. PMBOK [2], PRINCE2 [3], MSF [5], and RUP [4]) though this description is unstructured and, in the case of specific methodologies, represent just a single viewpoint of PM. For that reason, a separated concept model has been created for each of these sources.

The concept model integration has been performed to obtain the comprehensive concept model of the PM domain. Database models or XML schema integration principles and techniques have been used for integration of the concept models. The model integration consists of the following main tasks: pre-integration, model (schema) conforming, schema merging and schema restructuring [20]. The concept models have been created in pre-integration phase. During the conforming phase conflict detection and conflict resolution are performed. Three common types of the conflicts are naming, semantic, and structural conflicts [21]. Three main types of schema conflict resolution transformations are entity/attribute, entity/relationship, and attribute/subtype equivalence [20]. Related concepts are identified and a single federated concept model is produced in the merging phase. Four types of schema merging transformations are used, namely, entity merge, introduction of is-a relationships, addition of union, and addition of intersection [20]. During schema restructuring quality of the federated concept model is improved [20]. Four types of schema restructuring transformations are used – redundant attribute removal, optional attribute removal, generalization of attributes, and redundant relationship removal [20]. Schema conforming and schema merging are executed in an iterative manner in each stage adding one concept model. PMCM integration algorithm is shown in Fig. 2 [6].

After completing the integration process, the PMCM is obtained. The model defines all main entities and relationships present in the PM domain.

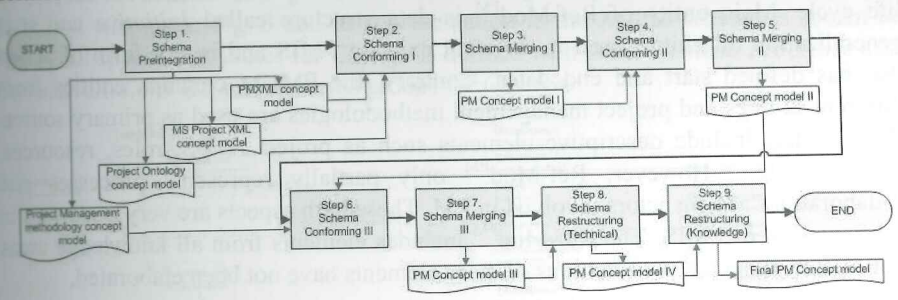

Fig. 2. PM concept models integration algorithm

A fragment of the PMCM is shown in Fig. 3. The central entity of the model is *Project*. This entity is directly or transitionary related to sets of entities describing project planning and project controlling (e.g. *IssueLog* and *RequestChanges*). The *KnowledgeArea* entity defines PM knowledge areas describing knowledge, skills and processes needed for successful PM. It represents a generic case, while there are several entities representing specific knowledge related entities such as *IntegrationManagement*, *IssueManagement* and *ChangeManagement*. *KnowledgeAreas* entity summarizes general *KnowledgeArea* and specific knowledge entities. KnowledgeAreas entity is part of *ProjectLifeCycle* entity that includes information about PM organization. The *Process* entity represents various PM processes.

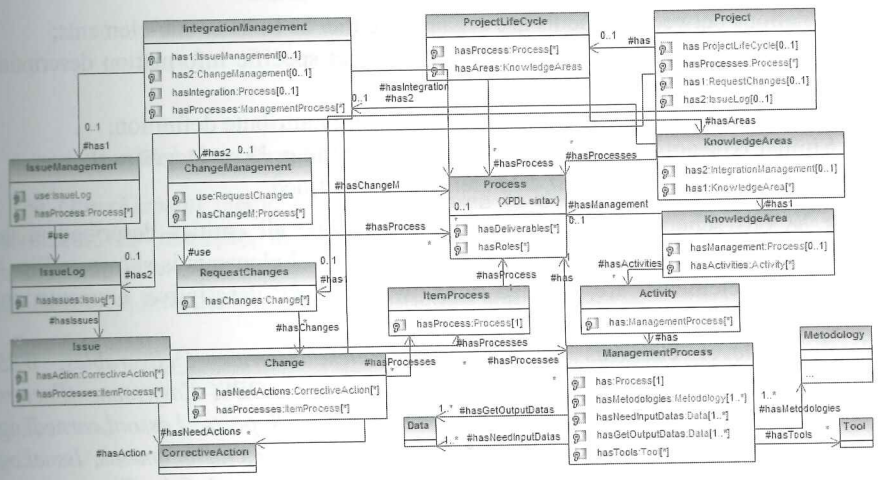

Fig. 3. Fragment of PMCM

The proposed conceptualization of the PM domain has been developed in parallel with development of reference model Ref-Mod^{PM} by Ahlemann [22]. The purpose of this model is to facilitate design of PM software, set-up of the surrounding

organization system and definition of PM software requirements. This model is based on a references project life-cycle model. PMCM allows to specify a custom project life cycle. Main entity of Ref-Mod^{PM} is data structure called *Initiative* and it is generalization of entities used in different existing PMIS and is any form of action that has defined start and end dates. Contrary, the PMCM contains entities from different sources and project management methodologies are used as primary source. Both models include descriptive elements such as project, WBS, roles, resources, calendars etc. However, Ref-Mod^{PM} only partially represents processes and collaborations among actors involved in PM. These both aspects are very important in configuration of PMIS. The Ref-Mod^{PM} includes elements from all knowledge areas of PMBOK though representations of these elements have not been elaborated.

4 Structure of XCPM

The forward engineering approach is used to develop XCPM from PMCM. That includes three development stages: development of conceptual schema, development of logical schema and development of physical schema [23]. In this case, PMCM is the conceptual scheme. The logical schema or XCPM is obtained using information in PMCM and transformation rules. In this case, the physical schema already includes certain data specifying particular project management methodology according to the structure provided by XCPM.

The following rules are used to design the XCPM schema from PMCM:

- XCPM schema main top-level elements are entities in PMCM, which are directly related to the *Project* entity;
- Relations between elements are organized either as links or sub-elements;
- Abstract elements are added to ensure project specific information description (e.g. *OtherElement*);
- Abstract elements are used in PM concept entity attribute definition;
- Similar type entities are merged in one element (e.g. *KnowledgeArea*);
- Both the configuration and data elements are defined.

The XCPM schema includes elements to describe all entities described in the PMCM. Overview of the XCPM schema main top-level elements is shown in Fig. 4. The schema elements are divided into two groups: descriptive elements and management elements.

Such top-level elements as *Calendars*, *Roles*, *Metrics*, *Deliverables*, *WBS*, *Limitations*, *Milestones*, *Activities*, *Schedule*, *Resources*, *OBS*, *EnvironmentFactors*, *RBS*, *RiskRegister*, *CBS*, *QualityRequirements*, *ProgressReports*, *LessonLearnedLog*, *PerformanceReviews*, *Baselines*, *Attributes*, *ProjectManagementDocuments*, *IssueLog*, *ChangesRequests*, *Assignments*, *CommuicationRequirements* and *OtherElements* have been included in the descriptive group. Sets of elements are subordinated to each of the top-level elements. Each set of elements describes or collects any information about an area of project or PM. Main descriptive top-level elements are described in Tab. 1.

ChangesRequests is one of descriptive elements that define changes, corresponding to corrective actions and processes. The change request definition in PMCM definition was given Fig. 3 and its representation in the XCPM schema is shown in Fig. 5. All specific attributes for *Change* is defined with *OtherElement*. *Processes* and *CorrectiveActions* are sub-elements groups for *Change*.

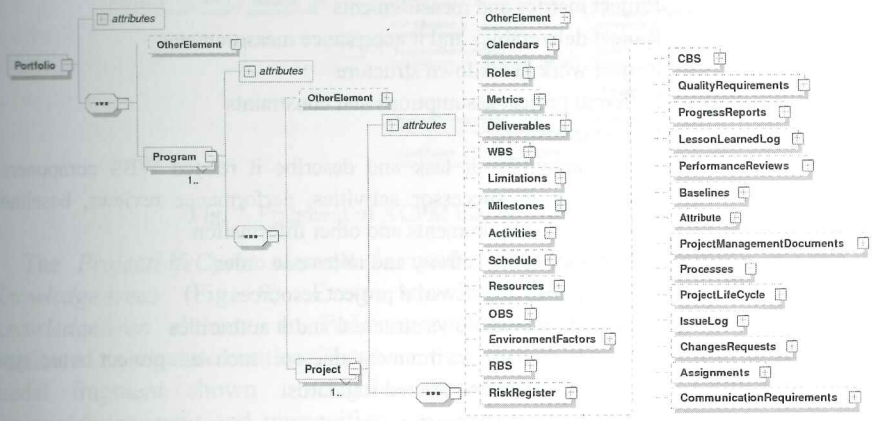

Fig. 4. Overview of XCPM schema main top-level elements

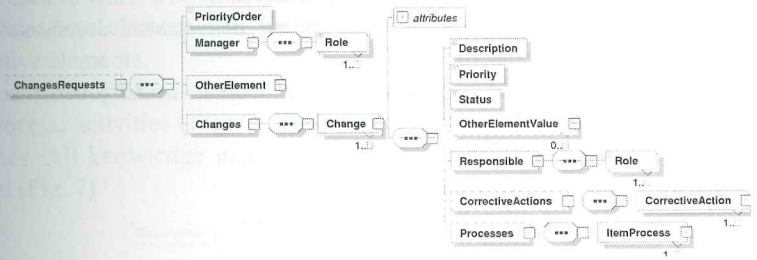

Fig. 5. Fragment of XCPM that describes *ChangesRequests*

The *OtherElement* element (Fig. 6) is used to define different specific project information, for example, in a software maintenance project it is used to define configuration items, test scenarios and code reviews. This element allows to define specific project information items, its attributes and values.

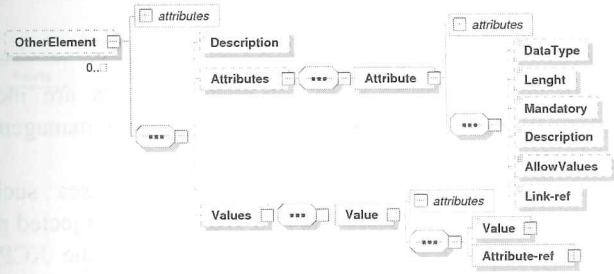

Fig. 6. Fragment of XCPM that describes *OtherElement*

Table 1. XCPM descriptive top-level elements

Element	Description
Calendars	All calendars in project such as project calendar, resource calendars, and contract calendars
Roles	Roles in project, it skills, works and related reports.
Metrics	Project metrics and measurements
Deliverables	Project deliverables and it acceptance measurements
WBS	Project work breakdown structure
Limitations	Different project assumptions and constraints
Milestones	Project milestones
Activities	Project activities or task and describe it related WBS components, predecessor or successor activities, performance reviews, baselines, assumptions, requirements and other information
Schedule	Information about activity and milestone order
Resources	Project human and material project resources
OBS	Organization breakdown structure and it authorities
EnvironmentFactors	Different project environment factors such as project area, type, organization type and related standards
RBS	Risk breakdown structure
RiskRegister	Project risks, it measurements, processes, corrective actions, responsible roles and other information
CBS	Project contracts, it tasks, sellers, deliverables, documentation and progress reports
QualityRequirements	Project quality requirements, it measurements, criteria, corrective actions, deliverables and standards
ProgressReport	Progress reports in project
LessonLearnedLog	Lessons learned information
PerformanceReviews	Earned values, variances and forecasts
Attributes	Different project attributes
ProjectManagement Documents	PM documents and plans
IssueLog	Project issues, corresponding corrective actions and processes
ChangesRequests	Project changes, corresponding corrective actions and processes
Assignment	Resource or role assignment to activity or milestone
Communication Requirements	Communication requirements, meetings, communication objects and other commutation information
OtherElement	Define different specific project information

ProjectLifeCycle and *Processes* main top-level elements are included in the management group. Elements in this group describe project management activities and processes and uses elements from descriptive group.

The *Processes* element describes dynamic PM processes such as change management, status reporting processes and risk/change/issue injected processes. The process itself is defined as a workflow. This workflow in the XCPM schema is included using exiting XML description of workflows – XPDL [16]. *Any* element is

used in *Process* definition to enable including fragments of XPDL document. Definition of the *Process* entity in PMCM was shown in Fig. 3 and its definition in the XCPM schema is shown in Fig. 7.

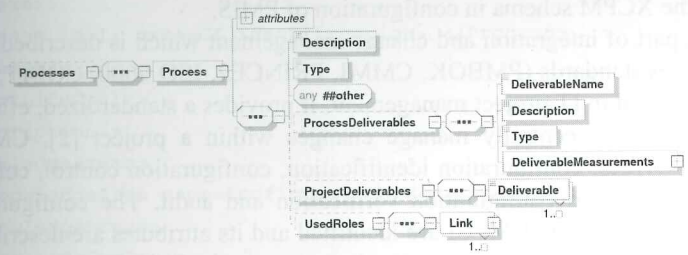

Fig. 7. Fragment of XCPM that describes *Processes*

The *ProjectLifeCycle* element is used to define main project process and *KnowledgeAreas* (Fig. 8). Every knowledge area can be described using *KnowledgeArea* element. In PM concept model classic knowledge areas are distributed as separate entities. There are four knowledge area entities in PM concept model fragment shown in Fig.3: *IntegrationManagement*, *IssueManagement*, *ChangeManagement* and unspecified *KnowledgeArea* entity. All these entities are related with *Process* and *ManagementProcess* entities. These entities differ in that each is used to work with different PM area entities. Consequently, it is sufficient that *KnowledgeArea* element will be provided with attributes which indicate involved descriptive elements.

The *KnowledgeArea* element (Fig. 8) describes a methodology used, knowledge area process, activities (Fig. 9), management processes, input/output data, tools and processes. All knowledge management processes are described using the *Processes* element (Fig. 7)

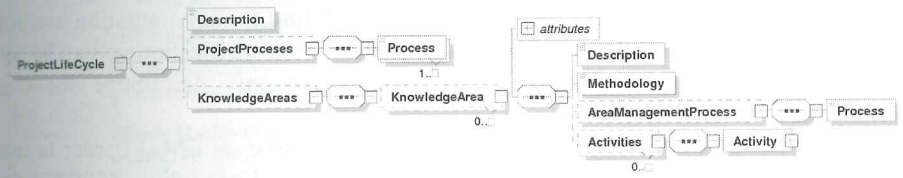

Fig. 8. Fragment of XCPM that describes *ProjectLifeCycle*

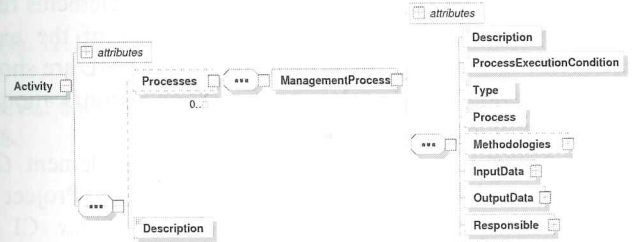

Fig. 9. Fragment of XCPM that describes *Activity*

5 Prototype of PMIS configuration

Configuration management (CM) is used as a representative example to demonstrate the use of the XCPM schema in configuration of PMIS.

CM is a part of integration and change management which is described in various methodologies/standards (PMBOK, CMMI, PRINCE2, ISO 9000, COBIT, ITIL) and is frequently used in IT project management. It provides a standardized, effective, and efficient process to centrally manage changes within a project [2]. CM includes following activities: configuration identification, configuration control, configuration status accounting and configuration verification and audit. The configuration item (CI) is central element in CM. CI are identified and its attributes are described during the configuration identification activity. In our example CI item creation is an event that injects process, which requires identifying the related documents and sent for approval to CI owner and project manager. This process is shown in Fig. 10.

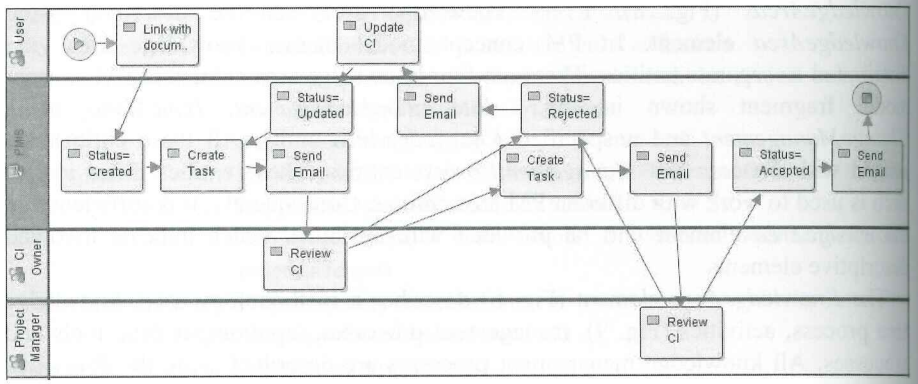

Fig. 10. CM process after a CI has been created

All changes related with CI item are identified during the configuration control activity. Configuration status accounting and configuration verification and audit not involve any management process. These actions are only descriptive for this prototype.

The above description provides an informal representation of CM. In order to use this information in configuration of PMIS, it is structured according XCPM. An XML document specifying CM in the standardized manner is obtained as the result of this transformation (in this case, the document contains only elements relevant to CM and other elements are omitted because of limited scope of the example). This document is referred as to CMXML. Main elements of CMXML are shown in Fig. 11 and more detail description of CI, roles, configuration identification process and configuration management area are in Appendix.

CI is a specific element in PM so it is defined using element *OtherElement*. Attributes for each CI are defined using the *Attribute* elements. Project roles are one of classic elements and it can be defined using element *Roles*. CI identification process is described using *Processes* element group. Configuration management and activities are defined using *KnowledgeArea* element.


```
<Project name='Example_SB_2010'> …
 <OtherElement name=Configuration Items; type=List>…</OtherElement>
 <Roles>…</Roles>
 <Processes>
   <Process id=1; name=CI_identification>…</Process>
 </Processes>
 <ProjectLifeCycle>
   <Description/>
   <KnowledgeAreas>
     <KnowledgeArea name=Configuration Management>…</KnowledgeArea>
   </KnowledgeAreas>
 </ProjectLifeCycle> …
</Project>
```

Fig. 11. Main elements of CMXML document

The last configuration step is loading the CMXML document into PMIS. The loading is performed using software specific transformation scripts. In the example, the Microsoft Project Server is configured according the defined CM process. CI list and workflow setting of Microsoft Project Server is shown in Fig. 12. Mapping between CMXML document (Fig. 11) and PMIS configuration (Fig. 12) is shown in Tab. 2.

With *OtherElement* defined Configuration Items is added as list in PMIS (Number 1 in Fig.12) and it attributes is defined as field or columns for CI (Number 2 – 7 in Fig. 12). *CI_identification* process after CI creation is identified by configuration identification action described with *KnowledgeArea* element. CI processes in PMIS can be viewed in CI workflow settings (Number 8 in Fig. 12).

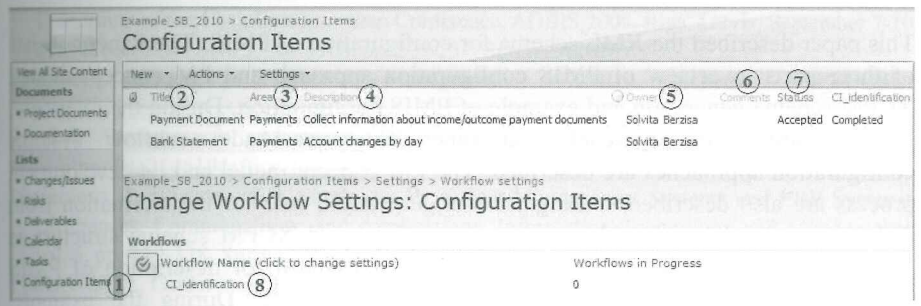

Fig. 12. Example PMIS configuration according CMXML document

This CMXML document or it parts can be reused in other projects and also loaded in other PMIS.

Table 2. PMIS elements mapping with CMXML document

Number (Fig.12)	PMIS element	XPath to CMXML document (Fig.11 and Appendix)
1	List *Configuration Items*	Project/OtherElement[@name=Configuration_Items]
2	Field *Title*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Title]
3	Field *Area*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Area]
4	Field *Description*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Description]
5	Field *Owner*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Owner]
6	Field *Comments*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Comments]
7	Field *Status*	Project/ OtherElement[@name=Configuration_Items]/ Attributes/ Attribute[@name=Status]
8	Workflow for configuration item	Project/ ProjectLifeCycle/ KnowledgeAreas/ KnowledgeArea [@name= Configuration Management]/ Actions/ Action[@name= Configuration_identification]/ Processes/ ManagementProcess[@id=023]
	CI_identification process	Project/ Processes/ Process[@name=CI_identification]

6 Conclusion and Future Work

This paper described the XML schema for configuration of PMIS. The Paper consists of three parts: overview of PMIS configuration approach and PM concept model, XCPM scheme generation and example of PMIS configuration. During the overview, PMIS configuration approach and other enterprise and workflow systems configuration approaches are described. The PM concept model and its development process are also described. During schema generation, rule for transformation from PM concept model to XCPM schema are defined and XCPM schema structure is generated. The result is XCPM schema that can be used for description of PMIS configuration include PM data, process and knowledge. During the example, prototype of PMIS configuration approach is created that shows usage of XCPM schema. The paper and future research on configuration of PMIS also contributes to the area of automated configuration of packaged applications. The PM domain is particularly well-suited for automated configuration because it is relatively well-defined domain.

Definition of XCPM schema structure is a part of the PMIS configuration approach. Definition of automatic transformation scripts and validation of XCPM schema are the next steps. XCPM schema elements can be changed or supplemented with technical details while transformation process will be defined. Schema validation

will be performed on PM methodologies and real life projects. The main steps of future research are approbation of the XCPM schema and elaboration of it transformation to different PMIS.

Acknowledgments. The author would like to acknowledge Dr.sc.ing. Jānis Grabis for his scientific guidance and support.

References

1. Raymond, L., Bergeron, F.: Project management information systems: An empirical study of their impact on project managers and project success. International Journal of Project Management, vol. 26, pp.213-220, Elsevier Ltd (2008)
2. Project Management Institute: Guide to the Project Management Body of Knowledge (PMBOK) Fourth edition. PMI Publication, Newtown Square (2008)
3. Hedemen, B., Heemst, G.V.van, Fredriksz, H: Project Management Based on PRINCE2 – PRINCE2 Edition 2005. Van Haren Publishing, Zaltbommel (2007)
4. Kruchten, P.: The Rational Unified Process: An introduction Third edition. Addison-Wesley, Boston (2004)
5. Turner, M.S.V.: Microsoft Solutions Framework Essentials: Building Successful Technology Solution. Microsoft Press, Washington (2006)
6. Bērziņa, S., Grabis, J.: An approach for Implementation of Project Management Information Systems. In: Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, G. (eds.) Information System Development: Toward a Service Provision Society, pp. 423-431. Springer (2009)
7. Bērziņa, S.: Towards an XML Scheme for Configuration of Project Management Information Systems: Conceptual Modelling. Grundspenkis, J., Kirikova, M., Monolopoulos, Y., Novickis, L. (eds.) Advances in Databases and Information Systems Associated Workshops and Doctoral Consortium of the 13-th East-European Conference, ADBIS 2009, Riga, Latvia, September 7-10, 2009 Proceedings LNCS, vol. 5968, pp. 229-237. Springer (2010)
8. Cover Pages, Project Management XML Schema (PMXML), http://xml.coverpages.org/projectManageSchema.html
9. XML Structure for Microsoft Office Project 2003, http://msdn.microsoft.com/en-us/library/aa210619(office.11).aspx
10. Cardoso, J., Bostrom, R.P, Sheth, A.: Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications. Information Technology and Management, vol. 5, pp. 319-338, Springer Netherlands (2004)
11. Gugerdil, P., Gaillard, G.: Model-Driven ERP Implementation. 8-th International Conference on Enterprise Information Systems (ICEIS'2006), Workshop on Model-Driven Enterprise Information Systems (MDEIS), Paphos, Cyprus, May 23 – 27, INSTICC Press (2006)
12. Soffer, P., Golany, B., Dori, D., Aligning an ERP system with enterprise requirements: An object-process based approach, Computers in Industry, vol. 56, pp. 639–662, Elsevier B.V. (2005)
13. Recker, J., Mendling, J., Aalst, W.M.P van der, Rosemann, M.: Model-driven enterprise system configuration. Advanced Information System Engineering, LNCS, vol. 4001, pp. 369-383, Springer Berlin/Heidelberg (2006)
14. Dreiling, A., Rosemann, M., Aalst, W.M.P. van der, Sadiq, W.: From Conceptual process models to running systems: A holistic approach for configuration of enterprise system processes. Decision Support Systems, vol. 45, pp. 189-2007, Elsevier (2007)

15. Holliday, J., Alexander, J., Julian, J., Robillard, E., Schwartz, B., Ranlett, M., Attis, J.D., Buenz, A., Rizzo, T.: Professional SharePoint 2007 Development, Wiley Publishing, Indianapolis (2007)
16. Workflow Management Coalition, XPDL Support and Resources, http://www.wfmc.org/xpdl.html
17. Gottschalk, F., Aalst, W.M.P van der., Jansen-Vullers, M.H., La Rosa, M.: Configurable Workflow Models. International Journal of Cooperative Information Systems, vol. 17, pp. 177-221, World Science Publishing (2008)
18. Abels, S., Ahlemann, F., Hahn, A., Hausmann, K., Strickmann, J.: PROMONT - A Project Management Ontology as a Reference for Virtual Project Organizations. On the Move to Meaningful Internet Systems 2006, OTM 2006 Workshops, LNCS, vol. 4277, pp. 813-823. Springer, Berlin/Heidelberg (2006)
19. Ruíz-Bertol, F.J., Dolado, J.: A Domain Ontology for Project Management. In: Berki, E., Nummenma,a J., Sunley, I., Ross, M., Staples G. (eds.) Software Quality Management XV: Software Quality in the Knowledge Society, pp. 317-326. British Computer Society (2007)
20. McBrien, P., Poulovassilis, A.: A formalisation of semantic schema integration. Information Systems, vol.23, No.5, pp. 307-334. Elserver Science Ltd, Great Britain (1998)
21. Aleksandraviciene, A., Butleris R.: A Comparative Review of Approaches for Database Schema Integration. Advances in Information Systems Development, pp. 111-122, Springer US (2007)
22. Ahlemann, F.: Towards a conceptual reference model for project management information systems. International Journal of Project Management, No.27, pp. 19-30., Elseviesr (2009)
23. Fong. J., Cheung S.K.: Translating relational schema into XML schema definition with data semantic preservation and XSD graph. Information and Software Technology, vol. 47, pp. 437-462, Elsevier (2005)

Appendix: Fragments of CMXML document

Configuration Items are described using OtherElement element:

```
<OtherElement name=Configuration Items; type=List>
   <Description> Collect information about configuration items
   </Description>
   <Attributes>
     <Attribute name=Title>
       <DataType>Text</DataType> <Length>100</Length>
       <Mandatory>1</Mandatory> <Description>CI title</Description>
     </Attribute>
     <Attribute name=Area> … </Attribute>
     <Attribute name=Description> … </Attribute>
     <Attribute name=Owner> … </Attribute>
     <Attribute name=Comments> … </Attribute>
     <Attribute name=Status> … </Attribute>
   </Attributes>
   <Values/>
 </OtherElement>
```


Project roles are described using *Roles* element:

```
<Roles>
  <Role name=Project Manager>
    <Description>Ensure project team work</Description>
    <Skills> <Skill>Risk and issue management<Skill>
             <Skill>Understand project constraints<Skill> …
    </Skills>
  </Role>
  <Role name=Programmer> … </Role>
  <Role name=System analyst> … </Role>
</Roles>
```

Configuration identification process is described using *Process* element:

```
<Processes>
  <Process id=1; name=CI_identification>
    <Description> Configuration Item approve </Description>
    <WorkflowProcesses>
     <WorkflowProcess Id="CI-identif_wp1" Name="CI identification">
       …
       <Participants>
        <Participant Id="CI-identif_wp1_par2"  Name="Project Manager">
          <ParticipantType Type="ROLE"/>
        </Participant>
        …
       </Participants>
       <Activities>
        <Activity Id="CI-identif_wp1_act1" Name="Status=Created">
          <Performer>CI-identif_wp1_par1</Performer>
          …
          <ExtendedAttributes>
           <ExtendedAttribute Name="Configuration_Item.Status"
                 Value="Created"/>
          </ExtendedAttributes>
         </Activity>
         …
       </Activities>
       <Transitions>
        <Transition From="CI-identif_wp1_act1"
          Id="CI-identif_wp1_tra2" To="CI-identif_wp1_act2"/>
        …
       </Transitions>
     </WorkflowProcess>
    </WorkflowProcesses>
  </Process>
  <ProcessDeliverables/>
```

```
  <ProjectDeliverables/>
  <UsedRoles>
    <Link>
      <Participant-ref>Project Manager<Participant-ref>
      <Role-ref>Project Manager<Role-ref>
      …
    </Link>
    …
  <UsedRoles>
</Processes>
```

Configuration Management knowledge area is described using *KnowledgeArea* element:

```
<KnowledgeArea name=Configuration Management>
  <Description/>
  <Methodology/>
  <Activities>
    <Activity name=Configuration identification>
     <Description/>
     <Processes>
      <ManagementProcess id=023>
       <Name>Configuration accept</Name>
       <Description/> …
       <ProcessExecutionCondition> Create Configuration Item
       </ProcessExecutionCondition>
       <Process ref=1>
       <InputData> Configuration Item </InputData>
       <OutputData> Configuration Item </OutputData>
      </ManagementProcess>
     </Processes>
    </Activity>
    <Activity name= Configuration control> … </Activity>
    <Activity name= Configuration status accounting> … </Activity>
    <Activity name= Configuration verification and audit>…</Activity>
  </Activities>
</KnowledgeArea>
```


Web Service Selection: Beyond Quality of Service

Mārtiņš Bonders, Jānis Kampars, Jānis Grabis
Institute of Information Technology, Riga Technical University,
1 Kalku Street, Riga, LV – 1658, Latvia
martins@iti.rtu.lv

Abstract. Selection of appropriate web services is an important step in development of composite applications. Quality of Service (QoS) data characterizing nonfunctional properties of candidate web services are usually used in web service selection. However, functional characteristics, which are difficult to measure, are equally important. Quantitative evaluation of functional characteristics is possible in the case of development of composite decision-making applications. This paper elaborates a web service selection method, which accounts for both functional and nonfunctional characteristics of candidate web services. The functional characteristics are evaluated according to decision-making results yielded by the composite application. A multi-objective linear programming model is used as the web service selection method. Application of the proposed web service selection method is demonstrated by developing a composite application for a vehicle routing problem, where web services are required to provide spatial data necessary for the decision-making algorithm. Experimental results show that accounting for functional characteristics substantially affects web service selection results.

Keywords: Web services, QoS, vehicle routing

1 Introduction

Composite applications use external services to gain access to vast data and processing resources. Web services relying on such open technologies as XML, SOAP, WSDL, REST and others are often used as building blocks for composite applications. That allows developing standards-based, scalable and efficient applications. However, properties of composite applications directly depend upon characteristics of external services used and environmental factors, which in the case of public networks, exhibit high degree of variability. Therefore, selection of appropriate and reliable services is of major importance. Multiple methods have been elaborate for selection of such services from the set of candidate services providing similar functionality [1,2].

These methods often use Quality-of-Service (QoS) measurements to evaluate services. These measurements including reliability, security, trust and execution cost are mainly concerned with nonfunctional characteristics of services. However, service consumers are equally concerned about both functional and nonfunctional characteristics of services and there have been attempts to expand the QoS concept in

the case of web service selection by defining it as: ''The degree to which a system, component or process meets customer or user needs or expectations" [1]. This definition includes evaluation of both functional and nonfunctional requirements. Unfortunately, formal evaluation of functional characteristics in the framework of web service selection is more difficult than evaluation of nonfunctional characteristics. A functional quality of service approach [2] uses similarity measures to identify interoperable web services. A QoS–aware service selection algorithm includes functional requirements in the model though these are represented only by a binary variable indicating either complete satisfaction or complete dissatisfaction of the requirement [3]. Generally, evaluation of functional characteristics either involves expert judgment or has limited resolution.

In order to address this issue, functional characteristics of web services also should be evaluated quantitatively and considered jointly with nonfunctional QoS characteristics. The objective of this paper is to elaborate a quantitative approach for selecting web services according to both functional and non-functional requirements and to demonstrate application of this approach in development of composite decision-making applications.

Application of the proposed approach is demonstrated using a decision-making application, which is used to determine optimal routing of multiple postal vehicles. Two main modules of this application are the routing algorithm and the data retrieval module. The data retrieval module uses external services to obtain data about distances between customers and travel time. Several web services are evaluated to determine the service giving the best results according to both functional and nonfunctional characteristics.

The remaining part of the paper is organized as follows. Section 2 describes web services characteristics most frequently used in web service selection. The approach for selecting web services according to both functional and nonfunctional requirements is elaborated in Section 3. The sample application and experimental results are discussed in sections 4 and 5, respectively. Section 6 concludes.

2 Web service selection criteria and methods

Web service selection has attracted large interest from academicians and practitioners alike. A number of web service selection methods have been elaborated and several typical QoS measurements used in web service selection can be identified by analyzing these methods. Table 1 surveys selected web service selection methods. All these methods are multi-criteria selection methods because web service selection is an essentially multi-criteria problem. Analytical Hierarchical Process (AHP) is the most frequently method used. Usually this method is used together with other methods. Different methods from the artificial intelligence domain such as fuzzy algorithms and artificial neural networks are also frequently considered to account for factors, which are difficult to express analytically.

Table 1. Overview of Web service selection and approach

Publication	Approach
[4]	QoS-aware composite service binding approach based on Genetic Algorithms
[5]	The Artificial Neural Network
[6]	Decentralized trust and reputation mechanisms for peer-to-peer based Web service systems
[7]	GFS (Goodness-Fit Selection algorithm) based on QoS prediction mechanism in dynamic environments
[8]	Fuzzy-based UDDI with QoS support
[9]	QoS Consensus Moderation Approach (QCMA) in order to perform QoS consensus
[10]	A query-by-example representing Web service descriptions and queries as vectors
[11]	Analytic hierarchy process (AHP) approach, Stateless and stateful session beans are applied in the CSPMP matrices
[12]	QoS meta-model as the basis for the QoS and AHP modeling
[13]	An QoS-aware services selection model based on LINear programming techniques for Multidimensional Analysis of Preference
[14]	Select services by considering two different contexts: single QoS-based service discovery and QoS-based optimization of service composition. Based on QoS measurement metrics, this study proposes multiple criteria decision making and integer programming approaches to select the optimal service
[15]	Convergent population diversity handling genetic algorithm (CoDiGA)
[16]	Analytic Hierarchy Process (AHP) and the Brown–Gibson (BG) methods
[17]	Graph transformation rules to describe the semantics of Web services. These rules provide precise semantic specifications needed for an automated service discovery in a visual and intuitive way

The literature reviews suggests that there are two main categories of attributes used in web services selection: QoS properties and business properties category [18,19]. The QoS properties category may be divided into two sub categories: execution and security properties.

1. Execution includes the performance parameters which characterize the interaction with the web service. The main execution nonfunctional characteristics are:

1.1. Response Time– time elapsed from the submission of a request to the time the response is received.

2. 2. Accessibility – represents the degree that a Web service is able to serve a request.

1.3. Compliance – represents the extent to which a WSDL document follows WSDL specification

1.4. Successability – represents the number of request messages that have been responded.

1.5. Availability – represents the percentage of time that a service is operating.

2. Security is related to the ability of a given Web service to provide suitable security mechanisms by considering the following three parameters. The main security nonfunctional characteristics are:

2.1. Encryption – the ability of a Web service to support the encryption of messages.

2.2. Authentication – the capacity of a Web service to offer suitable mechanisms dealing with the identification of the invoking party and allow operation invocation.

2.3. Access control – whether the Web service provides access control facilities to restrict the invocation of operation and the access to information to authorized parties.

The business properties category may be divided into two sub categories: strategic properties and environmental properties. Environmental properties include two nonfunctional features namely location and temporal properties.

1. The mains strategic non-functional features are:

1.1. Cost – represents money that a consumer of a Web service must pay in order to use the Web service.

1.2. Reputation – measures the reputation of Web services based on user feedback.

1.3. Organization arrangement – includes preferences and history (ongoing partnerships).

1.4. Payment method – represents payment methods accepted by a web service, i.e. transfer ban, credit card etc.

1.5. Monitoring– required for a number of purposes, including performance tuning, status checking, debugging and troubleshooting.

2. Environmental properties include two features namely location and temporal properties.

Table 2 lists nonfunctional QoS characteristics considered in selected papers. Response time, accessibility and availability are the most universally used characteristics in the QoS properties category. Cost is the most frequently used business related characteristic.

Table 2. Overview of QoS characteristics used in web service selection.

Source	QoS characteristics												
	Execution					Security			Strategic				
	Response/execution time	Accessibility	Compliance	Successability	Availability	Encryption	Authentication	Access control	Cost	Reputation	Organization arrangement	Payment method	Monitoring
[4]	x	x			x				x				
[20]	x	x		x	x				x		x	x	
[8]	x	x		x	x								
[9]	x								x				
[18]	x	x	x	x	x	x	x	x	x	x	x	x	x
[17]	x		x			x	x	x	x				
[21]	x	x			x		x	x					

3 Functional evaluation of Web services

Nonfunctional QoS characteristics usually used in service selection have been identified in the previous section. However, users are equally concerned with functional characteristics, which are unique to a particular application, and representation of functional characteristics is more difficult in formal service selection models. Two approaches can be used to evaluate web service conformance to functional requirements: 1) expert evaluation; and 2) quantitative performance measures. Existing service selection methods mainly use expert evaluations. Finding quantitative performance measures characterizing functional requirements is difficult. One approach could be using business key performance indicators though these indicators give only indirect measurements, and they are affected by different side-effects. Decision-making applications form one class of applications for which quantitative performance measurements can be determined because functionally these applications are required to produce specific performance indicators. For instance, if composite application is used in vehicle routing then the main functional requirement is finding a route between two locations. In this case, alternative services can be evaluated by comparing travel times returned by the services.

The proposed web service selection approach evaluates candidate services according to both functional and nonfunctional requirements for decision-making applications, where functional requirements can be evaluated by direct quantitative performance measures. The overall service selection approach is shown in Figure 1.

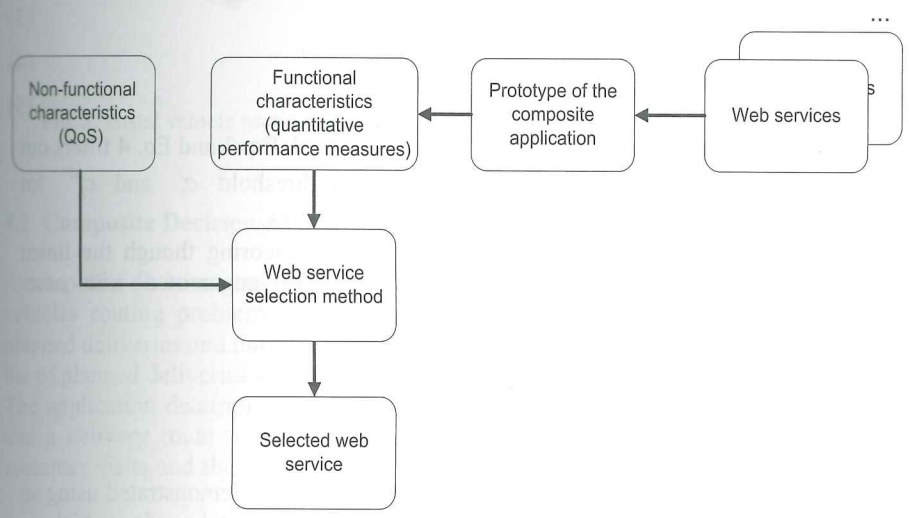

Fig.1. Web service selection according to functional and nonfunctional characteristics

It is assumed that requirements for a composite application relevant to web service selection are given by N functional requirements referenced by index i and M nonfunctional requirements referenced by index j. There are L alternative web services $\mathbf{W}=(w_1,...,w_L)$ satisfying the requirements to some degree. QoS data $\mathbf{G}_l=(g_{l1},...,g_{lM})$ characterize how well the lth service performs according to the jth nonfunctional requirement and quantitative performance measures $\mathbf{H}_l=(h_{l1},...,h_{lN})$ characterize how well the lth service performs according to the ith functional requirement. $\mathbf{G}_l$ can be determined according to data provided by the service provider or using simulation studies. In order to evaluate $\mathbf{H}_l$, a prototype composite application is built. A variant of the prototype S_l is developed for each candidate web service. Values of $\mathbf{H}_l$ are determined by executing the corresponding prototype S_l. A web service selection method takes $\mathbf{G}_l$ and $\mathbf{H}_l$ as input data and finds the best service using a multi-criteria selection methods.

As indicated in Section 2, different selection methods can be used to choose the most appropriate services given their quantitative characteristics. In this paper, a multi-objective linear programming model is used as the selection method. The objective function maximizes the service score Z:

$$Z=\sum_{l=1}^{L}\left(\sum_{i=1}^{N}u_i^F h_{li}+\sum_{j=1}^{M}u_j^{NF}g_{lj}\right)X_l\rightarrow\max, \quad (1)$$

where X_l indicates whether the lth service is selected and u_i^F and u_i^{NF} are importance weights for functional and nonfunctional requirements, respectively.

The optimization is performed subject to the following constraints:

$$\sum_{l=1}^{L}X_l=1, \quad (2)$$

$$h_i^F\le c_i^F,\forall i, \quad (3)$$

$$h_j^{NF}\le c_j^{NF},\forall j, \quad (4)$$

$$X_i\in\{0,1\},\forall i. \quad (5)$$

The first constraint implies that only one service is selected. Eq. 3 and Eq. 4 filters out services having worse characteristics than specified threshold c_i^F and c_i^{NF} for functional and nonfunctional requirements, respectively.

This model is equivalent to a simple weighted-linear scoring though the linear programming formulation is more flexible and can be augmented with other parameters and constraints.

4 Application example

Application of the proposed web service selection approach is demonstrated using an example of development of composite application for routing postal service vehicles.

4.1 Problem Description

In the case of routing postal service vehicles, the postal service receives a daily list with planned parcel deliveries and needs to schedule its fleet of vehicles to fulfill these requests. Each planned delivery is defined by a customer address and a delivery time window. The delivery time window specifies the earliest and latest possible delivery time. All vehicles depart from the single depot and also need to return to the depot within the normal operating hours. There are about a thousand delivery requests per day, and there are about 20 delivery vehicles. The postal service aims to minimize delivery costs and to minimize delivery time what directly influences a number of vehicles needed to serve all customers. Figure 2 shows the graphical representation of the vehicles routing problem. It can be observed that a separate route is created for each vehicle, and each customer is assigned to exactly one route.

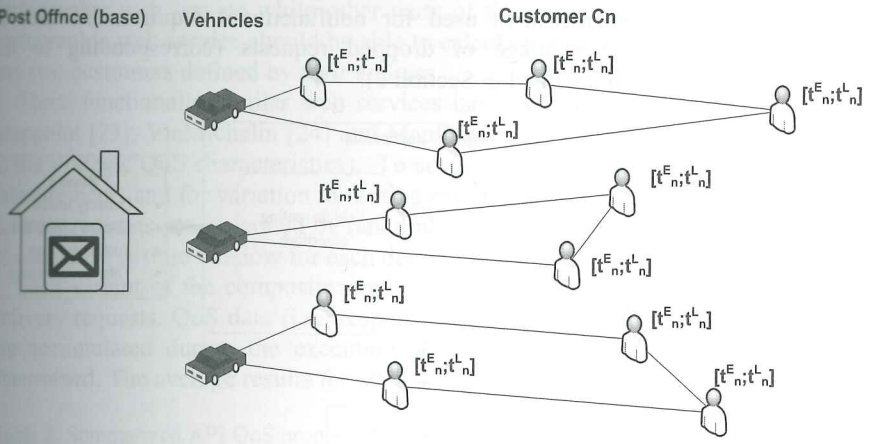

Fig. 2. Postal vehicle routing example, where $\left[t_n^E;t_n^L\right]$ represents the delivery window

4.2 Composite Decision-Making Application

A composite decision-making application (Figure 3) is developed in order to solve the vehicles routing problem described above. This application receives the list with planned deliveries and number of vehicles in the postal service fleet as input data. The list of planned deliveries shows addresses of customers and the delivery time window. The application determines the number of vehicles necessary to serve all customers and a delivery route for each vehicle. This delivery route shows the sequence of customer visits and the scheduled arrival time at each customer. The application also

yields the total travel time and travel distance for all vehicles. Thus, the main functional requirements for the composite vehicle routing application are determining number of necessary vehicles, creating the delivery route for each vehicle and fining the total travel time and distance.

Two main components of the composite application are the vehicle routing module and data gathering module. The vehicle routing module is responsible for finding delivery routes using the input data provided. The data gathering module is responsible for retrieving data necessary for finding the delivery routes. These data include input data and additional data retrieved using web services. The necessary additional data are distances between customers and travel times from one customer to another. They can be retrieved from cartographical web services such as MapPoint and ViaMichelin. There are a large number of such cartographical web services. In order to obtain the most efficient delivery routes, the most appropriate web service should be selected. This web services is selected using the service selection approach described in Section 3. The quantitative performance measures used for functional requirements are: 1) number of vehicles; 2) route length in kilometers; and 3) driving time in hours. The QoS measures used for nonfunctional requirements are: 1) response time; and 2) percentage of dropped requests (corresponding to the successability characteristic defined in Section 2).

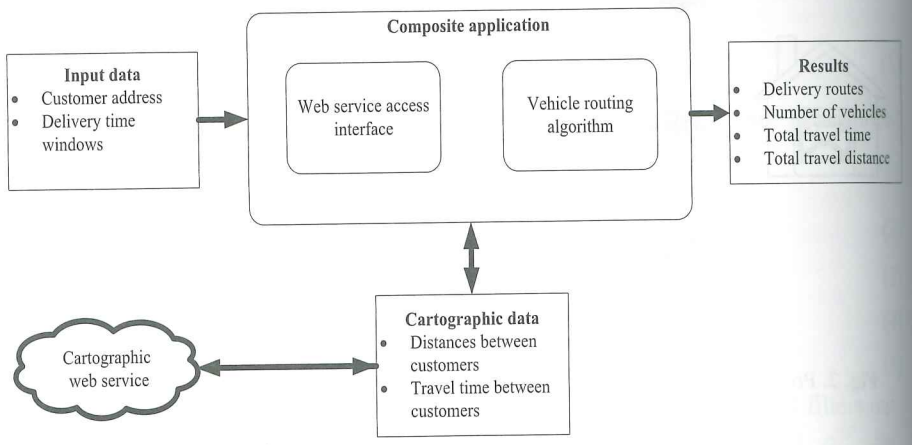

Fig. 3. The composite vehicle routing application.

4.3 Routing Algorithm

The routing algorithm is one of key modules of the composite application. The heuristics vehicles routing algorithm with time windows developed by Soloman [22] is this application. The main difference between this algorithm and routing algorithms used in cartographical web services is that this algorithm creates routes for multiple vehicles simultaneously and accounts for restrictions imposed by time windows.

The heuristic algorithm is used because it is computationally more efficient than optimal algorithms. It initializes a route for the first vehicle by assigning a customer to the route. A customer nearest to the first customer in the route is identified using

the distance measurements retrieved from web services. If time window constraints can be satisfied then this customer is added to the route. If time window constraints are violated then other nearest customers are checked. If none of customers can be added to the current route then a new route is initialized and the next vehicle is scheduled for deliveries. This process is continued until all customers have been included in one of the routes.

5 Experimental Results

Experimental studies are conducted to demonstrate service selection for the composite vehicle routing application. A set of 100 addresses representing customer locations was generated. The postal office depot address is chosen manually, and customer addresses are randomly generated within 150 km around the depot. Three candidate variants of the composite application are implemented. Each variant uses a different cartographic web service while other parts of the application remain the same. The cartographic web service should be able to calculate travel time and distance between any two customers defined by their address.

Three functionally similar web services have been identified, namely, Microsoft Mappoint [23], ViaMichelin [24] and MapQuest [25] (preselection is done according to the strategic QoS characteristics). To account for uncertainty in evaluation of QoS measurements and for variation in routing results due to input data used, ten lists with delivery requests are generated by randomly drawing 50 addresses from the initial set of addresses. A time window for each delivery request is also generated.

Each variant of the composite routing applications is executed for each list with delivery requests. QoS data (i.e., response time and percentage of dropped requests) are accumulated during the execution and quantitative performance measures are determined. The average results for all test lists are summarized in Table 3.

Table 3. Summarized API QoS property values

	Average response time (s)	Percentage of dropped requests	Average number of vehicles needed	Average travel distance (km)	Average travel time (h)
Mappoint	4.59	0	11.00	2704	59.52
ViaMichelin	4.53	0.41	11.50	2903	64.82
MapQuest	4.23	0.11	10.70	2808	60.59

It can be seen that the average response time of all services is similar, however, if the number of requests is large, the total time of data retrieval can vary by several hours. During data retrieval all three services were stable and the percentage of dropped requests is low. Evaluation of services according to the functional requirements exhibits more substantial differences.

The Mappoint web service yields the smallest travel distance and travel time while MapQuest yields routes requiring the smallest number of vehicles. These results are

obtained for 50 customers; it has been observed that differences between obtained results increase with increasing number of customers. It should be noted that all services yield similar travel speed and travel time differences are not merely due one service overestimating the travel speed.

To determine the best web service for the given vehicles routing problem, Eq.1 is solved using the normalized values of $\mathbf{H}_l$ and $\mathbf{G}_l$ and weights for nonfunctional requirements are set $u_i^{NF} = \{0.2, 0.2\}$ for functional requirements $u_i^F = \{0.3, 0.5, 0.3\}$. As the results, the Mappoint web service has the highest normalized service score equal to 1.19.If only nonfunctional requirements are considered (all u_i^F are set to zero), then MapQuest is the best web service. Mappoint is also the best service if only functional requirement are used in service selection.

6 Conclusion

An approach for selecting web services has been elaborated. The main distinctive features of this approach is accounting for both functional and nonfunctional characteristics of candidate web services. That has been achieved by using quantitative performance measures to quantify the functional characteristics. However, evaluation of these measures is challenging, and, in this paper, it has been restricted to class of composite decision-making applications. The obtained experimental results for the vehicle routing problem show that web service selection substantially depends upon accounting for functional characteristics. During implementation of the sample composite application, it has been observed that strategic QoS characteristics have major impact on pre-selection of web services.

One shortcoming of the proposed approach is necessity to implement a variant of the composite application for each web service to be evaluated. However, replacing a web service component for the properly design application is not very effort consuming tasks though there could be situations when developing multiple variants is prohibitive. A training data set is also necessary to perform functional evaluation of candidate web services. In the case of vehicle routing application, a field validation of the obtained results is also necessary because it is possible that the presumably best web service gives inaccurate estimates of travel distances and travel times.

Obviously, selection results depend upon importance weights assign to requirements. Procedures like AHP can be used to determine values of these weights. The relatively simple multi-objective linear programming model used in the paper can be easily extended to represent composite applications requiring multiple web services for fulfilling different functional requirements. This model also can be replaced with a more advanced service selection model if necessary because other web service selection methods surveyed in Section 2 can use quantitative performance measures characterizing functional requirements as input data.

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

References

1. Huang Angus F.M., Ci-Wei Lan, Stephen J.H. Yang, An optimal QoS-based Web service selection scheme, Information Sciences: an International Journal archiveVolume 179 , Issue 19, Pages: 3309-3322, 2009.
2. Buhwan J., Hyunbo C., Choonghyun L., On the functional quality of service (FQoS) to discover and compose interoperable web servicesExpert Systems with Applications 36, 5411–5418, 2009.
3. Dimitrios T., Ioanna R., Efstathios S., QoS-aware service evaluation and selection, European Journal of Operational Research Volume 191, Issue 3, Pages 1101-1112, 2008.
4. Massimiliano Gerardo C., Di P., Raffaele E., Maria L. V., A framework for QoS-aware binding and re-binding of composite web services, Journal of Systems and Software Volume 81, Issue 10, October 2008, Pages 1754-1769 Selected papers from the 30th Annual International Computer Software and Applications Conference (COMPSAC), Chicago, September 7–21, 2006.
5. Haibin C., Xiaohui H., Qiying C., A novel intelligent service selection algorithm and application for ubiquitous web services environment, Expert Systems with Applications Volume 36, Issue 2, Part 1, Pages 2200-2212, 2009.
6. Yao W., Vassileva J., A Review on Trust and Reputation for Web Service Selection, *Department of Computer Science University of Saskatchewan,* Vol. 0, 25, 2007.
7. Li M, Huai J., HuiPengGuo, An Adaptive Web Services Selection Method Based on the QoS Prediction, *School of Computer Science and Engineering, Beihang University, Beijing, China,* IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology – Workshops, 2009.
8. Wang Hei-Chia, Chang-Shing Lee, Tsung-Hsien Ho, Combining subjective and objective QoS factors for personalized web service selection, Expert Systems with Applications 32 571–584, 2007.
9. Wei-Li Lin, Chi-Chun Lo, Kuo-Ming Chao, Muhammad Younas, Consumer-centric QoS-aware selection of web services, Journal of Computer and System Sciences 74, 211–231, 2008.
10. Crasso M., Alejandro Z., Marcelo C., Easy web service discovery: A query-by-example approach, Science of Computer Programming 71, 144–164, 2008.
11. Jiachen H., Daizhong Su, Integration of Web Services technology with business models within the total product design process for supplier selection, Computers in Industry 57, 797–808, 2006.
12. Chen Wu, E., Chang, Intelligent Web Services Selection based on AHP and Wiki, IEEE/WIC/ACM International Conference on Web Intelligence, 2007.
13. Angus F.M. Huang, Ci-Wei Lan, Stephen J.H. Yang, An optimal QoS-based Web service selection scheme, A.F.M. Huang et al. / Information Sciences179, 3309–3322, 2009.
14. Menasc´eDaniel A., R., Honglei, G., Hassan, QoS management in service-oriented architectures, Performance Evaluation 64, 646–663, 2007.
15. Yue Ma, Zang C., Quick convergence of genetic algorithm for QoS-driven web service selection, Computer Networks 52, 1093–1104, 2008.
16. Sun Y., He S., Jack Y. Leu, Syndicating Web Services: A QoS and user-driven approach, Decision Support Systems 43, 243–255, 2007.

17. Hausmann J. H., Heckel R., Lohmann M., Towards Automatic Selection of Web Services Using Graph Transformation Rules, Faculty of Computer Science, Electrical Engineering and Mathematics University of Paderborn Warburger Str. 100 33098 Paderborn, 2003.
18. Badr Y., Abraham A., Biennier F., Grosan C., Enhancing Web Service Selection by User Preferences of Non-Functional Features, Proceedings of the 2008 4th International Conference on Next Generation Web Services Practices table of contents, Pages: 60-65, 2008.
19. Cardoso J., Miller J., A. Sheth, J. Arnold, Modeling quality-of-service for workflows and Web service processes, Web Semantics: Science, Services and Agents on the World Wide Web Volume 1, Issue 3, Pages 281-308, 2004
20. XuanV., T., Tsuji H., Masuda R., A new QoS ontology and its QoS-based ranking algorithm for Web services, Simulation Modelling Practice and Theory Volume 17, Issue 8, , Pages 1378-1398, Dependable Service-Orientated Computing Systems, 2009.
21. Diamadopoulou V., Makris C., Panagis Y., Sakkopoulos E., Techniques to support Web Service selection and consumption with QoS characteristics, Journal of Network and Computer Applications 31, 108–130, 2008.
22. Soloman M. Marius, Algorithms for the vehicle routing and scheduling problems with time window constraints, Northeastern University, Boston, Massachusetts, December 1985.
23. Mappoint Web Service, http://msdn.microsoft.com/en-us/library/dd877971.aspx
24. ViaMichelin Web Service, http://dev.viamichelin.com/wswebsite/gbr/jsp/vmdn/VMDN-WebServices.jsp
25. MapQuest Web Service, http://developer.mapquest.com/web/products/directions-ws

BPEL Engine and Complex Event Processing Network Integration

Normunds Blumbergs[1], Maksims Kravcevs [1],

[1] University of Latvia, Riga, Latvia
{NormundsBlumbergs@gmail.com, Maksims.Kravcevs@lu.lv}

Abstract. The paper describes principles, architecture and design of a general solution for business process management systems integration with complex event processing (CEP) engines.
As CEP query standards are still in early development we define a general abstract CEP Query Model and corresponding distributed infrastructure: CEP network to abstract from specific CEP engines. We propose integration between BPEL Engines and CEP Network using intermediary CEP Service. Key part of the solution is to represent CEP network queries as WSRF resources. Events from CEP network are received using WS-Notification. This approach allows to use standard BPEL and does not require any changes or extensions in BPEL Engine and designing tool.
We compare the proposed approach with some existing alternatives. We present a prototype which implements key components of the proposed solution provides integration between an open source CEP Engine Esper and BPEL Apache ODE engine.

Keywords: CEP, BPEL, BPMS, WSRF, WS-Notification

1 Introduction

Real Time Business Process Management (RTBPM) is a new area which evolves in Business Process Management (BPM). These are optimized business processes that use data from external systems and react in near to real time on events generated by these systems. Opposed to usual Business Activity Monitoring where events are from business process activities, here the sources of events can be rather arbitrary systems. Additional sources of events and capability of response in real time improve operational agility of the enterprise and interoperability in business processes.

Gartner Research states that event management will be an area, which will greatly influence BPM solution competitiveness in the nearest future [9]. Similarly Forrester Research mentions complex event processing as a new functionality required from ESB products, which will determine ESB competitiveness [10].

[1] Research is supported by grant ESS 2009/82 from University of Latvia and European Social Fund contract no 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044

The BPM area has gone quite far in standardisation and alignment with SOA approach. The BPM systems usually support open standards, with BPEL language most widely used for definition of executable business processes. BPEL supports invocation of web services as process steps.

At the same time Complex Events Processing (CEP) has been developed quite independently. The objective of CEP is to discover complex inferred events by analyzing and correlating other events in real time [1]. Inferred events usually have business meaning as compared to source events, which are low operational events; therefore they can be used in business processes. Opposed to BPM where common open standards have been generally adopted, CEP engine query language standards are in stage of early development and CEP engine vendors compete with different sets of language features.

Currently the existing integration solutions between CEP and BPM are specific for concrete products and not agile enough, therefore it looks quite promising to provide new approaches. In this paper we consider what would be sufficiently general and efficient way of integration between different BPEL engines and CEP Engines. The key requirement is that it should allow utilizing strong capabilities of both tools and (following SOA principles) ensure loose coupling between them and be based on open standard. We propose such a solution and implement a prototype as a proof of concept.

In order to achieve that our solution is independent from concrete CEP engine event processing model, we need to define an abstraction for CEP engines and their event processing models. We offer a notion of CEP Network, which represents an infrastructure that can connect several CEP engines, their clients and sources of events. This does not introduce any limitations as in the simplest case CEP network corresponds to a single CEP Engine. We also define a general CEP Query model as an event processing model in CEP Network. As a basis for CEP network and CEP Query definition we take common possibilities of following widely used CEP Engines: Coral8[2], Esper[3], Tibco[4] and StreamBase[5].

Given CEP network and BPEL Engine we decide on integration pattern with an intermediary component CEP Service that will provide native integration method towards both BPEL process and CEP Network. The advantage of this approach is that it does not require changes or extensions in the BPEL and CEP Network models. The key decision is to represent CEP Queries in CEP Service as WSRF resources, with event data as resource properties and event represented as change of the resource. WSRF specification provides notification mechanism on resource changes via WS-Notification. That makes it easy for BPEL process to subscribe and receive events from CEP network.

We implement a proof of concept prototype, which contains all key components of the proposed solution. The proof of concept is built using Open Source solutions; it includes implementation of CEP Network over Esper CEP Engine and CEP service which ensures access to CEP enquiries as WSRF resources. Apache ODE BPEL

[2] http://www.coral8.com/
[3] http://esper.codehaus.org/
[4] http://www.tibco.com/software/complex-event-processing/default.jsp
[5] www.streambase.com/

Engine is used for test integration scenario. The solution is published as Open Source Project at Google Code[6].

The paper is structured as follows. In the second section we provide some general information on web service standards and CEP concepts we use. The third section describes CEP Network and CEP Query model abstraction we introduce. The fourth section gives a general overview of proposed integration solution. The fifth section provides more details on the solution components and their functionality. The sixth section describes some alternatives and existing solutions that can be used for similar purpose. Final section describes implemented prototype and used technologies in it.

2 Background

2.1 Web services, WSDL and WS-BPEL standards

SOA propagates approach to IT system composition from loosely coupled and distributed components with clear contract – services. The most common technology to implement services on service-oriented principles is to use web service standards. Web service standards define a standard mechanism for exposure and consumption of data and application logic over Internet protocols such as HTTP using xml and are de facto standards for SOA implementations.

WSDL [12] is an xml language for description of web service. It contains abstract and concrete definitions of web service. Abstract definition describes web service interface names, operations that can be performed on them, messages used to perform these operations and data types used in messages. Concrete definition describes the location of the web service implementation and the way it can be accessed. WSDL for webservice contains complete information enough for client application to technically interact with the web service.

WS-BPEL [11] is an xml web service orchestration language that defines business processes interacting with web services. WS-BPEL is most widely supported process definition language by modern BPM systems. WS-BPEL contains elements for definition of complex process flow and it utilises invocation of web services as process activities. It is possible to extend WS-BPEL to support additional web service standards. Sure support for the standard should be built in into BPEL Engine where this BPEL process is deployed and also implementations of web services that are called also support these standards. An example of one of such extensions could be WS-Atomic Transaction support. Then in BPEL we can specify that invocation of several web services within the process should be in one distributed transaction. Most BPEL Engines will support this, and thus will ensure passing of necessary technical information and additional technical messages to web services called from BPEL process.

[6] http://code.google.com/p/ebpm/

2.2 WS-notification

WS-Notification [13] is a web service standard that provides message notification pattern. It allows web services to subscribe to receive notifications from other web services. The standard consists of several specifications. We use WS-Topics which allows to describe subscription topics and WS-BaseNotification that defines direct interaction between Notification Consumers and Notification Producers. A Notification Consumer is an endpoint, designated to receive notifications produced by a Notification Producer. In order to subscribe to these notifications Notification Consumer should send to a Notification Producer a standard subscription request and implement a standard operation which Notification Producer will call back to send notifications.

2.3 WSRF, WS-Resource

WSRF (*Web Services Resource Framework*) [14] is a set of specifications that extend web service definition (which is stateless by default) with state. WS Resource is defined as a logical entity extending web service, uniquely addressable, having some attributes and a lifecycle. WS-Resource refers to Resource Properties Document, which represents WS-Resource attributes. Clients can manipulate resources: create, modify and examine their properties and destroy them.

WSRF is typically used when it is necessary to model and provide some structured information dynamically, see for example [16] where BPEL process model and state is presented as WS resource with properties according to BPEL metamodel.

Important for us is the additional possibility defined to support WS-Notification for resource acting as a Notification Producer. There are separate WS topics defined for particular attribute and for resource as a whole. Thus according to WS-Notification standard client may subscribe to the necessary topics defined for the resource. Once resource properties are changed the notification for relevant topics is sent to subscribed clients.

2.4 CEP languages & CEP Engines

CEP language is a language used by CEP engine to manipulate complex events. As an example of CEP languages, we can consider Coral8 CCL [2] (*continuous computation language*), Esper EPL [3] (*event processing language*). Currently common standard development is still in early stage despite that there are several researches and attempts in this direction, e.g. [5]. There are common possibilities that languages provide, for an overview see [4].

Some of the basic notions are: Event Type –some data structure, content of the event, usually flat; Event Streams – grouping of events of the same type representing event flow; Event Patterns – certain relations between events of interest.

CEP language allows to define event patterns to be detected and manipulate events in event streams, also creating new events. Quite often CEP Languages use SQL (*Structured Query Language*) like syntax with additional elements for event analysis

and manipulation. As an example CCL query which analyses the incoming price stream to regularly estimate average price can be following

```
Insert Into StreamAvgPrice //target event stream
  Select Id, Avg(Price)  // data from incoming stream
 From StreamPrice Keep 10 minutes // incoming event stream
 Where Price >= 10000 // condition on incoming stream events
Group By Id // grouping  events
```

The event types for the streams in this case

```
StreamAvgPrice ( Id (String), AvgPrice (Long) )
StreamPrice ( Id (String), Price (Integer) )
```

3 CEP Network

Currently CEP Engines use different event processing models and different languages. There are several researches that offer certain general common model, e.g. [6], but these models are not yet merged into one common model.

For the purpose to consider a general mechanism of CEP Engines and BPEL Engines integration we define CEP Network as a general abstraction of CEP infrastructure. We also define CEP Query model as a model for query execution in this infrastructure. Our proposal is based on analysis of existing CEP Engines and can be considered as a modification of metamodel proposed in [6] focusing on distributed environment.

3.1 Event type model

We do not consider Event Type representation. We find it fairly straightforward and technical task to convert Event Types between different technical data representations. In general we assume that it can be set as certain XML structure with generally arbitrary complex hierarchical structure.

3.2 Generalised CEP Query Model

This model provides representation of CEP query which can be executed in distributed environment with multiple CEP Engines possible.

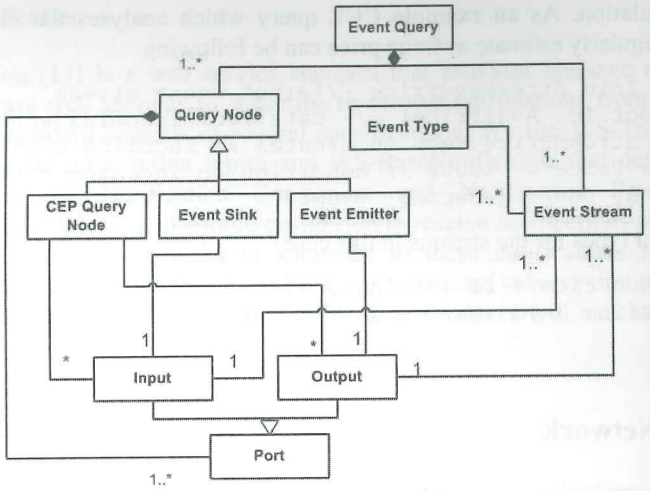

Fig. 1 CEP Query model

See **Fig. 1 CEP Query model** for our proposal of distributed CEP query model. We define Event Query as a set of Query Nodes connected by Event Streams. Event Streams are of defined Event Type and connect Query Nodes using Ports. Ports are either Inputs or Outputs. Each Event Stream connects to exactly one Input and one Output Port. There are 3 distinct Query Node types: Event Emitter nodes for events CEP network receives from outside, Event Sink nodes for events CEP Network sends outside and CEP Query Nodes representing CEP Engines manipulating events. An Emitter has exactly one output port, a Sink has exactly one input port, and CEP Query Nodes may have multiple input and output ports.

3.3 Event processing network infrastructure

Now we can define CEP network as a run time infrastructure corresponding to CEP Query model. The following diagram shows main elements of a CEP Network (**Fig. 2**).

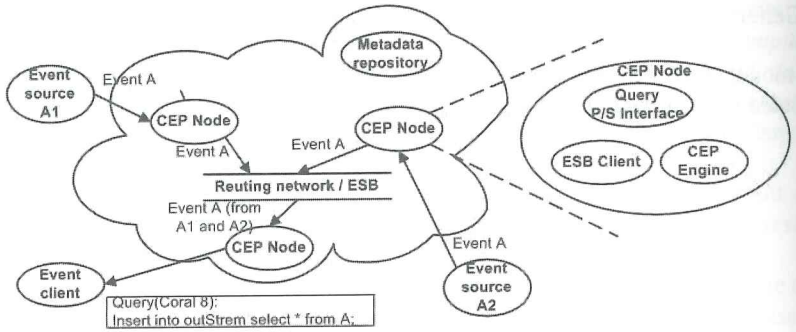

Fig. 2 CEP network runtime infrastructure

There are following components:

Event Sources – which publish definite event types to the Network connecting to Event Emitters. The usual pattern is that each Emitter contains a Listener to which Event source can fire events.

Event Clients - subscribe to CEP Network by submitting CEP Query. CEP Network should ensure CEP Query deployment, processing and results delivery as CEP Events of the corresponding Event Type to Clients. The usual pattern is that Clients subscribe to CEP Network to receive events.

CEP Network – Dynamic infrastructure that ensures CEP Query execution in a distributed environment includes multiple CEP Nodes and controls event flow there. CEP network consists of 3 main elements:

- **CEP processing Nodes** – that ensure execution of CEP Queries and provide interface for communication with CEP Network (using publish /subscribe interface). CEP Engine is logically part of CEP Processing Node and connects to Event Routing network.
- **Metadata repository** – keeps information on the CEP Network components and configuration, event types registered and other.
- **Event Routing Network / Event Service Bus** – middleware that ensures communication between Processing Nodes and also Event Sources and Consumers. The ERN can be considered as a specific case of Enterprise Service Bus optimised for processing of high volumes of such events, and possibly realising certain operations on them.

In the simplest case CEP Network would consist of one CEP Processing Node - CEP Engine.

3.4 Requirements to CEP network

To complete CEP Network definition we need to define integration possibilities and quality of service we expect from it:

- Publish/subscribe (P/S) communication – that is usual pattern supported when interacting with CEP engines, which integrate to some messaging engine.
- In order event delivery –delivery of events within CEP network and to the clients should not change the order of events.
- At least once delivery – events can not be lost in CEP Network.

4 BPM and CEP Network integration

In this chapter we describe principles of the solution and give high level overview of the proposed solution components.

The objective is to provide general and efficient integration mechanism between BPEL Engine and CEP network. We should consider both - integration at design (or modelling) phase, when BPEL process and CEP queries are defined, and runtime.

The functional requirements for runtime integration are to ensure BPEL process to receive CEP events from CEP Network as well as post events to CEP Network. The functional requirements for design time are to provide an easy way to define the event exchange in BPEL and CEP Network design environments.

Considering SOA principles of separation of concern and loose coupling we decide to keep BPEL Engine and CEP Network as independent as possible and also require integration solution to be based on open standards. We propose to implement intermediary CEP Service that will provide native integration methods towards both BPEL process and CEP Network. The key decision is to represent CEP Queries as WSRF resources, which makes it easy to work with them in BPEL Process. The advantage of the approach is that it does not require changes or extensions in the BPEL and CEP Network models.

The design and execution of united BPEL/CEP process and solution components are shown on **Fig. 3**. CEP Query is defined and deployed on CEP Network using CEP service client tool independently from BPEL process. CEP Service ensures automated creation of WSRF resources according to CEP queries. Resource properties are created according to the Event type. The CEP event is reflected as change of the resource properties. Event delivery to consumer in this case BPEL process is represented as WS Notification. CEP Service ensures WSRF standard and WS Notification support for CEP Query during its lifetime.

No changes in BPEL Engine or BPEL design tool are required. BPEL process is defined using standard BPEL design tool. Processing of events from CEP queries within BPEL process requires following code in BPEL Process: provide an operation where actual processing of event to be done, and put a subscription call on relevant WS resource in CEP service, usually to be done at the start of the process passing there the operation. That can be done using standard BPEL possibilities, since WSRF is an extension over webservice.

In order to utilise full possibilities during BPEL process design we do need to deploy CEP query first – since then we get WS resource generated automatically by CEP service and can use these metadata in BPEL process.

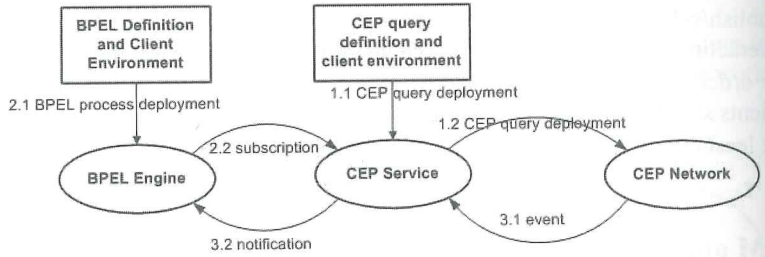

Fig. 3 High level solution description

The opposite direction in integration – BPEL process as Event Source to pass events to CEP Network is not shown on the diagram. We consider that this case in principle is much easier. For the consistency we propose to realise it in similar way - Event Sources to be exposed via CEP Service as WS Resources and BPEL Process

could pass event with data by invoking corresponding operation on the resource. But it could be also a simple web service invocation.

5 Detailed Solution Design

In this chapter we explain the design of the solution in more details. The following diagram Fig. 4 shows solution components and data flows.

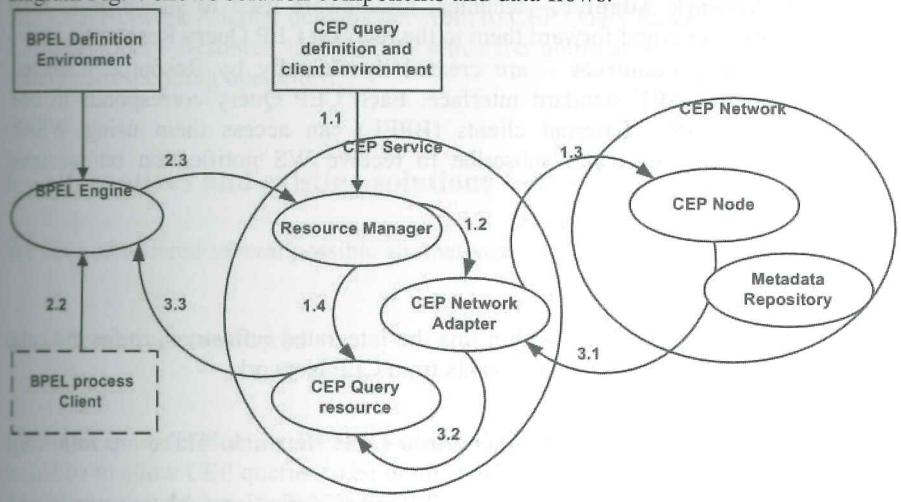

Fig. 4. Solution Components

5.1 Solution components

Here we list solution components.

Design tools. There are following components:

BPEL Design tool – tool for BPEL Process design. Does not require any changes.

CEP service client – an environment to define and manipulate CEP queries. It uses CEP Service to pass CEP queries to CEP Network. CEP Query definition client connects to CEP service not CEP Network because otherwise in case if CEP Network would be single CEP Engine, we would need a specific interface that would notify CEP service that there is new CEP Query deployed and that would require extension of CEP Engine which usually does not provide such notifications.

Runtime infrastructure. There are following components:

BPEL Engine – runtime environment where BPEL process is deployed. Does not require any changes.

CEP Network - components has been considered in details in chapter 3 CEP Network.

CEP service – integration application which consists of:

- **Resource Manager** – allows CEP service client to deploy CEP queries, based on CEP query definition creates CEP query resource components and connects them to CEP Network events (via CEP Network Adapter component). Also provide further access to manage created CEP Queries (update or destroy them).
- **CEP Network Adapter** – technical middleware used to subscribe to CEP Network events and forward them to the relevant CEP Query Resources.
- **CEP Query resources** – are created dynamically by Resource manager, implement WSRF standard interface. Each CEP Query corresponds to one WSRF resource. External clients (BPEL) can access them using WSRF standard operations and subscribe to receive WS notification on resource changes.

5.2 Process flows

We describe now 3 main flows within this the integrated infrastructure for the case when BPEL Engine needs to receive events from CEP Network.

Design and deployment of CEP Query into CEP Network. There are following steps:

1.1. User defines CEP query using some CEP query definition tool (not considered here) and inputs it to CEP service client. CEP service client connects to the Resource Manager in CEP Service and passes it there.

1.2. Resource manager validates the CEP query and starts its registration process. It requests CEP Network Adapter to deploy query to the CEP Network and initialize new CEP Resource.

1.3. CEP Network Adapter subscribes to CEP Processing Node with the received CEP Query passing along the address where events to be sent from CEP Network.

1.4. Resource manager completes creation of CEP Resource.

Design and deployment of BPEL Process. There are following steps:

2.1. BPEL process is designed as usual using BPEL Designer. There should be following 2 steps defined in BPEL process:

- operation defined where processing of received event to be done;
- invocation of subscription call to CEP query resource.

We should note that they can be done in different processes. The resource properties can be clarified either in runtime or during process design using „ResourceProperties" operation. BPEL Process is deployed to BPEL engine which validates and initialises it.

2.2. BPEL instance is initiated. That is independent from deployment of process. BPEL Engine ensures different mechanisms for BPEL process invocation, e.g. providing web service interface to process, once BPEL process web service is

invoked, the BPEL Process instance is created and executed. NO extensions from our solution.

2.3. Executing BPEL process instance according to the BPEL code the subscription call to CEP query resource web service is executed.

Event processing

3.1. CEP event fires in the CEP network. CEP processing node passes it to all subscribers and thus sends it to CEP Network Adapter in the CEP service.

3.2. CEP Network Adapter delivers the event to CEP Query Resource.

3.3. CEP Query Resource using WSN sends the notification to the BPEL process calling corresponding operations.

6 Alternatives and existing solutions

We have considered several possible alternatives.

6.1 Alternatives considered

CEP Query definition in BPEL process. The more tightly coupled integration would be to allow CEP queries to be defined and managed directly in BPEL, which is possible but would require BPEL Extension. That would allow dynamic creation of CEP queries inside BPEL, but would require building in support in BPEL for concrete CEP Engine languages.

ESB Message Channel. Most obvious loosely coupled integration would be to rely on some ESB standard functionality for messages routing. Some Adapter for CEP Network would put events as messages to ESB, and ESB would take care about their routing, as well as configuration of subscriptions would be done in ESB. Thus BPEL process would not need to subscribe to event source. However in this approach we loose the definition of event type – which should be kept separately as well as in this case the control who and how consumes events becomes too distributed.

Additional consideration about ESB usage – in general we should be quite careful when examining usage of ESB, not to route unnecessary messages there - if general purpose ESB is to be used to route the all event flow in CEP network (e.g. low level events) it can result in unnecessary overhead and can have significant impact on performance, since ESB needs to treat these messages as any other messages it will perform certain unnecessary validations and also transformation to XML format. Sure it depends on the ESB realisation. Similarly if ESB needs to take care about subscription and publication – it may unnecessary process events that has no subscriptions.

6.2 Observed Existing Solutions

We have also examined several existing solutions which integrate CEP with BPEL or ESB. Following open source solutions have been observed:

- Sci-Flex[7], integrating Apache Synapse[8] with Esper; Idea described in [7] is to use an Esper mediator deployed on Synapse ESB through which events are passed back and forth. Main disadvantage of the solution is usage of Synapse mediators that don't have support for BPEL processes.
- Servicemix-Esper[9]. Solution integrates Apache Servicemix[10] and Esper; event streams are realised as web services, Esper is encapsulated into component, which calls these webservices. This solution can't be used directly in BPEL, also Servicemix is a JBI container, therefore adds JBI environment dependency;
- Open ESB IEP [8], provides slightly different approach building CEP Support as part of ESB utilising ESB infrastructure for event exchange as ESB messages. This is attempt is to build in CEP Engine functionality into ESB. Main drawbacks include adding specific support to particular vendor solution, routing all CEP events through general purpose ESB, no direct support for BPEL process execution.

Commercial ones:

- Event adapters based: Coral8, Oracle CEP[11], Streambase; propose different technical adapters that allow events to be routed to ESB.
- Integrated solutions: IBM Websphere Business Events[12], Tibco Business Events, where event processing support is included into the general set of BPM tools. Main approach is a standard transformation between different messaging mechanisms. IBM Websphere Business Events for example offers adapters integration, transformations between adapter and BPEL process type definitions.

We see that these solutions in general are quite flexible and powerful. However as explained above in ESB message channel we see certain advantages in WSRF usage as it allows to publish event structure and allows more control for BPEL process itself. We should also note that these solutions are more focused on the opposite direction - business level events to be processed by CEP language queries.

7 Solution Prototype

The main purpose of prototype is to demonstrate the complete integration scenario of CEP Network events usage in BPEL process. The prototype contains realisation of all components of integrated infrastructure based on Open Source products. Prototype source is published as open source project eBPM at Google Code[13].

[7] http://code.google.com/p/sci-flex/
[8] http://synapse.apache.org/
[9] http://code.google.com/p/servicemix-esper/
[10] http://servicemix.apache.org/home.html
[11] http://www.oracle.com/technologies/soa/complex-event-processing.html
[12] http://www-01.ibm.com/software/integration/wbe/
[13] http://code.google.com/p/ebpm/

7.1 Prototype components

Prototype components and their technologies described below.

CEP Network Application

- Event Routing Bus – uses JMS, built on Apache ActiveMQ[14].
- Repository - HSQLDB[15] in memory database.
- CEP Processing Nodes – custom developed Java RMI[16] server which provides publish /subscribe (P/S) interface to CEP queries. CEP Engine encapsulated into processing nodes is Esper (alternatively Coral8 implementation is possible).
- Event Sources – post events to CEP processing Nodes using JMS.

CEP Service Application

- CEP Service Application is built on top of Apache Muse[17] WSRF standard implementation.
- Separate WSRF resource is created dynamically by Resource manager for each CEP Query. CEP Service uses RMI to connect to the CEP Processing Nodes publish subscribe interface. CEP service acts as a standard intermideary propagating client calls to WSRF resource to subscribe to CEP quiries and passes back the events received by RMI. (Note there was a technical problem for client applications to use WSDL files generated by Apache Muse, due to nonstandard type definition, therefore there was a need to make a technical workaround specific adapters).
- CEP Service Application is deployed to Apache Axis2[18] webservice container setup on Jetty[19] web server. Due to hystorical reasons Apache Muse service application is run on a separate Tomcat[20] 6 application server. We have left this distributed CEP service and Muse service setup to demonstrate that.

[14] http://activemq.apache.org/
[15] http://hsqldb.org/
[16] http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp
[17] http://ws.apache.org/muse/
[18] http://ws.apache.org/axis2/
[19] http://www.mortbay.org/jetty/
[20] http://tomcat.apache.org/

CEP Service management client application. This part of prototype has not been completed and thus not included to prototype. Java application. Instead CEP Query configuration for CEP Service is defined in CEP service configuration file and CEP Service implementation relies on Esper Query deployed directly to Esper Engine.

BPEL process environment. Apache ODE[21] 1.2 BPEL Engine deployed on on Apache Tomcat is used for BPEL process execution. Eclipse 3.4. Ganymede[22] has been used for BPEL Process definition. There has not been needed any customisation of the BPEL tools.

7.2 Sample

Sample run on this setup consists of 2 Event sources passing events to the CEP network with one Procesing Node where Esper CEP Engine is setup. We define one CEP Query there. Dummy BPEL process does nothing but subscribes to the WS resource that is created by CEP service for this Query and has some dummy operation for notification processing.

8 Conclusion

We find that WSRF standard is a general and convenient way for information exchange between different systems. WSRF support for notification allows to represent events and messages and gives additional advantage of publishing of the structure of information. We demonstrate that it can be successfully used for a general integration mechanism between CEP Network and BPM Solution. In the very similar and straightforward way we could provide integration between BPEL process and e.g. BI tools.

We find that this integration solution can be easily used in SOA environment where ESB middleware is used for message routing. Since we use web service standard message exchange between CEP Service and BPEL Engine can go through ESB as well.

We also find CEP Network quite useful abstraction which can be used when considering integration with multiple CEP Engines.

References

1. Complex Event Processing, white paper, IBM, 2007, http://domino.watson.ibm.com/comm/research.nsf/pages/r.datamgmt.innovation.cep.html
2. Coral8 CCL Reference 5.5, Coral8, http://www.coral8.com/system/files/assets/pdf/5.5.0/Coral8CclReference.pdf.

[21] http://ode.apache.org/
[22] http://www.eclipse.org/ganymede/

3. Esper Reference Documentation 2.3, Esper, 2009, http://esper.codehaus.org/esper-2.3.0/doc/reference/en/pdf/esper_reference.pdf
4. D.Luckham: A Brief Overview of the Concepts of CEP. http://complexevents.com/?p=399
5. S. Zdonik et. al.: Towards a streaming SQL standard, PVLDB 1(2): 1379-1390, 2008.
6. Event-processing network model and implementation G Sharon, O Etzion - IBM Journal of Research, V 47(2), ISSN 0018-8670, pp. 321-334 (2008)
7. P. Freemantle: Event Stream Processing in Synapse, http://pzf.fremantle.org/2007/07/event-stream-processing-in-synapse.html
8. Open ESB Intelligent Event Processor, https://open-esb.dev.java.net/IEPSE.html
9. Gartner Research white paper: Magic Quadrant for Business Process Management Suites (2007)
10. Forrester Research white paper: ESB Open Source Challenge (2008)
11. Web Services Business Process Execution Language 2.0, OASIS, 2007.; http://docs.oasis-open.org/wsbpel/2.0/
12. Web Services Description Language (WSDL) 1.1., W3C, 2001.; http://www.w3.org/TR/wsdl
13. Web Services Notification 1.3., OASIS, 2006.; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn
14. Web Services Resource Framework 1.2., OASIS, 2006.; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
15. Louis Perrochon, Stephane Kasriel, David C. Luckham: Managing Event Processing Network, Technical Report No.: CSL-TR-99-788 ftp://reports.stanford.edu/pub/cstr/reports/csl/tr/99/788/CSL-TR-99-788.pdf
16. Tammo van Lessen at al.: A Management Framework for WS-BPEL, ecows, 2008 Sixth European Conference on Web Services, pp.187-196 (2008)

Information Systems and Security