



PROJEKTU VADĪBAS PROCESU DEFINĒŠANA UN FORMALIZĒTA ATTĒLOŠANA DEFINITION AND FORMAL REPRESENTATION OF THE PROJECT MANAGEMENT PROCESSES

Solvita Bērziša

Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultāte
Meža iela 1/3 – 400b, Rīga, LV 1048, Latvija
Tel: +(371)26301110, e-pasts: berzisa@gmail.com

Abstract. *Project management information systems are used to support project management processes. The typical project management processes are defined by the project management methodologies, such as PMBOK. It is necessary to formalize representation of these processes to use them in modelling, analysis and implementation of the project management information systems. The purpose of the article is defined key processes of project management and to identify appropriate techniques and tools for their formal representation.*

In the article groups of the project management processes are listed and representation techniques such as UML, XPDL and YAWL are evaluated. The comparison of process representation techniques is shown that the most appropriate is XPDL. Guidelines for project management process definition using XPDL are formulated. These formalized processes further will be used in configuration of the project management information systems.

Keywords: *PMXML, project management processes, workflow, XMI, XPDL, YAWL.*

Ievads

Projektu vadība ir sarežģīts process, kurā būtiska nozīme ir informācijas apstrādes un apmaiņas procesiem. Projektu vadības veiksmīgai realizācijai ir pieejamas dažādas projektu vadības metodoloģijas un projektu vadības rīki. Problēmas rodas, ja projektu vadības rīks ir jāpiemēro projektu vadības metodoloģijai, kas nav saistīta ar konkrēto rīku. Sistemātiska informācijas tehnoloģiju risinājuma izmantošana ļautu uzlabot projekta plānošanas un izpildes kvalitāti un efektivitāti, bet šādas sistēmas ieviešana ir darbietilpīgs process. Lai atvieglotu šo procesu, ir nepieciešama projektu vadības informācijas sistēmu ieviešanas metodoloģija. Šādas metodoloģijas, kuras pamatā ir projektu vadības problēmapgabala formalizēta attēlošana un projektu vadības sistēmas automatizēta konfigurēšana, pamatprincipi ir definēti [1]. Viena no nozīmīgākajām šīs metodoloģijas daļām ir projektu vadības XML shēma (PMXML), kuru izmanto, lai problēmapgabala formalizēto attēlojumu no vizuāla modelēšanas rīka pārnestu uz datorizētu projektu vadības informācijas sistēmu atbilstoši modeļu vadītas izstrādes pieejas principiem. Šajā PMXML shēmā var definēt visus ar projektu vadību saistītos elementus, kurus ir iespējams iedalīt četrās grupās: vienkārši elementi (piemēram, lomas, aktivitātes un artefakti), saliktie elementi (piemēram, izmaiņu un risku vadība), metodes elementi (piemēram, iegūtu vērtību analīze) un operatīvie/ transakcijas elementi (piemēram, kalendāri un bāzlīnijas) [1].

Viens no sarežģītākajiem PMXML shēmas izveides uzdevumiem ir definēt saliktos elementus, jo tie satur dažādu projektu vadības dinamisko procesu aprakstus. Procesu definēšanai šobrīd ir izstrādātas vairākas vispārīgas darbplūsmu definēšanas tehnoloģijas un arī XML shēmas, piemēram, XPDL un YAWL. Vairs nav nepieciešams izstrādāt jaunu shēmu, lai aprakstītu projektu vadības procesu, ir tikai jānoskaidro, kura ir piemērotākā, lai to iekļautu PMXML shēmā kā apakšshēmu procesu aprakstam. Līdz ar to šī raksta pētījuma objekti ir projektu

vadības procesi, to formalizēta attēlošana, izmantojot darbplūsmu tehnoloģijas, un saglabāšana XML formātā.

Šī raksta mērķis ir noteikt galveno projektu vadības procesu attēlošanu esošajām projektu vadības metodēm un izstrādāt to formalizētus attēlojumus. Lai to izdarītu, sākumā ir jānoskaidro, kādi vispār ir projektu vadības procesi un kā tos var sagrupēt. Tāpat arī ir jānoskaidro darbplūsmu modelēšanas līdzekļi, ar kuriem var modelēt procesus un saglabāt tos kādā no XML formātiem. Rezultātā, balstoties uz apkopoto informāciju, ir jānoskaidro piemērotākais XML formāts procesu aprakstīšanai PMXML shēmā un ir jāizvirza vadlīnijas projektu vadības procesu definēšanai, lai tos pēc tam varētu izmantot projektu vadības informācijas sistēmu konfigurēšanā.

Lai atbildētu uz izvirzīto mērķi un uzdevumiem, sākotnēji ir apskatīti vairāki projektu vadības standarti, lai noskaidrotu projektu vadības procesus un procesu grupas. Kā pamata standarts tiek pieņemts PMBOK (*Project Management Body of Knowledge* [2]), bet papildus informācijai ir apskatīti arī citi projektu vadības standarti, kā RUP (*Rational Unified Process* [3]), MSF (*Microsoft Solutions Framework* [4]) un PRINCE2 [5]. Paralēli ir noskaidrotas darbplūsmu modelēšanas un saglabāšanas iespējas un izveidota to klasifikācija. Izmantojot iegūto informāciju, ir izvēlēti un praktiski izmēģināti iespējamie darbplūsmu modelēšanas rīki, un novērtētas to iespējas saglabāt darbplūsmas XML formātā, raugoties no iegūto XML dokumentu tālākās izmantošanas parocīguma. Rezultātā ir noskaidrots projektu vadības procesu glabāšanai iespējami piemērotākais procesu modelēšanas rīks un procesu definēšanas XML standarts. Noslēgumā ir apkopotas vadlīnijas projektu vadības procesu definēšanai.

Projekta vadības procesi un to grupas

Projekts sastāv no daudziem procesiem, kurus vispārīgi iedala divās pamata kategorijās [2]:

- projekta vadības procesi, kas apraksta, organizē, kontrolē un noslēdz projekta darbus;
- uz produktu orientēti procesi, kas specificē un veido projekta produktu.

Projekta vadības procesus, kā tos definē PMBOK, var iedalīt grupās pēc diviem kritērijiem – pēc projektu vadības zināšanu apgabaliem un projektu vadības procesu grupām. PMBOK definē, ka ‘Projekta vadības plāns’ ir galvenais projekta vadības procesu organizējošais dokuments, kas ietver vairākus apakšplānus: satura, grafika, izmaksu, kvalitātes, procesu uzlabošanas, personāla, komunikācijas, risku un iepirkumu vadības plānu. Šis saraksts vēl ir papildināms ar izmaiņu un problēmu risināšanas plāniem. Daži no šiem plāniem ir apvienoti vienā zināšanu apgabalā, tādēļ viss projekta vadības process tiek sadalīts atbilstoši deviņiem PMBOK zināšanu apgabaliem. Kā vienu procesu var apskatīt arī visu projektu vadību kopumā [2].

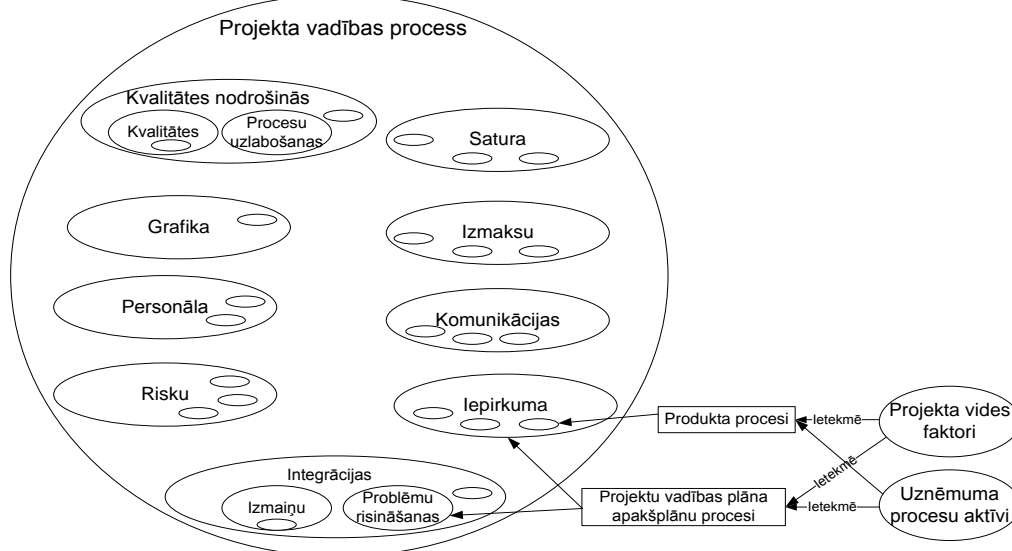
Uz produktu orientēti procesi tiek noteikti atbilstoši projekta un produkta veidam. Tie tiek definēti, balstoties uz atbilstošiem standartiem vai uz uzņēmumā jau eksistējošiem procesiem. Informācijas sistēmu izstrādes apgabalā šādus procesus palīdz definēt tādi standarti kā RUP [3] un MSF [4].

Rezultātā projekta vadības procesi nosacīti tiek iedalīti trīs līmeņos:

- I - projekta vadības process;
- II - projekta vadības plāna apakšplānu procesi;
- III - produktu un specifisku notikumu orientēti procesi.

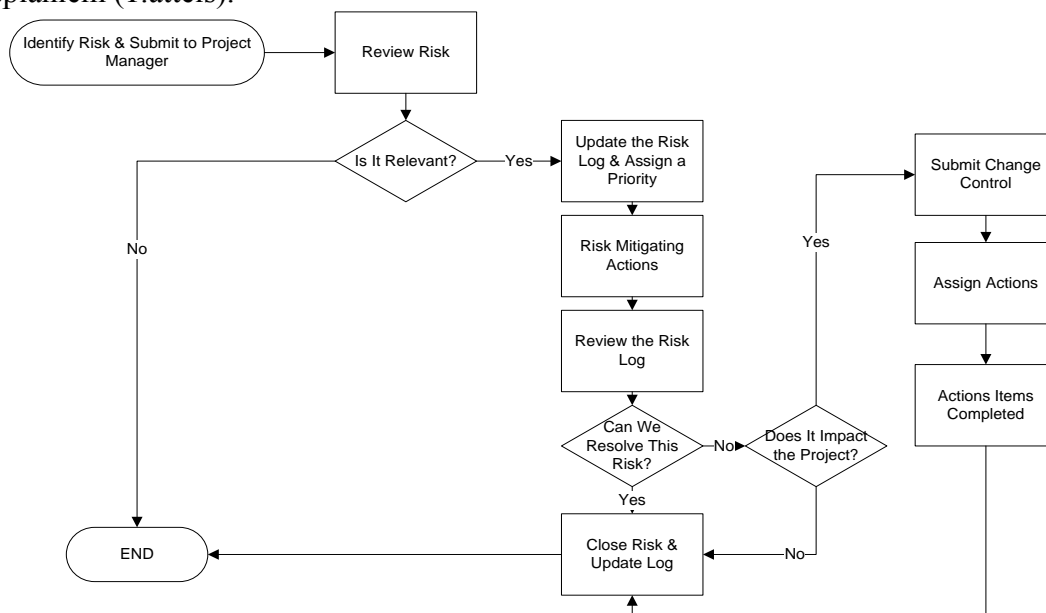
Vispārīgs šo procesu līmeņu strukturējums un attiecības ir redzamas 1.attēlā. Projekta vadības process ir definēts, balstoties uz PMBOK aprakstītajiem procesiem, savienojot tos vienā kopumā [2]. Projekta vadības process ir noteikts, tā struktūra nedaudz var tikt manīta atkarībā no apgabaliem, kas nav saistoši konkrētajam projektam. Piemēram, projektam nav iepirkumu, līdz ar to nav arī procesu, kas saistīti ar iepirkumu plānošanu, organizēšanu un kontroli. Visi procesi, kas ir projekta vadībā, nosacīti tiek apvienoti šajā projekta vadības procesā (1.attēls).

Ar projekta vadības plāna apakšplānu procesiem tiek nodrošināti dažādi projekta plānošanas, izpildes un kontroles procesi. Apakšplānu grupējums tiek noteikts pēc *PMBOK*, bet tas, kas kurā plānā ir jādarā, tiek definēts atkarībā no projekta vides faktoriem, uzņēmuma procesu aktīviem vai kādiem izstrādātiem standartiem. Bet šajos plānos ietilpst tikai projekta organizatoriskie procesi, un to savstarpējo mijiedarbību nosaka projekta vadības process.



1. att. Procesu iedalījums projektā vadībā

Produktu procesos tiek ietverti visi procesi, kas konkrētāz noteiktu projektu un tā rezultātu. Šie procesi tiek saistīti ar konkrētu produktu vai produkta standartu, vai projekta notikumu, piemēram, noteikta riska iestāšanās. Produktu procesi ir pakārtoti projekta vadības apakšplāniem (1.attēls).



2.att. Risku apstrādes procesa piemērs

Viens no projekta vadības procesu piemēriem ir risku apstrādes process (2.attēls), kas šajā gadījumā ietilpst projekta vadības riska apakšplāna procesos un ir definēts, par pamatu ņemot J.Charvata piedāvāto risku apstrādes procesu [6]. Šis process tiek izmantots turpmākajos piemēros.

Projekta procesu darbplūsmu attēlošanas rīki un darbplūsmu saglabāšanas iespējas

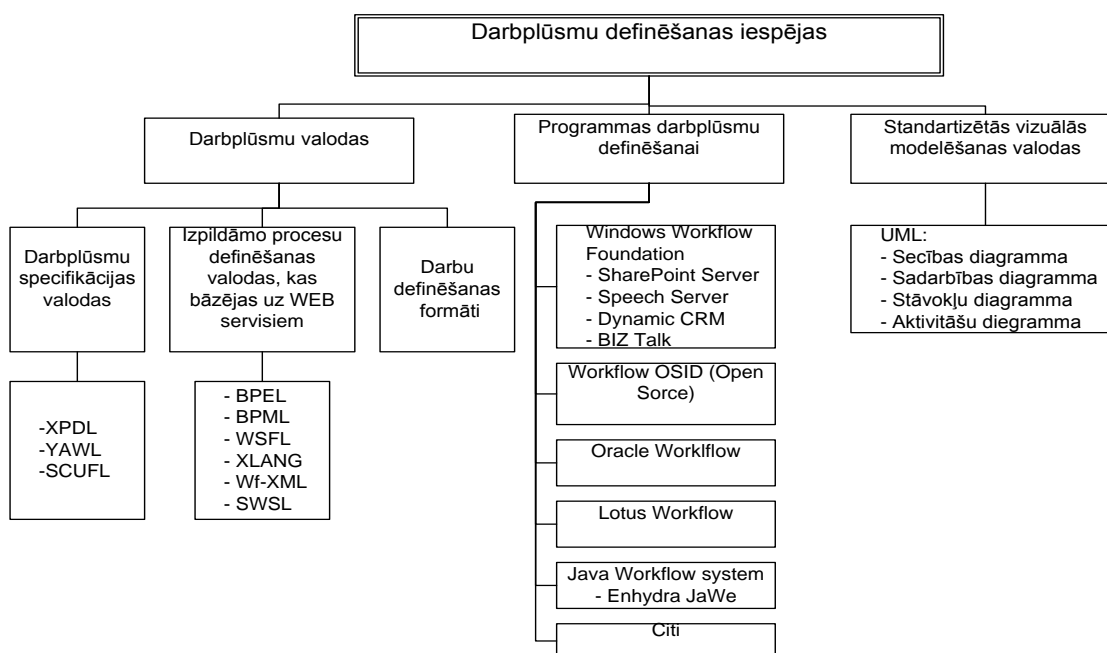
Formālā darbplūsmas definīcijā tiek teikts, ka darbplūsma aptver apstākļus, resursus un triggerus, kas ir saistīti ar specifisku procesu [7]. Reāli darbplūsma ir secīgu operāciju attēlojums, kas abstrakti apraksta reālus darbus [8]. Darbplūsmu sākotnējais teorētiskais pamatojums ir saistīts ar Petri tīkliem [7].

Aplūkojot darbplūsmu modelēšanas iespējas, tiek izdalītas vairākas darbplūsmu definēšanas iespēju grupas (3.attēls). Trīs galvenās grupas ir:

- 1) darbplūsmas sistēmu specializētu iekšējo modelēšanas valodu izmantošana;
- 2) specializētu darbplūsmu un biznesa procesu modelēšanas valodu izmantošana;
- 3) vispārīgu modelēšanas valodu izmantošana.

Darbplūsmas sistēmas ir paredzētas gan darbplūsmu definēšanai, gan izpildīšanai. Šādu sistēmu piemēri ir *SharePoint Server*, *Oracle Workflow*, *Lotus Workflow*, *YAWL* un citas. Šajās darbplūsmu sistēmās informācija pārsvarā tiek glabāta iekšējos darbplūsmu glabāšanas formātos. Jaunākās paaudzes sistēmas nodrošina arī eksportēšanu uz XML bāzētos formātos.

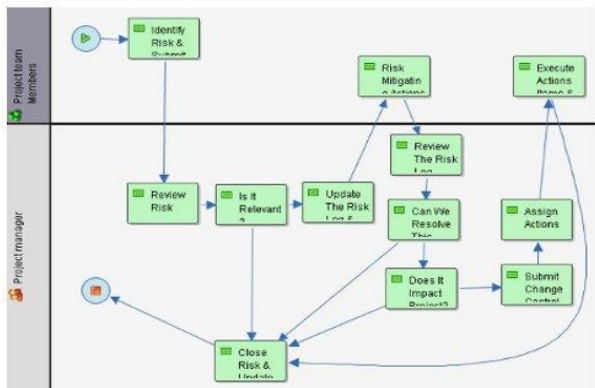
Ir pieejamas vairāku veidu specializētas darbplūsmu definēšanas valodas (3.attēls), no darbplūsmu specifiskajām valodām XML bāzētas ir, piemēram, *XPDL* un *YAWL*. Pārējām valodām bieži ir nodrošināta eksportēšana uz šiem formātiem, visbiežāk uz *XPDL*. Šajā grupā ir iekļautas arī grafiskās modelēšanas valodas, kurās ir nodrošināta XML bāzēta darbplūsmas attēlojuma ģenerēšana.



3.att. Darbplūsmu definēšanas iespējas

Vispārīgo vizuālo modelēšanas valodas izmantošana, lai definētu procesus, ir nedaudz savādāka pieeja nekā darbplūsmu modelēšana, jo tiek izmantota vispārpieņemtās dinamiskās modelēšanas diagramma. Viena no šādām modelēšanas valodām ir *UML*, no kuras procesu modelēšanai var izmantot diagrammas, kas nodrošina dinamisku sistēmas skatu, tas ir, aktivitāšu, secības, sadarbības un stāvokļu diagrammas [7]. *UML* diagrammas parasti tiek saglabātas izstrādes videi atbilstošā formātā, bet kā *UML* datu apmaiņas XML bāzēts formāts tiek izmantots *XMI*.

Rakstā, aplūkojot darbplūsmu modelēšanas rīkus, uzmanība tiek pievērsta tikai valodām, kuras ir bāzētas uz XML formāta. Visbiežāk lietotas šāda veida valodas ir *XPDL* un *XMI* [9], bet ir arī trešā iespēja – *YAWL*.

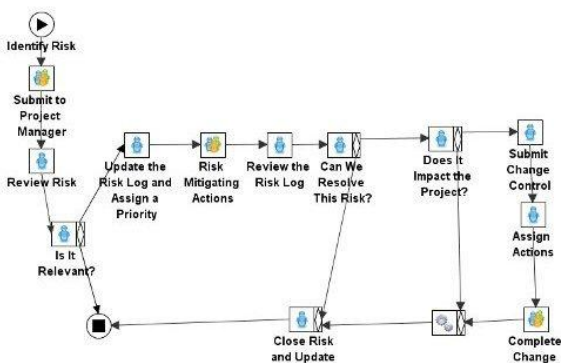


```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Package xmlns="http://www.wfmc.org/2002/XPDL1.0" xmlns:xpd="http://www.wfmc.org/2002/XPDL1.0" ...
...
<Participants>
  <Participant Id="Risk_pa1" Name="Project team Members">
    <ParticipantType Type="RESOURCE_SET"/>
  </Participant>
  <Participant Id="Risk_pa2" Name="Project manager">
    <ParticipantType Type="ROLE"/>
  </Participant>
</Participants>
<WorkflowProcesses>
  <WorkflowProcess Id="risk-process" Name="risk-management-process">
    ...
    <Activities>
      <Activity Id="risk-process_act12" Name="Does It Impact Project?">
        <Implementation>
          <No/>
        </Implementation>
        <Implementation>
          <Performer>Risk_pa2</Performer>
          <TransitionRestrictions>
            <TransitionRestriction>
              <Split Type="XOR">
                <TransitionRef Id="risk-process_tra13"/>
                <TransitionRef Id="risk-process_tra12"/>
              </Split>
            </TransitionRestriction>
          </TransitionRestrictions>
        </Implementation>
      </Activity>
    </Activities>
    <Transitions>
      <Transition>
        <Transition From="risk-process_act12" Id="risk-process_tra12" To="risk-process_act7">
          <Condition Type="CONDITION"/>
        </Transition>
      </Transition>
    </Transitions>
  </WorkflowProcess>
</WorkflowProcesses>
  
```

4.att. Riska apstrādes procesa piemēra darbplūsma un ģenerētās XPDL shēmas fragments

XPDL saīsinājums nozīmē XML procesu definēšanas valoda. Šo valodu ir izstrādājuši darbplūsmu vadības koalīcija (WfMC – Workflow Management Coalition) ar mērķi, lai nodrošinātu procesu definīciju apmaiņu starp dažādiem darbplūsmu produktiem [10]. XPDL tiek definēts, izmantojot XML shēmu, lai specificētu darbplūsmu/ biznesa procesu deklarējamās daļas. Arī XPDL pašlaik ir labākais faila formāts, lai nodrošinātu informācijas apmaiņu starp BPMN (biznesa procesa darbplūsmas) modelēšanas vidēm. XPDL tiek nodrošināti daudzi no darbplūsmu šabloniem [11]. 4.attēlā ir izveidots neliels paraugs XPDL shēmai iegūšanai, kurā, izmantojot Together Workflow Editor, ir uzzīmēts process, kas redzams 2.attēlā, un ir apskatīti ģenerētās XML shēma fragmenti.



```

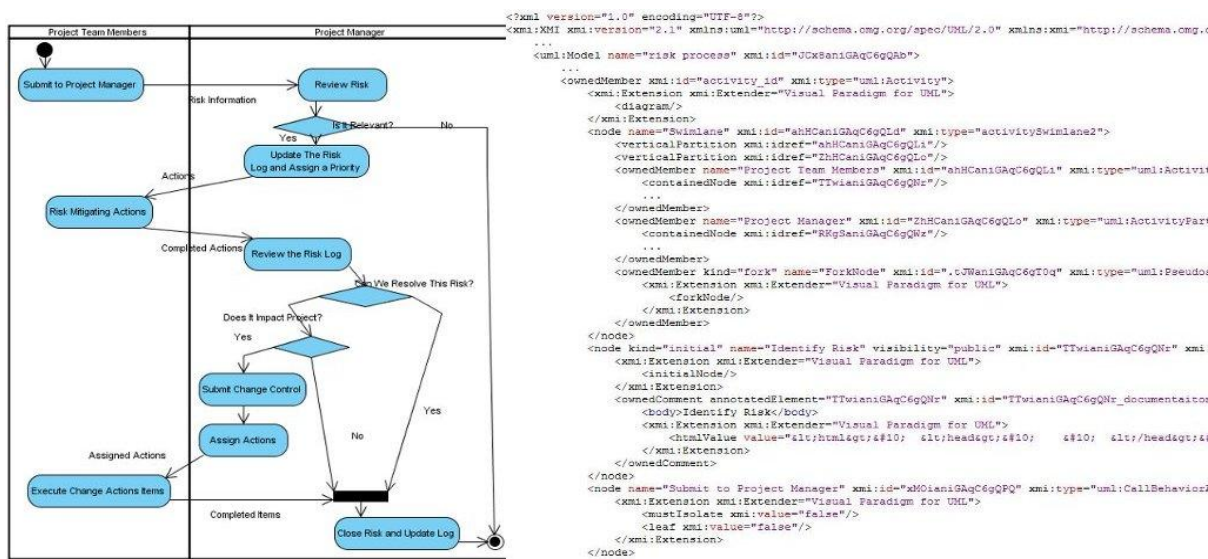
<?xml version="1.0" encoding="UTF-8"?>
<specificationSet xmlns="http://www.yawlfoundation.org/yawlschema" xmlns:xsi="http://www.
...
<specification uri="risk.yawl">
  <metaData>
    ...
  </metaData>
  <metaData>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" />
    <decomposition id="Risk_management_process" isRootNet="true" xsi:type="NetFactsType">
      <externalInteraction>manual</externalInteraction>
      <processControlElements>
        <inputCondition id="InputCondition_1">
          <name>Identify Risk</name>
          <flowsInto>
            <nextElementRef id="Submit_to_Project_Manager_9" />
          </flowsInto>
        </inputCondition>
        ...
        <task id="Review_Risk_11">
          <name>Review Risk</name>
          <flowsInto>
            <nextElementRef id="Is_It_Relevant_36" />
          </flowsInto>
          <join code="xor" />
          <split code="and" />
        </task>
        <task id="Is_It_Relevant_36">
          <name>Is It Relevant?</name>
          <flowsInto>
            <nextElementRef id="Update_the_Risk_Log_and_Assign_a_Priority_38" />
          </flowsInto>
          <predicate>true(YES)</predicate>
        </task>
        <nextElementRef id="OutputCondition_2" />
      </processControlElements>
    </decomposition>
  </specification>
</specificationSet>
  
```

5.att. Riska apstrādes procesa darbplūsma un ģenerētās YAWL shēmas fragments

YAWL (Yet Another Workflow Language) ir darbplūsmu definēšanas valoda, kas atšķirībā no XPDL balstās uz stāvokļiem bāzētiem darbplūsmu šabloniem un augsta līmeņa Petri tīkliem, tādējādi tiek nodrošinātas daudz plašākas iespējas [12]. Šī valoda ir izstrādāta Eindhovenas Tehniskajā universitātē un Kvīnslendas Tehniskajā universitātē pēc eksistējošo darbplūsmu sistēmu un valodu problēmu izanalizēšanas, lai novērstu visus to trūkumus. [12, 13]. YAWL shēmas iegūšanas demonstrēšanai 5.attēlā ir izveidots neliels piemērs, kurā, izmantojot YAWL

Editor, ir uzzīmēts process, kas redzams 2.attēlā, un ir attēloti fragmenti no iegūtās XML shēmas.

XMI (XML metadata interchange) ir formāts, kuram viena no izmantošana iespējām ir UML modeļu informācijas attēlošana XML formātā [14,15]. Viens no XMI pamatzdevumiem ir nodrošināt vieglu metadatu apmaiņu starp UML bāzētu modelēšanas rīku un programmatūras ģenerēšanas rīku, lai realizētu modeļu vadāmu izstrādi [14]. UML diagrammas tiek izstrādātas, izmantojot dažādas vizuālās UML modelēšanas programmas, kas nodrošina diagrammu saglabāšanu XMI formātā. Daži UML rīki piedāvā arī modificētas UML iespējas, piemēram, biznesa procesu diagrammu, un atļauj to eksportēt uz XMI, bet šī XMI shēma ir nedaudz modificēta aktivitāšu diagramma izstrādātāju vajadzībām. Vienkāršs piemērs XMI shēmas iegūšanai ir redzams 6.attēlā, kur, izmantojot Visual Paradigm for UML, ir izveidota aktivitāšu diagramma procesam, kas ir redzams 2.attēlā, un ir apskatīti fragmenti no ģenerētās XML shēmas.



6.att. Riska apstrādes procesa aktivitāšu diagramma un ģenerētais XMI shēmas fragments

Projektu vadības procesu modelēšana valodas izvēle un procesu definēšanas vadlīnijas

Lai saprastu, kuru XML bāzēto formātu labāk izmantot kā PMXML shēmas sastāvdaļu, ir izveidots priekšrocību un trūkumu apkopojums, kas ir apskatāms 1.tabulā. Apskatot šīs tabulas rezultātus, par vispiemērotāko ir jāatzīst XPDL shēma, lai aprakstītu projektu vadības procesus. Šī shēma pašlaik ir visizplatītākā, salīdzinot ar XMI, arī daudz vienkāršāka, un nodrošina visas iespējas, kas ir vajadzīgas saistībā ar PMXML. Kā piemēri var tikt minēti artefakti un izpildītāji (lomas), kas ir iekļautas arī pašā PMXML shēmā [1].

Lai XPDL formātā saglabāto projekta vadības procesu varētu pēc iespējas labāk izmantot projektu vadības informācijas sistēmu konfigurēšanā, tas tiek definēts, balstoties uz šādām vadlīnijām:

- 1) procesa izpildītāji tiek aprakstīti, izmantojot projektā definētās lomas vai resursu grupas, kas ir resursu sadalījuma diagrammā;
- 2) aktivitāšu nosaukumi tiek formulēti pēc iespējas konkrētāki, īsāki un vēlams angļu valodā, lai pēc tam tie būtu vieglāk apstrādājami konfigurēšanas procesā;
- 3) zarošanās punkti tiek definēti ar pēc iespējas precīzākiem nosacījumiem, pa kuru zaru iet un kas to nosaka, tas ir, automātiski vai ar lietotāja ar noteiktu lomu iejaukšanos;
- 4) definējot procesu, aktivitātēs tiek norādīti artefakti, lai tiktu nodrošināta procesa darbības kontrole. Artefakti jeb sasniegumi tiek definēti tā, lai tos arī varētu iekļaut kopējā projekta artefaktu sarakstā.

Darbpļūsmu XML formāta iespēju salīdzinājums

	<i>Priekšrocības</i>	<i>Trūkumi</i>
<i>XPDL</i>	Plaši izmantots formāts, daudzas citas darbpļūsmu valodas (piemēram, <i>BPMN</i>) un sistēmas nodrošina eksportēšanu uz <i>XPDL</i> Jau ir izstrādāta transformācija no <i>XMI</i> uz <i>XPDL</i> [16] Tiek definēti artefakti, asociācijas, kontroles un informācijas plūsmas, kas ir aktuālas arī projektu vadībā Tiek definētas lomas un citas resursu grupas, kas izpilda aktivitāti	<i>XPDL</i> ir ierobežotākas iespējas kontroles plūsmu definēšanā, salīdzinot ar <i>YAWL</i> un augsta līmeņa Petri tīkliem
<i>YAWL</i>	Plašākas iespējas nekā <i>XPDL</i> , jo tiek izmantoti darbpļūsmu šabloni Tiek nodrošināta <i>BPMN</i> diagrammu transformēšana uz <i>YAWL</i> tīklu [17]	Nevar tikt norādīti izpildītāji (lomas) Nav plaši izmantojama, reāli var tikt izmantota tikai <i>YAWL</i> sistēmā.
<i>XMI</i>	Daudz rīku, kuros tiek nodrošināta <i>UML</i> diagrammu modelēšana Ir izstrādātas pieejas, kā dažādas <i>UML</i> diagrammas pārveidot par procesu diagrammām [18]	Sarežģīta <i>XML</i> shēma un tās transformācijas Modelēšanā ir nepieciešamas zināšanas par <i>UML</i>

Secinājumi

Apskatot projektu vadības procesus un dažādas procesu modelēšanas iespējas, izmantojot darbpļūsmas, rakstā ir atbildēts uz trīs jautājumiem, kas ir aktuāli *PMXML* shēmas salikto elementu definēšanā. Pirmkārt, ir noskaidrots, kādi procesi ir projektu vadībā un kā tos ir iespējams sagrupēt trīs grupās, kas ir pakārtotas viena otrai. Šīs visas procesus grupas ir sastopamas *PMXML* shēmā, tikai katra grupa shēmā tiek iekļauta citā vietā un attiecībās. Otrkārt, ir apkopotas procesu definēšanas iespējas un pieejamie procesu attēlošanas *XML* bāzētie formāti. Salīdzinot šos formātus, ir noskaidrots, ka projektu vadības procesu aprakstīšanai labāk ir piemērots *XPDL* standarts. Šī *XML* shēma arī turpmāk tiks izmantota kā *PMXML* sastāvdaļa, lai aprakstītu procesus. Treškārt, ir nodefinētas vadlīnijas, lai precīzāk aprakstītu procesus, kuri tiks izmantoti *PMXML* un vēlāk, lai konfigurētu projektu vadības informācijas sistēmu.

Turpmāk rakstā iegūtā informācija tiks izmantota, lai definētu *PMXML* shēmas saliktos elementus un izstrādātu projektu vadības informācijas sistēmas ieviešanas metodoloģijas piemērus.

Summary

One problem of implementation of project management information systems is their compliance with specific project management standards or methodologies. Systematic use of information technology solutions allows improving the situation, but it is a laborious process. In order to ease this problem, a semi-automated model driven methodology for implementation of project management information can be used. Main principles of such a methodology have been elaborated by the author in previous investigations. This methodology is based on formal representation of the project management problem domain and automated configuration of the project management system. One of the most important parts of this methodology is the project management XML schema (PMXML), which is used for transfer of the formal representation of project management methodologies from a visual modelling tool to a computerized project management information system. The PMXML schema defines all features of the project management that can be divided into four groups:

simple entities (e.g., roles, activities and artefacts), composite entities (e.g., change and risk management), methods entities (e.g., earned value analysis) and the operational/transactional entities (e.g., calendars and baseline). Definition of composite entities is challenging because they contain description of various dynamic process of project management. Several technologies are available for definition of general processes and workflows including XML based workflow modelling languages such as XPDL and YAWL. Therefore, it is not necessary to develop a new framework for describing project management processes. Though it is necessary to determine, which languages are suitable for inclusion in the scheme PMXML as a subschema for process description. This paper answers three questions that are important to definition of PMXML scheme composite entities. First, characteristics of project management processes have been explored. Breakdown and grouping of the processes necessary in order to organize these processes in PMXML scheme are provided. Second, workflow definition options, which can be used for representation of project management processes in XML format, are analyzed. Three such representations have been selected: XPDL, YAWL and XMI. In the paper, all these languages have been viewed, practically tested and compared. Consequently, it is concluded that the most suitable is the XPDL standard because it provides all necessary components what are needed to describe project management processes. Third, the paper summarizes guidelines how the process has been defined, so that it could be used for configuration of project management information systems.

Literatūra

1. Bērziša S., Grabis J. An Approach for Implementing of Project Management Information Systems. Information Systems Development: Towards a Service Provision Society, Springer-Verlag: New York (IN PRESS), 2008.
2. A Guide to the Project Management Body of Knowledge (PMBOK guide) Third Edition. Project Management Institute, Four Campus Boulevard, Newtown Square, 2004. 388 p.
3. Kruchten P. The Rational Unified Process: An Introduction. Third Edition. Addison-Wesley., 2003. 336 p.
4. Turner M.S.V. Microsoft Solutions Framework Essentials: Building Successful Technology Solutions. Microsoft Press, 2006. 312 p.
5. Hedeman B., Heemst G.V. van, Fredrskz H. Project Management Based on Prince2 - Edition 2005. Van Haren Publishing, 2007. 250 p.
6. Charvat J. Project management methodologies: selecting, implementing and supporting methodologies and processes for projects. John Wiley& Sons, 2003. 260 p.
7. Aalst W.M.P. van der, Hee K. Van. Workflow Management: Models, Methods, and Systems. The MIT Press, Cambridge, Massachusetts, London, England, 2002. 368 p.
8. Workflow – Wikipedia: http://en.wikipedia.org/wiki/Workflow_system (20.01.2009.)
9. Crusson T. Business Process Management Essentials. GLiNTECH, 2006. [<http://www.glintech.com/downloads/BPM%20Essentials%20with%20Open%20Source.pdf>] (15.01.2009.)
10. Workflow Standard. Process Definition Interface – XML Process Definition Language, Version 2.1a. The Workflow Management Coalition, October 10, 2008.
11. Aalst W.M.P. van der. Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language [<http://is.tm.tue.nl/research/patterns/download/ce-xpdl.pdf>](01.12.2008.)
12. Aalst W.M.P. van der, Aldred L., Dumas M., Hofstede A.H.M. ter. Design and implementation of the YAWL system [http://www.yawl-system.com/documents/yawl_system.pdf](11.11.2008.)
13. Arthur H. Aalst W.M.P. van der. YAWL: yet another workflow language. Information Systems 30(4), 2005. pp. 245-275 [Author version: eprints.qut.edu.au/archive/00010244/01/10244.pdf] (01.07.2008.)
14. XML Metadata Interchange – Wikipedia: http://en.wikipedia.org/wiki/XML_Metadata_Interchange [20.01.2009]
15. MOF 2.0/XMI Mapping, Version 2.1.1. Object Management Group, December 2007. [<http://www.omg.org/docs/formal/07-12-01.pdf>] (20.01.2009.)
16. Jiang P., Mair Q., Newman J. Using UML to design distributed collaborative workflows: from UML to XPDL. Enabling Technologies: Infrastructure for Collaborative Enterprises. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Volume, Issue, 9-11 June 2003. pp. 71-76.
17. Decker G., Dijkman R., Dumas M., Banuelos L.G. Transforming BPMN Diagrams into YAWL Nets. [<http://bpt.hpi.uni-potsdam.de/pub/Public/GeroDecker/bpmdemo2008-bpmn2yawl.pdf>] (10.01.2009.)
18. Lassen K.B., Dongen B.F. van, Aalst W.M.P. van der. Translating Message Sequence Charts to other Process Languages using Process Mining. [<http://is.tm.tue.nl/staff/wvdaalst/publications/p379.pdf>] (24.01.2009.)