

基于模型驱动架构的 workflow 管理系统开发研究

武 凌, 马 季

(安徽财经大学 管理科学与工程学院, 安徽 蚌埠 233030)

摘 要:针对企业在导入或转换新的 workflow 平台时重新建模的问题, 提出基于 MDA 的 workflow 管理系统开发架构。此架构中 workflow 的建模分成两个阶段, 首先设计了 workflow 的 PIM 元模型并进行 PIM 建模, 然后将 PIM 转换成 PSM 模型, 最后再由 PSM 自动产生流程定义文件, 这样在转移 workflow 平台时, 流程的 PIM 模型可以保留并以模型转换技术对应至新平台的 PSM 模型。给出了由 PIM 元模型转换为 WfMC PSM 模型的程序以及由 WfMC PSM 模型转换为 XPDL 流程定义的程序, 并以电子订单 workflow 为例进行了验证。此开发架构降低了流程平台转移时的成本与复杂度, 提高了 workflow 模型的可重用性。

关键词:模型驱动架构; workflow 管理; 平台独立模型; 平台相关模型; XML 流程定义语言

中图分类号:TP393.07; TP311.52

文献标识码:A

文章编号:1673-629X(2011)06-0089-06

Developing Research of Workflow Management System Based on Model-Driven Architecture

WU Ling, MA Ji

(School of Management Science and Engineering, Anhui University of
Finance & Economics, Bengbu 233030, China)

Abstract: Facing the problem of modeling renewably when importing or switching to a new workflow platform, propose a workflow management system developing architecture based on MDA (Model-Driven Architecture). The architecture divides the modeling procedure into two phases. First phase is PIM modeling, then PIM to PSM and PSM to workflow definitions. So PIM can be kept and transferred to new platform's PSM. Introduce the program of PIM to WfMC PSM and WfMC PSM to XPDL, and validate the architecture with an electronic order workflow. The architecture reduces the transition cost and complexity, and enhances the model reusability.

Key words: MDA; workflow management; PIM; PSM; XPDL

0 引 言

过去几年来, 已有许多 workflow 软件问世, 但是大部分软件其流程模型与流程定义格式自成格局且互不兼容, 形成了 workflow 平台上的异质性^[1], 造成不同的 workflow 平台在合作与转移过程中的困难。当企业需要转移 workflow 的执行平台时, 由于旧有平台的 workflow 模型无法使用于新平台, 所以必须重新在新平台下建模。另外, 进行流程整合时也会因为双方的流程平台不认识对方流程的信息造成流程整合的困难^[1]。因此, WfMC (Workflow Management Coalition) 制定了开发 workflow 系统的标准来减少异质性环境带来的困

扰。但建立标准之后, 在 workflow 平台的转移与合作上面面临的困境其实并没有完全解决, 在导入或转换新的流程平台时, 流程开发人员仍须重新将流程建模在新平台上。

过去在系统建模时, 常将系统的逻辑与实现平台的信息一起建构在模型中, 造成模型在转换实现平台时往往无法再继续使用。为此 OMG (Object Management Group) 于 2001 年提出了模型驱动架构 (MDA, Model-Driven Architecture) 的软件开发方式^[2], 此架构中将系统模型的开发分为两阶段, 首先建立只描述系统抽象逻辑设计的 PIM (Platform Independent Model), 再将 PIM 结合实现平台的信息产生 PSM (Platform Specific Model), 最后按照 PSM 实现系统的程序代码。所以当系统需要转移到新的实现平台时, 不需要重新建立系统的模型, 只需要将系统的 PIM 按照新的实现平台转换成对应的 PSM, 再实现新平台的程序代码即可, 由此可大幅增加模型的重复使用性。

收稿日期: 2010-11-11; 修回日期: 2011-02-19

基金项目: 安徽省高等学校自然科学基金项目 (KJ2009B124Z); 安徽财经大学青年科研项目 (ACKYQ0943ZC); 安徽财经大学信息工程学院青年项目 (xgky2008001)

作者简介: 武 凌 (1977-), 男, 安徽蚌埠人, 讲师, 硕士, 研究方向为计算机网络与软件、管理信息系统。

workflow 管理系统实施过程中通常需要针对具体的业务过程进行业务系统的改造和重构,这就为 MDA 进入 workflow 管理系统提供了现实的基础^[3,4],有一些研究者也开始从 MDA 的角度研究 workflow 系统^[5-8]。笔者将 MDA 的开发思想导入 workflow 管理系统的开发中来,以降低在转移流程平台时花费的成本。由于流程模型无法重复利用的主要原因在于流程模型的异质性,所以将 workflow 模型的建立划分成两阶段,第一阶段建立仅描述流程的 PIM,第二阶段按照部署的流程平台将 PIM 转换成该平台的 PSM,再由 PSM 产生该平台的流程定义文件。除此之外,当需要将旧有的流程平台转移至新的平台时,也可以通过反向工程技术将旧平台的流程模型先转换成 PIM 之后,再重新部署到新的流程平台上。

1 PIM 元模型 (meta-model) 设计

1.1 元模型设计

XPD L (XML Process Definition Language) 是 WfMC 以 W3C 的 XML Schema 制定的流程定义格式^[9],目的在于建立统一的流程定义格式。XDPL 的 Process meta-model 和 Package meta-model 中定义了一组基本的工作流程实体与属性进行流程定义的交换,包括了 Workflow Process Definition、Workflow Activity、Workflow ActivitySet、Workflow Participant、Transition Information、Workflow Application、Workflow Relevant Data、Workflow Package 等。参考 WfMC 参考模型中描述的工作流程元素,设计了一个具有扩充性的且平台独立的工作流程 PIM,称之为 MDAWF meta-model,并以 OMG 的 MOF (Meta Object Facility) 与 XMI (XML Metadata Interchange) 对此模型进行定义。

MDAWF 流程模型由 5 种元素组成,分别为 Process、Task、Transition、Resource 与 RelevantData。其中 Process 元素代表一个 workflow,Task 元素代表流程内部的工作单元,工作单元间的转移以 Transition 元素来描述,流程中使用的资源以 Resource 元素来表示,在流程中传递的数据以 RelevantData 来描述。

图 1 是 MDAWF 与 Process 有关的部分。在 MDAWF 中,由于每个 workflow 元素都有各自的识别码(id)、名称(name)、描述(description)等属性,所以独立出包含这些属性的 WFElement 来代表模型中最抽象的基本元素,再由它继承衍生出 workflow 中的 5 个基本元素。除了有继承的关系外,还有 Composite 与 Association 的关系,Composite 的关系如 Process 元素可包含 1 至多个 Task 元素、0 至多个 Transition 元素等,而 Association 的关系如 Transition 元素与 Task 元素间的关联。流程内的工作单元与资源可以做进一步的

细分,基本上工作单元除了可以是执行单一工作(程序)的活动外,也可以是一个子流程或是一个为完成某个目标的群组工作,所以进一步地从 Task 元素继承衍生出 SubFlow 与 GroupTask 这两个子元素来分别建模子流程与群组工作,其中 GroupTask 可以包含 1 至多个其它 Task 元素,而 SubFlow 元素则会参照到模型中的其它 Process。另外, workflow 中的资源也可以再细分为参与 workflow 的角色与应用程序,所以由 Resource 元素会再衍生出 Role 与 Application 两个子元素。

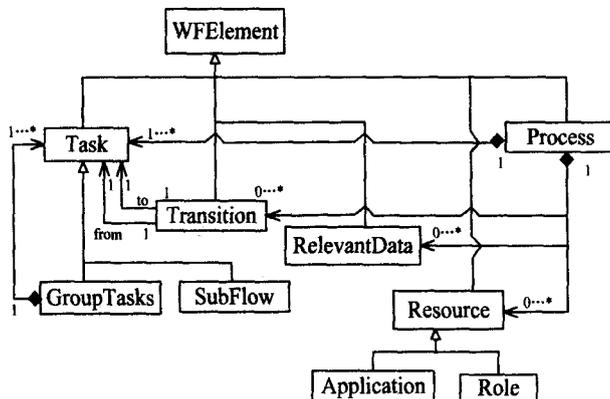


图 1 MDAWF 与 Process 有关的部分

图 2 是 MDAWF 与 Package 有关的部分。在 MDAWF 中定义了 Package 元素来分类与管理 workflow。Package 元素除了可包含 0 至多个 Process 元素、0 至多个 Resource 元素、0 至多个 DataType 元素、0 至多个 RelevantData 元素外,也可以导入 (Import) 外部的 Package (External Package) 让内部的流程使用或是参考到其它 Package 的流程与资源。另外,除了已经定义好的 6 个基本数据类型 (String、Boolean、DateTime、Float、Integer 与 Reference) 外,用户也可以通过 DataType 元素来定义新的数据类型。采用 Package 元素来定义共同使用的 Resource、DataType 与 RelevantData,可以避免这些信息的重复定义。

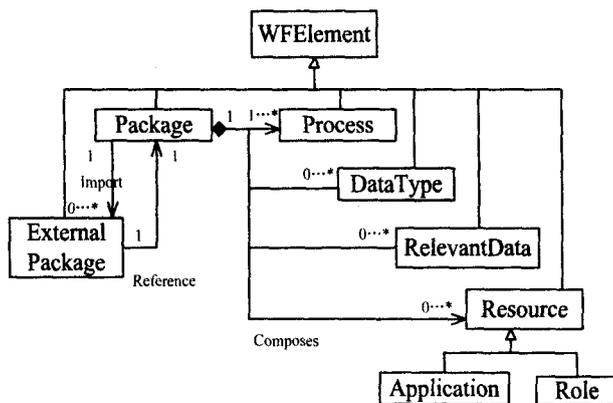


图 2 MDAWF 与 Package 有关的部分

1.2 模型元素属性说明

对于模型中的各个元素详细地定义了每个元素的

属性,如表 1 所示。

以 Process 元素为例,除了包含继承自 WFElement 元素的属性外,还有管理与模拟类型的信息属性。管理类型的信息包含了版本控制(version control)与执行时期(run-time)的管理属性。版本控制的属性有版本号(version)、制定日期(creationDate)、作者(author)与出版状态(publicationStatus)。执行时期的管理属性有预设执行优先权(priority)、预设限制执行时间(limit)、有效起始时间(validFromDate)、有效结束时间(validToDate)以及流程负责角色(responsibles)。Process 元素模拟信息的设计目的在于设定内部工作的预设模拟数据,包含了预设前置准备时间(waitingTime)、预设运作时间(workingTime)、预设持续时间(duration)、预设持续时间单位(durationUnit)以及预设花费成本(cost)。Process 元素的属性除了上述两大类信息外,另外还有说明文件(documentation)、图标(icon)、存取等级(accessLevel)与参数设定(parameters)这四个属性。

表 1 模型元素及其属性

元 素	属 性
WFElement	id, name, description, extendedAttributes
Process	derived attributes, version, creationDate, author, accessLevel, publicationStatus, priority, limit, validFromDate, validToDate, parameters, responsibles, documentation, icon, duration, waitingTime, workingTime, cost, durationUnit
Task	derived attributes, implementation, execMode, flowType, priority, limit, startMode, finishMode, deadLine, performer, icon, documentation, durationUnit, duration, waitingTime, workingTime, cost
SubFlow	derived attributes, referenceId, actualParameters, executionType
GroupTask	derived attributes
Transition	derived attributes, source, destination, conditionType, expression
Resource	derived attributes
Application	derived attributes, parameters
Role	derived attributes, type
RelevantData	derived attributes, dataType, initialValue, isArray, length
Package	derived attributes, author, version, vendorcreationDate, documentation, priorityUnit, costUnit, script, conformanceClass, publicationStatus, responsibles
ExternalPackage	derived attributes, reference
DataType	derived attributes, typeRef

2 MDAWF to PSM 的模型转换

选择以 WfMC XPD L 规范中描述的元模型作为要实现的 PSM 模型,称为 WfMC PSM,并建立由 MDAWF 模型转换至 WfMC PSM 模型的转换程序。模型信息的转换方法可以采用以下几种^[10,11]:①将模型文件通过 W3C 的 XSLT 转换,即 XML 文件间的转换;②将转换逻辑以 script 语言撰写,即模型的转换是通过执行 script 来转换;③以模板的方式进行转换,模板方式常用于将模型信息转化为代码。笔者采用了 JavaScript 撰写 PIM 到 PSM 以及 PSM 到程序代码的转换规则,

具体实现中利用 BSF (Bean Script Framework) 环境搭配 JavaScript 语言来进行转换。JavaScript 语言作为一种直译式语言与编译式语言相比更有弹性,由于不需要经过编译,更能满足快速反应的需求。转换过程分为两部分:第一部分是建立走访与转换 MDAWF 的程序,第二部分是建立 MDAWF 模型与 WfMC PSM 模型对应元素的转换程序。

2.1 MDAWF 模型的走访与转换

在 MDAWF 模型的转换程序中以递归的方式来走访模型中的元素,并根据走访到的元素类型做对应的转换,步骤如下:

- (1)取得目前走访到的 MDAWF 元素对象。
- (2)判别该元素类型。若无法识别则抛出 UnknownTypeException,并中断转换。
- (3)按照元素类型取得对应的转换 script。若无法取得 script 时,则抛出 NoScriptException,并中断转换。
- (4)将转换所需信息传入 ScriptEngine。
- (5)执行 Script 以取得对应的 PSM 元素对象。若转换过程中发生错误或是例外时,则抛出 TransformationException,并中断转换。
- (6)检验目前走访到的 MDAWF 元素对象是否有其它相关的元素对象。若有的话,则重复步骤(1)~(5)做相关元素对象的转换,并将相关元素转换出的对象按照类型加入转换结果;若没有的话,则跳至步骤(7)。
- (7)回传转换出的 WfMC PSM 元素对象。

2.2 模型对应元素的转换

表 2 是 MDAWF 与 WfMC PSM 模型元素的对应关系表。

表 2 MDAWF 与 WfMC PSM 对应关系

MDAWF 元素	WfMC PSM 元素
Package	Package
ExternalPackage	ExternalPackage
DataType	TypeDeclaration
Process	WorkflowProcess
Task	Activity which contains Tool/Route implementation attribute
SubFlow	Activity which contains SubFlow implementation attribute
GroupTask	BlockActivity
Role	Participant
Application	Application
RelevantData	DataField
Transition	Transition

以 Package 元素转换至 WfMC PSM Package 元素为例,转换的步骤如下:

- (1)从 ScriptEngine 中取得 MDAWF PIM 的 Package 元素对象。
- (2)产生一个 WfMC PSM 的 Package 元素对象。
- (3)执行元素间对应属性的转换:
 - ①建立 PSM Package 元素中的 PackageHeader 属

性。将 PIM Package 元素的 version、vendor、creationDate、description、documentation、priorityUnit 与 costUnit 属性依次对应至 PackageHeader 属性中的 XPDLVersion、Vendor、Created、Description、Documentation、PriorityUnit 与 CostUnit 属性。

②建立 PSM Package 元素中的 RedefinableHeader 属性。将 PIM Package 元素的 author、responsibles 与 publicationStatus 属性对应至 RedefinableHeader 属性中的 Author、Responsibles 与 PublicationStatus 属性。

③建立 PSM Package 元素中的 Script 属性。将 PIM Package 元素的 script 属性对应至 PSM Package 的 Script 属性。

④建立 PSM Package 元素中的 ConformanceClass 属性。将 PIM Package 元素的 conformanceClass 属性对应至 PSM Package 的 ConformanceClass 属性。

(4) 回传 PSM Package 元素对象。

3 PSM to XPDL 的模型转换

为了由 WfMC PSM 产生 XPDL 工作流程定义文件,对每个 PSM 元素定义了流程定义模板,模板中包含了静态与动态两种信息,静态信息为不会改变的描述,而动态信息为在执行时期时会按照元素属性值来置换的描述。在模板中,以“<%”为起始且以“%>”为结束符号表示为动态信息,例如 Activity 元素的模板如下所示:

```

<Activity Id="<% = attribute( " Id" ) % >"
Name="<% = attribute( " Name" ) % >">
<Description><% = attribute( " Description" ) % ></Description
>
<Limit><% = attribute( " Limit" ) % ></Limit>
<% association( " Choice" ) % >
<Performer><% = attribute( " Performer" ) % ></Performer>
<StartMode><% = attribute( " StartMode" ) % ></StartMode>
<FinishMode><% = attribute( " FinishMode" ) % ></Finish-
Mode>
<Priority><% = attribute( " Priority" ) % ></Priority>
<% association( " DeadLines" ) % >
<% association( " SimulationInformation" ) % >
<Icon><% = attribute( " Icon" ) % ></Icon>
<Documentation><% = attribute( " Documentation" ) % ></
Documentation>
<% association( " TransitionRestrictions" ) % >
<% association( " ExtendedAttributes" ) % >
</Activity>

```

WfMC PSM 产生 XPDL 工作流程定义文件的转换逻辑为:走访 PSM 的每个元素,并按照走访元素的类型来读取其定义模板,并将模板中动态部分的信息以元素对应的属性值置换后输出。当模型走访完毕后,

即会产生对应的 XPDL 工作流程定义文件,其转换步骤则如下:

(1)取得转换的 WfMC PSM 元素对象。

(2)判定元素对象的类型。若无法判别时,则抛出 UnknownTypeException,并中断转换。

(3)取得元素对象的定义模板。若无对应的模板文件时,则抛出 NoTemplateException,并中断转换。

(4)解析模型元素的模板文件。

①找出文件中所有动态部分信息。在模板文件中,以<%... %>的方式来描述动态部分的信息。

②处理找出的动态部分信息。动态部分的信息可以分成两种,一种为<% = attribute(name) % >,其处理方式将这部分的描述以目前走访到元素的某个属性值来置换,例如遇到<% = attribute(" Id") % >的描述,则会以目前转换元素的 Id 属性来置换。另一种是<% association(name) % >,其处理方式则是将目前元素的某个属性值做模板的转换,即以该属性值为转换元素,然后重复步骤(2)到步骤(4)。

(5)回传解析结果。

4 应用实例

对基于 MDA 的工作流程开发定义了两种角色,一种为工作流程开发人员,其职责在于建立与验证工作流程的 PIM 与 PSM 模型;另一种为模型转换规则的开发人员,其职责为建立工作流程 PIM 与 PSM 模型的 meta-model 以及定义工作流程 PIM to PSM 与 PSM to Code 的转换程序。首先以模型转换规则开发人员的角度在 Java 平台下设计了建模与转换系统,再从流程开发人员的角度开发了一个电子订单工作流程验证文中提出架构的可行性。设计时采用了 JMI(Java Meta-data Interface)^[12],并以 WfMC XPDL Compliant 的工作流程平台作为流程的执行环境。JMI 是 Sun 于 2002 年提出的一个 API,用于 Java 平台的应用程序存取与交换 meta-data 的信息,由于 OMG 的 MOF 与 XMI 标准已经规范了 meta-data 的定义与存储格式,JMI 即按照这些标准来设计其 API 的架构。

E-Order 是一个处理电子订单的工作流程,由处理订单的 EOrder 主流程、检验客户信用信息的 CreditCheck 子流程、处理成交货的 FillOrder 子流程组成。首先采用笔者设计的 MDAWF 对 E-Order 工作流程进行 PIM 建模。建立一个名为“sample workflow process”的 Package,并在其中建立三个 Process 分别代表 EOrder 主流程、FillOrder 子流程与 CheckCredit 子流程,接着建立名为 DBConnection 的 Role 用于连接数据库,再申明四个流程中共用的 DataType:订单状态类型 OrderStatus、信用卡类别类型 CardType、信用信息类型 Cred-

itInfo、订单类型 Order,之后分别再对 EOrder 主流程、FillOrder 子流程与 CheckCredit 子流程进行建模。在对 E-Order 的三个流程 PIM 建模之后,采用 PIM to PSM 模型转换程序将 PIM 转换成 WfMC 的 PSM,再利用 WfMC PSM to XPD L 转换程序产生对应的 XPD L 流程定义文件,如下所示:

```
<? xml version="1.0" ? >
- <Package xmlns="http://www.wfmc.org/2002/XPD L1.0"
xmlns:xpd l="http://www.wfmc.org/2002/XPD L1.0" Id="
sample workflow package" Name="sample workflow package">
- <PackageHeader>
<XPDLVersion>0.2</XPDLVersion>
<Vendor>none</Vendor>
<Created />
<Description />
<Documentation />
<PriorityUnit />
<CostUnit />
</PackageHeader>
- <RedefinableHeader PublicationStatus="UNDER _ REVI-
SION">
<Author>WU Ling</Author>
<Version />
<Codepage />
<Countrykey />
<Responsibles />
</RedefinableHeader>
<ConformanceClass GraphConformance="NON-BLOCKED" /
>
- <Script>
<Type>javascript</Type>
<Version />
<Grammer />
</Script>
<ExternalPackages />
- <TypeDeclarations>
- <TypeDeclaration Id="DataType0" Name="OrderStatus">
- <DataTypes>
- <SchemaType>
- <xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
- <xsd:element name="Status">
- <xsd:simpleType>
- <xsd:restriction base="xsd:NMTOKEN">
<xsd:enumeration value="ValidData" />
<xsd:enumeration value="InvalidData" />
<xsd:enumeration value="Accept" />
<xsd:enumeration value="BadCredit" />
```

```
<xsd:enumeration value="OverLimit" />
<xsd:enumeration value="BadDataFormat" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:schema>
</SchemaType>
</DataTypes>
<Description />
<ExtendedAttributes />
</TypeDeclaration>
- <TypeDeclaration Id="DataType1" Name="CardType">
- <DataTypes>
<ExternalReference xref="cardType" location="http://
wfmc.org/standards/docs/xpd l_sample/orderschema.xsd"
namespace="orderschema/Order" />
</DataTypes>
<Description />
<ExtendedAttributes />
</TypeDeclaration>
- <TypeDeclaration Id="DataType2" Name="CreditInfo">
- <DataTypes>
<ExternalReference xref="CreditInfo"
location="http://wfmc.org/standards/docs/xpd l _ sample/
creditService.wsdl" namespace="" />
</DataTypes>
<Description />
<ExtendedAttributes />
</TypeDeclaration>
- <TypeDeclaration Id="DataType3" Name="Order">
- <DataTypes>
<ExternalReference xref="" location="http://wfmc.org/
standards/docs/xpd l_sample/orderschema.xsd" namespace="" />
</DataTypes>
<Description />
<ExtendedAttributes />
</TypeDeclaration>
</TypeDeclarations>
- <Participants>
- <Participant Id="Role0" Name="DBCConnection">
<ParticipantType Type="SYSTEM" />
<Description />
<ExtendedAttributes />
</Participant>
</Participants>
<Applications />
<DataFields />
<ExtendedAttributes />
<WorkflowProcesses>
...
</WorkflowProcesses>
```

</Package>

5 结束语

文中将 MDA 的开发思想导入 workflow 系统的开发中来,提出基于 MDA 的 workflow 管理系统开发架构。基本上,流程的 PIM 模型可视为流程模型与 workflow 实现平台模型间的中介模型,它具体化了流程模型中的抽象概念,同时也省略了不同实现平台间的差异化特征。

在目前的研究中仅提出了初步的架构来验证 MDA 开发 workflow 的可行性,在以后的研究中希望从以下几点改进:

①在 workflow PIM 的设计方面,由于仅参考了 WfMC 的参考模型,难免会有遗漏或不足之处,希望能改善 workflow PIM 模型的设计,使其更能符合不同领域对企业流程建模的需求。

②针对模型转换部分,目前仅以 BSF 的环境撰写模型转换程序的 script,在转换程序的撰写上感觉缺乏弹性且不够方便,希望能开发流程模型转换的框架与工具集来辅助模型转换规则的开发人员建立 workflow PIM to PSM 与 PSM to PIM 的转换程序。

③建立 workflow 模型的验证机制与程序来协助流程开发人员验证 workflow PIM 模型以及转换出的 PSM 模型的正确性。

希望通过以上的改进,改善目前 MDA 的 workflow 架构的设计,更加贴近企业在工作流程上的需求。

参考文献:

[1] Hollingsworth D. The Workflow Reference Mode[EB/OL].

(上接第 88 页)

参考文献:

[1] 王仁芳. 点模型数字几何处理若干技术研究[D]. 杭州: 浙江大学,2007.

[2] 刘立国. 点云模型的光顺去噪研究[D]. 杭州: 浙江大学, 2007.

[3] 刘大峰,廖文和,戴 宁,等. 散乱点云去噪算法的研究与实现[J]. 东南大学学报(自然科学版), 2007(6):1108-1112.

[4] Pauly M, Mitra N J, Guibas L J. Uncertainty and variability in point cloud surface data[C]//Eurographics Symposium on Point Based Graphics. Zurich:[s. n.], 2004:77-84.

[5] 戴静兰. 海量点云预处理算法研究[D]. 杭州: 浙江大学, 2006.

[6] 张 毅,刘旭敏,隋 颖,等. 基于 K-近邻点云去噪算法的研究与改进[J]. 计算机应用, 2009(4):1011-1014.

[7] Pauly M, Kobbelt L, Gross M. Multiresolution modeling of

1995 - 01 - 19. <http://www.wfmc.org/standards/docs/tc003v11.pdf>.

[2] OMG. Model Driven Architecture (MDA) [EB/OL]. 2010-06-02. <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf>.

[3] 汪文元,沙基昌. 基于 MDA 的 workflow 管理系统开发研究[J]. 微型电脑应用, 2005, 21(11):29-31.

[4] 赵 文,袁崇义,张世琨,等. 一种模型驱动的 workflow 过程定义途径[J]. 计算机科学, 2006, 33(12):10-15.

[5] 魏佳璇. MDA 在科学 workflow 建模中的应用研究[D]. 兰州:兰州大学, 2009.

[6] 杨 萍,曹 健. 基于 MDA 的 workflow 门户整体风格的配置[J]. 信息技术,2007(11):50-55.

[7] 魏 歌. 基于 MDA 的 workflow 建模框架[J]. 软件导刊, 2009(10):49-50.

[8] 胡 健. 基于 workflow 的电信增值业务管理系统的建模及实现[D]. 北京:北京邮电大学, 2009.

[9] XML Process Definition Language Specification [EB/OL]. 2008-10-10. <http://www.wfmc.org/Download-document/WFMC-TC-1025-Oct-10-08-A-Final-XPDL-2.1-Specification.html>.

[10] Milicev D. Automatic Model Transformations Using Extended UML Object Diagrams in Modeling Environment[J]. IEEE Transactions on Software Engineering, 2002, 28(4):413-431.

[11] Oldevik J, Solberg, A, Elvesæter B, et al. Framework for model transformation and code generation[C]// IEEE Enterprise Distributed Object Computing Conference. [s. l.]:[s. n.], 2002:181-189.

[12] JMI Expert Group. Java Metadata Interface (JMI) Specification[EB/OL]. 2002-06-28. <http://java.sun.com/products/jmi/collateral.html>.

point-sampled geometry[R]. [s. l.]:[s. n.], 2002.

[8] Diebel J R, Thrun S, Bruning M. A Bayesian Method for Probable Surface Reconstruction and Decimation[J]. ACM Transactions on Graphics, 2006,25(1):39-59.

[9] Alexa M, Behr J, Cohen-or D, et al. Point set surfaces[C]// Proceedings of IEEE Visualization. San Diego, California:[s. n.], 2001: 21-28.

[10] 张学工. 关于统计学习理论与支持向量机 [J]. 自动化学报,2000(1):32-42.

[11] 李 雷,鲁延玲,周蒙蒙,等. 基于核方法的一种新的模糊支持向量机[J]. 计算机技术与发展,2010,20(2):9-11.

[12] 张 苗,张德贤. 多类支持向量机文本分类方法[J]. 计算机技术与发展,2008,18(3):139-141.

[13] 赵越超,李忠科,王 勇,等. 基于 OpenGL 的三维牙颌模型可视化研究[J]. 计算机技术与发展,2008,18(1):119-121.