

# 语义 Web 语言 DAML+OIL 及其应用初探<sup>\*</sup>

尹奇麟 李善平

(浙江大学 CAD&CG 国家重点实验室 杭州310027) (浙江大学计算机科学与工程系 杭州310027)

## Semantic Web Language DAML+OIL and its Initiatory Application

YIN Qi-Wei LI Shan-Ping

(State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027)

(College of Computer Science, Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027)

**Abstract** It's not easy for traditional standards such as STEP to describe product information in a semantic way, and these limitations result in difficulties in knowledge-based product information integration. In addition, contemporary computing environment needs a Web-based language to describe product information, while STEP standard and its EXPRESS language have no such features however. The development of Semantic Web is making itself be an important method to facilitate the knowledge-based information integration. This paper presents a new semantic Web language, DAML+OIL, and tries to use it to describe product information. It is proved that using DAML+OIL to describe product information is quite a feasible way.

**Keywords** Semantic Web, DAML+OIL, Product information, STEP, Ontology

## 1. 引言

### 1.1 产品信息集成的现状和面临的局限

产品信息描述格式的不统一使得传统 CIMS 之间产品信息不能共享,数据交换产生障碍<sup>[1]</sup>。STEP 标准及其 EXPRESS 语言<sup>[1,2]</sup>统一了产品信息的建模标准和数据结构,使得产品信息由原来的机器不可读(machine-unreadable)变为机器可读(machine-readable),从而在数据格式上解决了产品信息的共享和交换问题。

但是,分布、开放、动态的计算环境,使得企业内部及企业间异构计算环境下实现产品知识交换和共享的问题变得越来越迫切。企业不仅要求在数据格式上实现产品信息的互操作能力,还要在知识上实现产品信息的共享和交换,这便要求将产品信息的共享层次由数据共享提升为知识共享,即要求产品信息不仅是机器可读的(machine-readable)还要是机器可理解的(machine-understandable)。这就要求从语义上而不仅是语法上(结构上)来描述产品信息。STEP 标准及其 EXPRESS 语言并没有提供基于语义的描述方法<sup>[3]</sup>。

另外 Web 技术为信息的广泛传播和共享提供了一种非常优秀的手段,采用基于 Web 标准的语言来描述产品信息将能更好地适应当今 Web 计算环境,更加有效地发挥 Web 技术的优势来实现产品信息的集成。而基于 STEP 标准的 EXPRESS 语言也没有这样的特性。

### 1.2 语义 Web 的目标和发展

Web 技术取得成功的最重要因素之一就是建立了一套广泛的标准,使其保证了在不同层次上的互操作能力<sup>[4]</sup>。例如 TCP/IP、HTTP、FTP 等协议保证了不同信息资源在网络上的通信,而在此之上的 HTML、XML 等 Web 语言在语法上

统一了网上信息的表示形式和数据结构。

但是,仅依靠一种统一 Web 语言从语法上来实现信息的交换和共享还是不够的。比如两个应用系统都采用 XML 作为信息交换的格式,但怎么能保证它们表述的是同一个意思?这就要求计算机不但能读取数据,还要能理解和区分数据的语义。

要在语义层次上实现信息的互操作性,需要对信息的意思有一个一致的共同的理解。World Wide Web Consortium (W3C)已经认识到与“机器可读”相比,“机器可理解”的必要性和重要性,于是提出了语义 Web<sup>[5]</sup>的概念。语义 Web 的思想就是利用元数据(metadata)对 WWW 上站点和网页的内容进行描述,从而使计算机能够利用这些元数据对这些站点和网页的内容进行理解。并且,语义 Web 语言对信息资源进行描述采用了本体(Ontology)的思想。Ontology 被视为是建立智能系统(或狭义地,知识表示系统)的世界观,是对应用领域概念化的说明<sup>[6]</sup>。Ontology 为某领域提供了的一个共享的普通的理解,从而无论使人还是应用系统之间都能够有效地进行语义上的理解和通信。

语义 Web 语言可以看作是在 Web 上对信息资源的 Ontology 进行定义和描述的语言,并能够用其所定义的 Ontology 对具体信息资源进行描述<sup>[7]</sup>。语义 Web 语言的发展从最初的 RDF(S)<sup>[8,9]</sup>、OIL<sup>[10]</sup>、DAML<sup>[11]</sup>到现在已提交给 W3C 讨论的 DAML+OIL<sup>[12,13]</sup>,其语法上都遵循 XML<sup>[14]</sup>以及 XML Schema<sup>[15,16]</sup>,并利用 XML 名字空间<sup>[17]</sup>机制提供一种通用方法来定义元数据(包括关系属性),从而对信息资源进行语义上(Ontology)的描述。

## 2. DAML+OIL 语言

### 2.1 OIL 和 DAML 各自的发展及结果

<sup>\*</sup> 本项目研究工作得到国家自然科学基金的资助(项目编号60174053)。尹奇麟 硕士研究生,主要研究领域为人工智能、Web 技术、信息集成。李善平 教授,博士生导师,主要研究领域 CIMS、信息集成、嵌入式操作系统。

作为一种语义 Web 语言的 OIL (Ontology Interface Layer) 是 On-To-Knowledge 计划的产物<sup>[10,18]</sup>。On-To-Knowledge 计划是欧洲的几所大学、研究机构和公司于1999年发起的,其目标是支持有效的基于知识的管理,注重于对网上弱结构化(weakly-structured)信息资源的知识获取、表示以及访问。OIL 的指导委员会则是由来自美国和欧洲的几所大学和研究机构共同组成。OIL 的实现基础来自于三个方面:描述逻辑(Description Logics),提供正规语义和推理支持;基于框架的系统(Frame-based systems),提供认识论上的建模原语;基于 XML 和 RDF(S)语法的 Web 标准。

2000年8月,美国 DARPA 启动了一个为期六年的计划,目的是发展一系列技术使软件自治体(software agent)能够对信息资源进行动态的确认和理解,并为自治体之间提供基于语义上的互操作能力<sup>[9]</sup>。DAML (DARPA Agent Markup Language)就是这个计划第一阶段所创建的一种语义 Web 语言,它允许用户在其数据上标记语义信息,从而使计算机能对所标注的信息资源进行“理解”。

2000年12月,美国和欧洲两个组织成立联合委员会将 DAML 和 OIL 合并,命名为 DAML+OIL,并已经提交给 W3C 讨论,以使其成为语义 Web 的标准描述语言<sup>[12,13]</sup>。DAML+OIL 是在 W3C 早期的标准如 RDF<sup>[3]</sup>和 RDF Schema<sup>[9]</sup>基础上建立起来的,并且用丰富的建模原语对它们进行了扩展。

## 2.2 DAML+OIL 语言

DAML+OIL 知识基础是 RDF 三元组的集合。DAML+OIL 使用自己的词汇给 RDF 三元组以具体的意思表述。DAML+OIL 将整个世界划分为两个不相交的部分。一部分是由属于 XML Schema 数据类型(datatype)的值所组成的,称作数据类型域。另一部分则是由(单个)对象所组成的,这些对象应被看作是 DAML+OIL (或 RDF)中所定义的类的成员,此部分称作对象域。

2.2.1 DAML+OIL 的类(class) 为了描述现实中的具体对象,就需要对各种类型的对象进行抽象,这就是类(class)。下面用一个“Animal”类的例子来说明 DAML+OIL 中类的定义:

```
<daml:Class rdf:ID="Animal">
  <rdfs:label>Animal</rdfs:label>
  <rdfs:comment>
    This is an "Animal" class.
  </rdfs:comment>
</daml:Class>
```

(daml:Class)与</daml:Class>之间就是类的定义,用 rdf:ID="Animal" 来声明类的名字是"Animal"。

下面再用 rdfs:subClassOf 来定义"Animal"类的一个子类"Male":

```
<daml:Class rdf:ID="Male">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>
```

那么"Female"类就可这样定义:

```
<daml:Class rdf:ID="Female">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <daml:disjointWith rdf:resource="#Male"/>
</daml:Class>
```

其中 daml:disjointWith 表示"Female"类与"Male"是不相交的。

2.2.2 DAML+OIL 的属性(Property) DAML+OIL 的属性是一种二元关系,其连接着两个项。DAML+OIL 的属性有两种:一种描述对象与对象之间的关系,称为 daml:Ob-

jectProperty;另一种描述对象与数据类型值之间的关系,称为 daml:DatatypeProperty。下面定义一个描述对象与对象之间关系的属性"hasParent":

```
<daml:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</daml:ObjectProperty>
```

这里引用了 RDFS 中定义的"domain"和"range",如果 A hasParent B,那么 rdfs:domain 用来限定 A 所属的类,rdfs:range 用来限定 B 所属的类(在这里都为"Animal"类,说明一个 Animal 其 parent 也要是 Animal)。

接下来同样地可用 rdfs:subPropertyOf 来定义"hasParent"的一个子属性"hasFather":

```
<daml:ObjectProperty rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</daml:ObjectProperty>
```

下面定义一个另一种类型的属性,其描述的是对象与数据类型值之间的关系:

```
<daml:DatatypeProperty rdf:ID="shoesize">
  <rdfs:comment>
    shoesize is a DatatypeProperty whose range is xsd:decimal.
    shoesize is also a UniqueProperty (can only have one shoesize)
  </rdfs:comment>
  <rdfs:type rdf:resource="http://www.w3.org/2001/10/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#decimal"/>
</daml:DatatypeProperty>
```

数据类型属性"shoesize"的"range"类型是十进制数,这是通过引用 XML Schema 的数据类型"decimal"来实现的。另外 rdfs:type 引用 DAML+OIL 定义的"UniqueProperty"来说明"shoesize"的属性类型是唯一的,即一个对象只能有一个鞋号值。

2.2.3 DAML+OIL 的属性约束(property restrictions) 一个已被定义的属性不能单独使用,它只能存在于类的定义中,从而对类进行约束。例如以下一个"Person"类的定义:

```
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasParent"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

在<rdfs:subClassOf>和</rdfs:subClassOf>之间的超类是一个匿名类,且<daml:Restriction>和</daml:Restriction>之间的内容就是对这个匿名超类的约束(限制条件),表示这个匿名类有属性"hasParent",并且 Parent 本身要是"Person"类。

2.2.4 DAML+OIL 类的扩展 DAML+OIL 的一个优点就是可扩展性非常好,如对刚才定义的"Person"类进行扩展:

```
<daml:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#shoesize"/>
      <daml:minCardinality>1</daml:minCardinality>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

这样 Person 类又增加了"shoesize"属性,并且规定其最小值为1。如果所要扩展的类不在本地,只要在要扩展的类名前加上相应的 URI 地址即可。

2.2.5 DAML+OIL 个体(individual)的定义 对于属

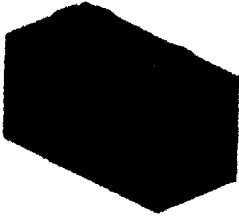
于某个类的对象, DAML+OIL 可将其定义为一个个体。比如, 定义一个叫做“Adam”的人的个体:

```
<Person rdf:ID="Adam">
  <rdfs:label>Adam</rdfs:label>
  <rdfs:comment>Adam is a person.</rdfs:comment>
  <age><xsd:integer rdf:value="13"/></age>
  <shoesize><xsd:decimal rdf:value="9.5"/></shoesize>
</Person>
```

这样便用“Person”类定义了一个叫做“Adam”的人, 其年龄是13, 鞋号是9.5。属性“age”的定义前面没有具体给出, 其和“shoesize”的定义是类似的。

### 3. 运用 DAML+OIL 描述产品信息

#### 3.1 运用 DAML+OIL 对下图所示部件进行描述



首先定义其所属的类:

```
<daml:Class rdf:ID="TechnicBrick-1x2">
  <daml:subClassOf rdf:resource="#LEGOTechnicBrick" />
  <daml:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#engpart; #representation" />
      <daml:hasValue>
        <LDrawRepresentation>
          <lDrawPartNumber>3700</lDrawPartNumber>
        </LDrawRepresentation>
      </daml:hasValue>
    </daml:Restriction>
  </daml:subClassOf>
</daml:Class>
```

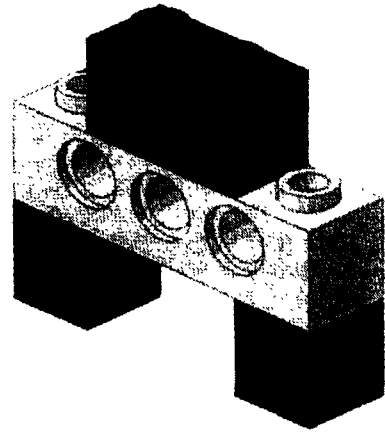
其中“LEGOTechnicBrick”为“TechnicBrick-1x2”所属的父类。

然后就可以用这个类将如图所示的部件定义为个体, 名为“RedBrick1”:

```
<legos:TechnicBrick-1x2 rdf:ID="#RedBrick1">
  <engpart:pose>
    <frames:Pose3D>
      <frames:position>
        <frames:Ground3DCoordinate>
          <frames:x>
            <frames:DistanceMeasure>
              <frames:measure>
                <legos:LDrawUnit rdf:value="0" />
              </frames:measure>
            </frames:DistanceMeasure>
          </frames:x>
          <frames:y>
            <frames:DistanceMeasure>
              <frames:measure>
                <legos:LDrawUnit rdf:value="-48" />
              </frames:measure>
            </frames:DistanceMeasure>
          </frames:y>
          <frames:z>
            <frames:DistanceMeasure>
              <frames:measure>
                <legos:LDrawUnit rdf:value="0" />
              </frames:measure>
            </frames:DistanceMeasure>
          </frames:z>
        </frames:Ground3DCoordinate>
      </frames:position>
      <frames:orientation rdf:resource="#frames; #OrientFront" />
    </frames:Pose3D>
  </engpart:pose>
</legos:TechnicBrick-1x2>
```

这里“TechnicBrick-1x2”类是在本体“legos”中定义的。

#### 3.2 对如下图所示的一个装配体(assembly)进行描述



```
<assembly:Assembly rdf:ID="#Assembly01">
  <assembly:component rdf:resource="#RedBrick1" />
  <assembly:component rdf:resource="#YellowBrick1" />
  <assembly:subassembly rdf:resource="#SubAssembly" />
</assembly:Assembly>
<assembly:Assembly rdf:ID="#SubAssembly">
  <assembly:component rdf:resource="#BlueBrick1" />
  <assembly:component rdf:resource="#BlueBrick2" />
</assembly:Assembly>
```

其中, 本体 assembly 中定义的“Assembly”类用来说明所描述对象是一个装配体, 名字叫做“Assembly01”。而 assembly:component 用来说明某个资源是该装配的一个组成部分。个体“YellowBrick1”、“BlueBrick1”和“BlueBrick2”的定义与“RedBrick1”类似。“SubAssembly”也是“Assembly01”的组成部分, 其本身也是一个装配体, 是由“BlueBrick1”和“BlueBrick2”组成的。

**结论** 在知识级上实现产品信息集成需要对产品信息进行语义上的描述, 并且当今主流计算环境需要基于 Web 的语言来描述产品信息。而传统的产品信息建模标准(STEP)并没有提供对实现上述特性的支持。

DAML+OIL 是在 Web 标准上采用 Ontology 思想建立起来的一种语义 Web 语言, 事实证明, 它是一种描述产品信息的非常可行的方法。

下一步的工作将进一步探讨运用 DAML+OIL 从语义上描述产品信息的原理与方法, 建立基于本体论与语义 Web 的产品信息基础结构。

### 参考文献

- 李善平, 何志均. 基于开放式标准化思想的 CIMS 信息集成方法. 自动化学报, 1995, 21(4): 440~447
- ISO 10303-11: Industrial automation systems and integration - Product data representation and exchange - Description methods: The EXPRESS language reference manual [S], 1994
- Metzger F J. Zurich I-E. The Challenge of Capturing the Semantics of STEP Data Models Precisely. In: Proc. of PDTAG/PAKM Workshop on Product Knowledge Sharing for Integrated Enterprises, PAKM'96 Proceedings, Basel, Oct. 1996
- Decker S, et al. The Semantic Web-on the respective Roles of XML and RDF. July, 2000. <http://www.ontoknowledge.org/oil/downl/IEEE00.pdf>
- Berners-Lee T. Weaving the Web. Orion Business Books, 1999
- Guarino N. Formal Ontology and Information Systems. In: Proc. of FOIS'98, Trento, Italy, June 6-8, 1998. Amsterdam, IOS Press, 1998. 3~15
- Heflin J, Volz R, Dale J. Requirements for a Web Ontology Language. W3C Working Draft 07, March 2002. <http://www.w3.org/TR/2002/WD-webont-req-20020307>

(下转第157页)

活操作 C 的前提条件是操作 A 或 B 已经完成;And-Split 表示完成操作 A 后,可以激活操作 B 和 C;Or-Split 表示完成操作 A 后,可以同时激活操作 B 或 C。

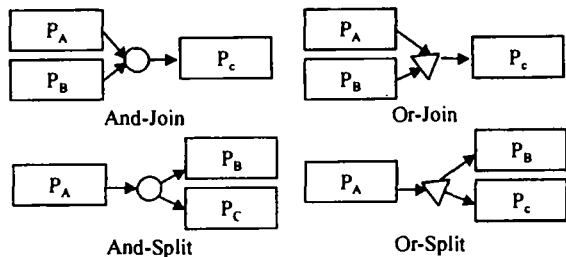


图7 操作之间约束的表示方法

需注意的是,各种操作的协同,以及规则的确定,是基于

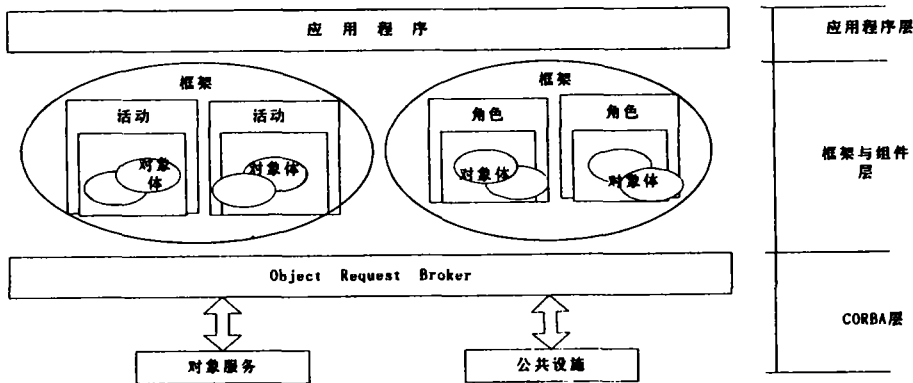


图8 基于角色模型组件的框架结构体系

这样,在某业务实施过程中,基于角色模型组件框架体系,以支持 BPR 实施.通过业务规则,触发活动跳转,快速配置角色,来完成最终完成业务过程的重组(图8示)。

**结束语** BPR 的理念,表述了“对经营过程的彻底的重新构思、根本的重新设计以达到在一些诸如成本、质量和速度等关键性能方面的显著提高”(海默·潜培语)。本文试图从更广泛的意义上抽象出角色模型的概念,根据业务过程中的一系列逻辑相关的活动步骤,通过角色有效地执行它们以达到预定的业务结果.特别是在业务过程变化时,能够基于角色模型,快速实现重组.例如,对一个生产型项目,活动实施角色可能是人员、组织,甚至设备(如各种机床设备)。当加工工艺过程需要变化时,就可以通过角色模型—机床设备转换,实现加工工艺过程的重组.这在 CIMS 系统中,无疑极具现实意义。

在活动实例中的角色前驱分量  $R_i$  和角色实例中活动前驱分量  $A_i$  制定的.在称之为里程碑的 Key Point 上的  $R_i$ 、 $A_i$  的状态是发生活动跳转的触发源.亦即,当关键点  $R_i$ 、 $A_i$  输出结果显示与设计状态发生异常,则转而执行另一个有效规则,来实现活动跳转.同时,根据活动的性质,配置新的角色,从而迅速实现业务过程的再工程(参照规则的 PDL 的描述及图2表示的工作流程图)。

**4.2.2 基于角色模型的 BPR 实现** 一个业务过程,其活动和角色是相对稳定的对象,而业务规则,随着不同的业务项目,多变的市场状况而呈现出不同.为此,我们把表现实体或过程及其相互关系的相对稳定部分(比方活动、角色及其它它们之间的相互关系)与表现为规则及策略方法指导下的动态行为的易变部分(比方业务规则)进行剥离,构建基于角色模型的组件框架,如活动组件、角色组件,如图8所示。

当然,在这方面的还有许多亟待研究的工作,但有一点是肯定的,那就是利用现代信息技术,基于角色模型的概念,重新设计我们的业务过程,完全能够达到诸如成本、质量和速度等方面的极大提高。

### 参考文献

- 1 余菁编著.企业再造:重组企业的业务流程.广东经济出版社,2001.10
- 2 何克清,等.角色模型化:一个软件模式知识级的建模方法.计算机科学,2001,28(8)
- 3 Object Management Group. Common Object Request Broker: Architecture and Specification, Feb. 1998
- 4 W3C Note 18 December 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218>
- 5 Connolly D, et al. DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001. <http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218>
- 6 <http://www.w3c.org/XML/>
- 7 Thompson H S, et al. XML Schema Part 1: Structures, W3C Recommendation, 2 May 2001. <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- 8 <http://www.oasis-open.org/cover/schemas.html>
- 9 Bray T, Hollander D, Layman A. Namespaces in XML, World Wide Web Consortium 14-January-1999. <http://www.w3.org/TR/REC-xml-names>
- 10 <http://www.ontoknowledge.org/>

(上接第141页)

- 8 Lassila O, Swick R R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>
- 9 Brickley D, Guha R, et al. Resource Description Framework (RDF) Schema Specification, W3C Candidate Recommendation 27 March 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>
- 10 Horrocks I, et al. The Ontology Interchange Language OIL: [Technical Report]. Free University of Amsterdam, 2000. <http://www.ontoknowledge.org/oil/>
- 11 The DARPA Agent Markup Language Homepage. <http://www.daml.org/>
- 12 Connolly D, et al. Annotated DAML+OIL Ontology Markup,