(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102147731 A
(43)申请公布日 2011.08.10

(21)申请号 201110099316.7

(22)申请日 2011.04.20

(71)申请人 上海交通大学
地址 200240 上海市闵行区东川路 800 号

(72)发明人 王英林 郭健美 王楷翔 唐琦 郭俊

(74)专利代理机构 上海交达专利事务所 31201
代理人 王锡麟 王桂忠

(51)Int.Cl.
G06F 9/44(2006.01)

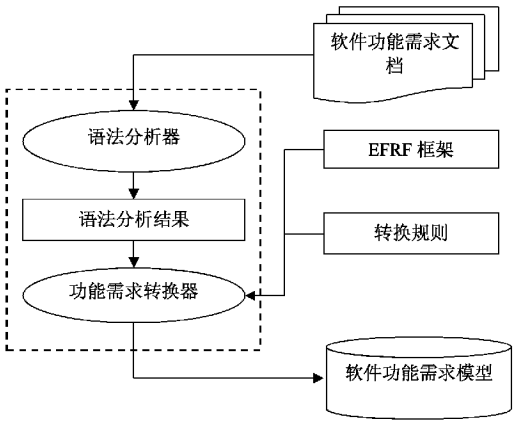
权利要求书 1 页 说明书 6 页 附图 2 页

(54)发明名称

基于扩展功能需求描述框架的功能需求自动抽取系统

(57)摘要

一种计算机应用技术领域的基于扩展功能需求描述框架的功能需求自动抽取系统，包括：语法分析模块和功能需求转换模块，语法分析模块基于语法分析器，功能需求转换模块根据预定义的转换规则将语法分析器的分析结果映射到 EFRF 上，通过两个模块的分析和转换。本发明将功能需求文档作为输入后基于语法分析器对功能需求文档进行分析，通过一系列预定义的转换规则对语法分析结果进行转换，映射到 EFRF 上，以 XML 的形式输出软件功能需求结构化表达并存储在系统中。



CN 102147731 A

1. 一种基于扩展功能需求描述框架的功能需求自动抽取系统，其特征在于，包括：语法分析模块和功能需求转换模块，其中：语法分析模块基于语法分析器，功能需求转换模块根据预定义的转换规则将语法分析器的分析结果映射到 EFRF 上，通过两个模块的分析和转换。

2. 根据权利要求 1 所述的基于扩展功能需求描述框架的功能需求自动抽取系统，其特征是，所述的 EFRF 包括 10 个功能需求的描述维度，每个功能需求都可以用一个 EFRF 中的这 10 个维度来描述。

基于扩展功能需求描述框架的功能需求自动抽取系统

技术领域

[0001] 本发明涉及一种计算机应用技术领域的装置，具体是基于扩展的功能需求描述框架 (Extended Functional Requirements Framework，EFRF) 的功能需求自动抽取系统。

背景技术

[0002] 软件功能需求描述了一个系统的外部功能。现有的软件功能需求大多以文本形式描述在文档中。由于缺乏结构化表达，这些功能需求文档很难有效利用到后续软件开发过程中。因此，如何从这些功能需求文档中自动抽取结构化的功能需求表达，有助于提高功能需求的利用率，提高软件开发的效率。

[0003] 经过对现有技术的检索发现，N. Niu 和 S. Easterbrook 人的“抽取和建模产品线功能需求”(“Extracting and modeling product line functional requirements”，发表于 2008 年在西班牙巴塞罗那召开的第 16 届需求工程国际会议集，Proceedings of 2008 International Requirements Engineering Conference，第 155-164 页）公开了一种半自动化的软件产品线功能需求抽取和建模方法。该方法在动宾关系对的基础上，提出了一个包括 6 个描述维度：施动者 (agentive)，对象 (objective)，地点 (locational)，时间 (temporal)，处理过程 (process) 和条件 (conditional) 功能需求可变性模型。缺点是：其可变性模型的定义中还存在不清晰的地方。例如，“Agentive”和“Objective”的修饰语没有列入维度；“Process”太具有一般性，包含信息不够细节化；在抽取 verb-directObject 关系对的时候，采用统计方法，结果不够准确。

[0004] Liaskos 等人的“基于目标的可变性获取和分析”(“On goal-based variability acquisition and analysis”，发表于 2006 年在美国明尼阿波利斯召开的第 14 届需求工程国际会议集，Proceedings of 2006 International Requirements Engineering Conference，第 76-85 页）定义了一个具有可变性的目标模型，该模型基于 Fillmore 提出的 Case Grammar Theory。他们关注于每个目标的 OR- 组合的语义特征，并且用三个语义维度定义了目标的上下文的可变性。文章中还强调了这三个语义维度对正交可变性模型 (Orthogonal variability models) 建模的重要性。将目标的语义描述维度作为可变点中的变量，在此基础上进行正交可变性建模能更好地体现可变性及其如何可变。

[0005] 上述提出的方法还不能实现对软件功能需求的全面的、自动的抽取。因此，目前软件功能需求的构建仍需要付出大量的人工劳动。

发明内容

[0006] 本发明针对现有技术存在的上述不足，提供一种基于扩展功能需求描述框架的功能需求自动抽取系统，将功能需求文档作为输入后基于语法分析器对功能需求文档进行分析，通过一系列预定义的转换规则对语法分析结果进行转换，映射到 EFRF 上，以 XML 的形式输出软件功能需求结构化表达并存储在系统中。

[0007] 本发明是通过以下技术方案实现的，本发明包括：语法分析模块和功能需求转换

模块，其中：语法分析模块基于语法分析器，功能需求转换模块根据预定义的转换规则将语法分析器的分析结果映射到 EFRF 上，通过两个模块的分析和转换。

[0008] 所述的 EFRF 包括 10 个功能需求的描述维度，每个功能需求都可以用一个 EFRF 中的这 10 个维度来描述。

[0009] 本发明的原理是：针对软件功能需求，其上下文可以用所定义的 EFRF 框架来描述，该框架包括 10 个描述维度，即：Agentive，功能内容的发起者；Action，功能描述的行为；Objective，功能行为的作用对象；Agentmod，功能发起者的约束；Objmod，功能作用对象的约束；Locational，功能相关的地点；Temporal，功能相关的时间，包括发生频率，持续时间等；Manner，功能实现的方式，包括工具，条件等；Goal，功能的目标；Constraint，功能实现的其他限制条件。进一步地，经观察和实现发现，上述软件功能需求的描述维度都与功能需求的语法分析结果具有关联性，可以依据一定的转换规则实现自动抽取。因此，本发明借助于语法分析器和一组定义的转换规则实现了软件功能需求的自动抽取。

[0010] 本发明有益的效果是：提出了基于 EFRF 的软件功能需求的自动抽取方法并实现了相应系统；借助于所构建的语法分析器，提高了对软件功能需求文档进行自动化的分析和处理的能力；基于所构建的 EFRF 描述框架和转换规则，语法分析器的分析结果被映射到一个多维度描述的软件功能需求的结构化模型，从而提高了软件功能需求描述的全面性和准确性。此外，本发明所构建的系统，可以分析所有符合 IEEE-STD-830 标准的软件需求文档，能大大降低在软件功能需求分析中所涉及的手工劳动，从而为企业节省了人力物力，为企业软件的大规模定制提供了帮助，在企业软件开发上具有很高的应用和商业价值。

附图说明

[0011] 图 1 是本发明的系统框架图。

[0012] 图 2 是语法结构分析结果示例。

[0013] 图 3 是语法依存分析结果示例。

[0014] 图 4 是系统最终生成的功能需求模型示例。

具体实施方式

[0015] 下面结合附图对本发明的实施例作详细说明，本实施例在以本发明技术方案为前提下进行实施，给出了详细的实施方式和具体的操作过程，但本发明的保护范围不限于下述的实施例。

[0016] 如图 1 所示，本实施例包括：语法分析模块和功能需求转换模块，其中：语法分析模块基于语法分析器，功能需求转换模块根据预定义的转换规则将语法分析器的分析结果映射到 EFRF 上，通过两个模块的分析和转换。

[0017] 所述的 EFRF 包括 10 个功能需求的描述维度，每个功能需求都可以用一个 EFRF 中的这 10 个维度来描述。

[0018] 该实施例的输入是一组现有软件系统的功能需求文档，所有文档为符合 IEEE-STD-830 标准的需求文档。所采用的实施例是由 249 个句子组成的功能需求描述，包含 5,669 个词，不含标点的单词为 5,189 个。

[0019] 系统第一个主模块是语法分析器，基于斯坦福语法分析器 Stanford Parser 和文

本工程框架平台 GATE 开发实现语法分析器，并将上述功能需求文档输入到语法分析器中进行处理。语法分析器的实现部分包括 :1) 分词与词性标注模块，对需求文本进行分词、词性标注 ;2) 语法结构树分析模块，对经过分词和词性标注的需求文本进行语法结构分析，分析出句子中的所有结构信息 ;3) 依存关系分析模块，通过这个模块，得到的是句子内各成分之间的依存关系，这里的依存关系指的是二元关系，比如直接宾语，补语等关系。

[0020] 本实施例通过将 Stanford Parser 整合到 GATE 框架下，采用了两个步骤来实现语法分析器。第一步，调用 Stanford Parser，将解析结果的依存关系加入到 GATE 下的文档中作为 Token 的属性。从 Stanford Parser 处理之后的文档中除了标注词性信息之外还标注了句子内部成分的依存关系属性。对依存关系解析的最小单元是一个句子，生成的是一个句子内部各个成分之间的依存关系。第二步，扩展 GATE 中的 NE 模块，利用分析好的语义角色，将其与功能需求本体中的实体概念相结合，利用之前定义好的一系列转换规则，将这套规则用 GATE 可识别的符合 JAPE 语法规则表达，从而实现 EFRF 实体的识别。因为在命名实体识别的时候的依据是依存关系，所以，在 JAPE 规则中很重要的一个输入就是 Dependency。这里 Lookup，Token，Dependency 都是解析之后的词的属性，将这些属性的值作为规则的判别标准。

[0021] 系统第二个主模块是功能需求转换器，基于所提出的 EFRF 框架和预定义的转换规则，对语法分析结果进行再处理，从而将其映射到软件功能需求的 EFRF 框架上。功能需求转换器的实现部分是一个 EFRF 命名实体识别模块，通过事先定义的转换规则最终实现自动分析和映射。

[0022] 本实施例通过定义有效的转换规则，将语法分析结果映射到功能需求描述框架 EFRF 下，实现了功能需求转换器。表 1 是其中的部分转换规则，本实施例对功能需求中的主动和被动语态分别进行分析，针对不同的语态建立不同的规则。表格第一列是要匹配的实体概念，与 EFRF 中的维度一一对应，表示当规则匹配了之后，该文本片段将要加注的标签名称，也就是概念名称。第二列和第三列分别是主动语态和被动语态下的转换规则。规则中出现的符号的含义如下。规则中的概念名称，如“Action”:指的是已经被其他规则识别出来并加了标注的“Action”实体。规则左括号前面的名称，如“Subj”:指的是通过自然语言处理得到的依存关系。如果两个文本片段之间的关系是 Subj，则利用这条规则进行匹配，看是否满足规则的其他条件。规则“obj(Action，×)”表示的是，当识别出了 Action 之后，而且在文本中有一个短语与 Action 之间的依存关系是 obj(包含 obj 的各个子关系)，那么这个短语就会被标记为 Objective。规则“con_j(objective，×)”表示的是，当识别出了 Objective，而且在文本中有一个短语与 Objective 之间的依存关系是 con_j(包括 con_j 的各个子关系)，那么这个短语也同样被标记为 Objective。最后根据识别出来的所有 Objective 的语序进行排序，得到一个连续的文本片段或者几个独立的短语。

[0023] 实施例的工作过程 :

[0024] 第一步，用户向该抽取系统发出如下请求 :

[0025] Teacher assistant mark students′ homework in the lab at 7′ clock.

[0026] 该请求是目标应用需要满足的一个功能需求的文本描述。

[0027] 第二步，上述文本描述通过语法分析器的词性标注模块进行分析。对上述文本需求进行分词与词性标注，输出结果如下所示 :

[0028] Teacher/NN assistant/NN mark/VBP students/NNS' /POS homework/NN in/IN the/DTlab/NN at/IN 7'clock/NN./.

[0029] 在上面结果中,每个单词紧跟的"/"后面的符号表示分析结果中该单词的词性,例如 assistant/NN 表示 assistant 的词性是名词。

[0030] 第三步,使用上述分词与词性分析结果,通过语法分析器对其进行语法结构树解析,分析出句子中的所有结构信息。例如"(NP(NN Teacher)(NN assistant))"表示 Teacher assistant 是由两个名词组成的一个名词短语 (Noun Phrase)。对此示例的语法结构树展示请见图 2。

[0031] 第四步,基于上述语法结构树信息,使用语法分析器对实施例进行语法依存关系解析,分析出句子中的直接 / 间接主语、直接 / 间接宾语、补语、修饰语等等语法上的依存关系对。例如 :dobj(mark-3, homework-6) 表示 mark 的直接宾语 (direct object) 是 homework。上述需求文本的语法依存对展示请见图 3。

[0032] 第五步,将上述获得的语法依存对输入到功能需求转换器中,利用表 1 定义的转换规则将语法依存对中识别的语法成分对应到相应地 EFRF 描述维度上。例如,nsubj(mark,assistant) 中"mark"被识别为 Action,而"assistant"被识别为 Agentive。

[0033] 表 1(? -指的是任意一个依存关系 ;* -指的是任意长度的字符 ;×-指的是当规则匹配之后需要加注标签的文本片段 ;&-指的是前后两条规则应该同时满足)

[0034]

Cases	语法依存对		x 对应的语义类型
	主动语态	被动语态	
Agentive	Subj(*, ×)	Agent(*, ×)	
Action	Subj(×, *)	Agent(×, *)	
Objective	obj(Action,×) & con_j(Objective, ×)	subjpass(*,×)& con_j(Objective,×)	
Agentmod	mod(Agentive,×)	mod(Agentive,×)	
Objmod	mod(Objective,×)	mod(Objective,×)	
Locational	prep_at(Action,×); prep_in(Action,×)	prep_at(Action,×); prep_in(Action,×)	PLACE, PLACENAME, …
Temporal	prep_at(Action,×); prep_in(Action,×); prep_for(Action, ×); tmod(Action,×) &	prep_at(Action,×); prep_in(Action,×); prep_for(Action, ×); tmod(Action,×);	TIME, DATE, DURATION, FREQUENCY, …
Manner	prep_at(Action,×); prep_in(Action,×); prep_by(Action, ×); prep_through(Action, ×)	prep_at(Action,×); prep_in(Action,×); prep_by(Action, ×); prep_through(Action, ×)	x 不属于 Locational, Temporal 中定义的词表
Goal	purpcl(Action, ×)& ?(Goal,×)	purpcl(Action, ×)& ?(Goal,×)	
Constraint	prep_if(Action, ×)& mod(Constraint, ×)	prep_if(Action, ×) & mod(Constraint, ×)	

[0035] 然而，某些语法成分可能会被识别到多个描述维度上。例如，根据表 1 定义的规则，prep_in(homework, lab) 中“lab”可能会被识别为 Locational，Temporal 或 Manner 等。为了避免冲突，进一步地判定这三个由介词短语构成的语法依存对中相应成分的语义类型。

[0036] 1) 针对 Locational 维度，定义了如下的词表（可根据需要进一步扩充）表示某个词的语义类型：

[0037] PLACE :home, lab, office, any place, bus stop, station, train station, crossroad,...

[0038] PLACENAME :Asia, Europe, Africa, Shanghai, Beijing,...

[0039] 2) 针对 Temporal 维度，定义了如下的词表：

[0040] TIME :morning, noon, afternoon, evening, night, midnight, Number' clock,...

[0041] DATE :Monday, Tuesday, Wednesday, January, February,...

[0042] DURATION :Number Days/Months/Years...

[0043] FREQUENCY :once, Number Times,...

[0044] 3) 针对 Manner 维度，采用“排除法”，即相应的介词短语所引导的语法成分不属于 Locational 和 Temporal 定义的词表中，即可判定为 Manner。

[0045] 根据上述定义的词表，语法依存对中的相应成分被划分到某个语义类型上，从而进一步判定其所属的维度。例如，在上述实施例中，prep_in(homework, lab)，由于 lab 属于 PLACE 语义类型，因而“in the lab”被判定为 Locational 维度。再如，prep_at(lab, 7’clock)，由于 7’clock 属于 TIME 语义类型，因而“at 7’clock”被判定为 Temporal 维度。

[0046] 图 4 展示了上述请求经系统处理后的最终结果。可以看到，在整个界面的右边的地方是扩展的要抽取的十种 EFRF 语义命名实体 (Action, Agentive, Agentmod, Constraint, Goal, Locational, Manner, Objective, Objmod, Temporal)。抽取出来的功能性需求由 10 个维度描述，这个 EFRF 框架反映了功能需求描述中的语义特征，因此，它能作为一个结构化框架，来帮助进一步全面地分析功能需求。

[0047] 表 2 是通过运行系统对实验数据进行分析和评价后得到的结果。

[0048] 表 2 系统抽取效果的评测结果

[0049]

	召回率	准确率	F 值
Agentive	98.5	95.1	96.8
Action	96.1	93.8	94.9
Objective	81.1	80.4	80.7
Agentmod	86.4	90.1	88.2
Objmod	80.1	84.0	82.0
Locational	35.3	39.3	37.2
Temporal	40.2	91.7	55.9
Manner	18.6	60.5	28.5
Goal	47.4	78.1	59.0
Constraints	78.2	89.7	83.5

[0050] 实验结果显示了准确率普遍比召回率高，这是由基于规则的信息抽取的特点所决定的。其中 Agentive, Action, Objective, Agentmod 和 Objmod 的抽取效果在准确率和召回率上都较高。但是对于其他的语义成分，基于规则的方法得到的召回率相对较低。但总体上，平均 F 值能达到 72.6%，说明该方法以及形成的系统是比较有效的。本实例结合 Stanford Parser，通过引入 EFRF 模型和定制好的转换规则，实现了一个可以实际应用的系统。对实验数据的测试结果表明了该系统的有效性。

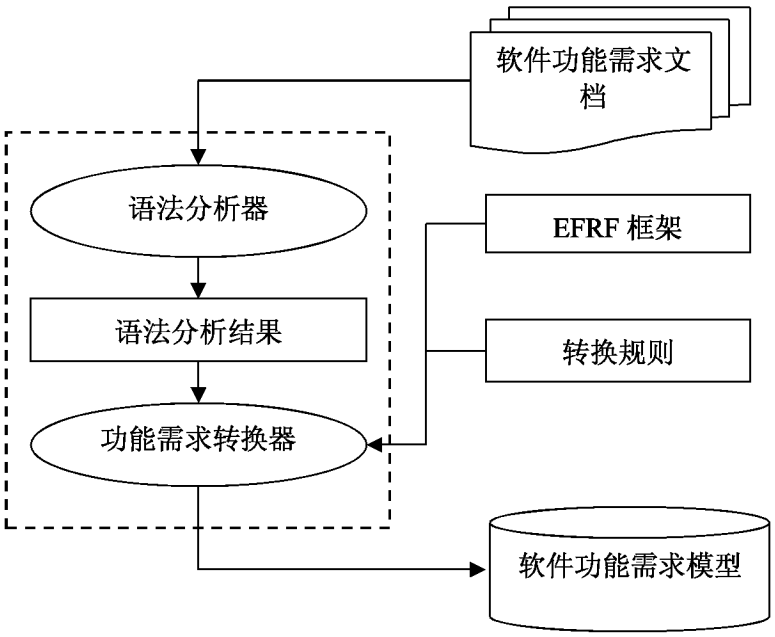


图 1

```
(ROOT
  (S
    (NP (NN Teacher) (NN assistant))
    (VP (VBP mark)
      (NP
        (NP
          (NP (NNS students) (POS '))
          (NN homework))
        (PP (IN in)
          (NP
            (NP (DT the) (NN lab))
            (PP (IN at)
              (NP
                (NP (CD 7'clock)))))))))
    (. .)))
```

图 2

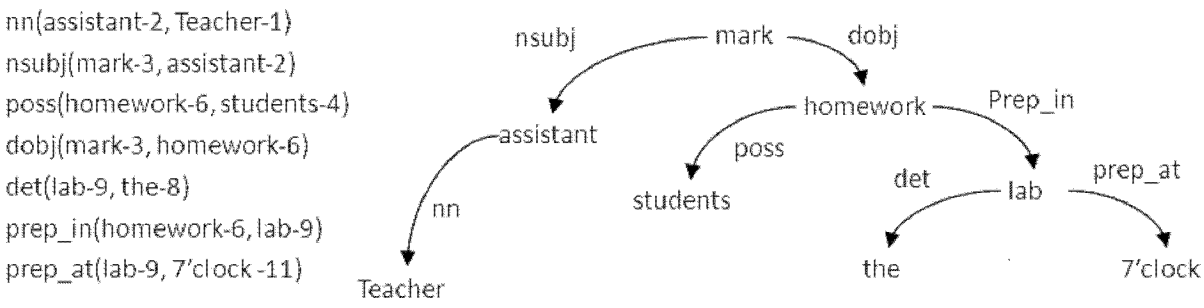


图 3

{Teacher}$_{\text{Agentmod}}$ {assistant}$_{\text{Agentive}}$ {mark}$_{\text{Action}}$ students' homework {in the lab}$_{\text{Locational}}$ {at 7'clock}$_{\text{Temporal}}$.

{Senior}$_{\text{Agentmod}}$ {student}$_{\text{Agentive}}$ do {C++}$_{\text{Objmod}}$ {homework}$_{\text{Objective}}$ {at home}$_{\text{Locational}}$ carefully {in order to get an A}$_{\text{Goal}}$.

{TA}$_{\text{Agentive}}$ {mark}$_{\text{Action}}$ {assignment}$_{\text{Objective}}$ {every Sunday}$_{\text{Temporal}}$ through {Internet}$_{\text{Manner}}$ if {Internet connection is available}$_{\text{Constraint}}$.

. . .

Agentive
Action
Objective
Agentmod
Objmod
Locational
Temporal
Manner
Goal
Constraint

图 4