

文章编号:1001-9081(2010)02-0406-05

基于语义的组件组合方法研究

曾婧娜, 余 丹, 李先军, 马世龙

(北京航空航天大学 软件开发环境国家重点实验室, 北京 100191)

(jnzeng@cse.buaa.edu.cn)

摘 要:在基于组件的一般性应用系统中,面向自然语言描述需求的组件自动组合技术的研究仍然存在很多挑战。但是,在限定的上下文中,以基于语义的方式对特定应用提供组件自动组合的方法仍然是可行的。采用关键字描述语义信息,提出了一种基于语义的组件组合方法,对组件和请求过程建模,采用规划器搜索组件序列,完成组件的自动组合。方法已经应用于航天器自动化测试数据查询应用系统中,取得了较好的应用效果。

关键词:组件组合;关键字描述;语义信息;航天器测试

中图分类号: TP311.52 **文献标志码:** A

Research on semantic-based component composition approach

ZENG Jing-na, YU Dan, LI Xian-jun, MA Shi-long

(State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China)

Abstract: In component composition-based system, the automatic composition oriented to high-level requirements is still confronted with many difficulties. Nevertheless, to support the automatic component composition based on semantic information to specific types of application in restricted context is still feasible. An approach that used keywords to describe semantic information, modeled the components and process of querying and applied a planner to search for the component sequence was proposed. The method has been used in a data query system for satellites testing and obtains favorable result.

Key words: component composition; keywords description; semantic information; satellites testing

0 引言

航天器自动化测试系统是一个典型的企业应用系统,为确保航天器工作的可靠性,在投入使用前要对其展开反复的测试工作,并对测试过程产生的数据进行查询、分析和评估,从而衡量航天器各部件的性能指标和航天器整体的可靠程度,同时跟踪航天器的测试进度。近年来,随着我国航天器研制和发射数量的增加,测试任务量加剧导致测试系统产生的数据量急剧膨胀。因此,面对用户特定的查询分析需求,如何有效地从日益增长的测试数据中提取有用信息为用户提供数据分析支持成为构建航天器自动化测试系统中的重要问题。

标准解决方案的流程是在进行详细需求分析的基础上提炼出具有普遍意义的功能,针对这些功能提供查询条件选择页面,用户可以通过特定查询条件的选择进行满足个人需求的数据查询。但是这种解决方案是复杂且费时的,并且存在如下问题:

1) 因为程序开发遵循传统流程,系统应对变更和增加需求的方式是以程序增量的方式添加,这样会随着时间的推移和需求的不断变动,导致系统结构臃肿,使得维护难以进行;当前的数据查询分析平台已经能够满足既定用户对测试任务、工作任务与工作项目等数据的查询与分析功能需求。但是,随着新业务和新角色的不断加入,系统出现新的需求,如需要测试任务与测试项目之间完成情况的对应统计信息,如果按照上述方法解决,则需要再次进行需求分析,并增加新的程序模块实现此需求,从而引起系统不同程序模块之间相似

程度增大,导致日后维护困难。

2) 系统在交付用户使用前一般需要开发人员对使用者进行培训,说明系统功能和使用方式等,用户在使用系统的过程中经常会遗忘某些功能的使用方式,只对自己常用的一些功能比较熟悉,这样无形中缩减了系统功能,没有做到对系统最大限度的使用。

本文提出一种基于语义的组件组合方法,采用具有层级结构的关键字表达语义,构建适当的关键字库、组件库,采用规划器自动搜索组件序列,构建具有良好扩展性的系统,该系统可高效地为用户的查询分析需求提供有效的应答,具有以下特点:

1) 在数据查询分析系统中使用基于语义的组件组合方法,该方法的特点决定系统具有良好扩展性,面对变动和增加的需求,可以直接通过组件的不同组合序列提供应答,或者增加缺少的组件扩大组件组合的应答范围。

2) 组件组合方法采用目标驱动的查询分析方式,用户只需要了解自己的业务术语,以关键字的方式提交给系统,如“工作项目”,“工作任务”,基于带有语义信息的组件库的设计,系统列出符合用户请求目标的组件组合应答。

1 航天器自动化测试和查询优化

航天器自动化测试数据分为索引数据和实测数据两类。索引数据由航天器测试的若干测试阶段决定,包括卫星、研制阶段、电测阶段、分系统、测试任务和工作任务等。实测数据是在航天器测试过程中产生的数据,包括待测试的内容和测

收稿日期:2009-08-21;修回日期:2009-10-01。

基金项目:软件开发环境国家重点实验室探索性自主研究课题资助项目(SKLSDE-2009ZX-13)。

作者简介:曾婧娜(1985-),女,湖北荆门人,硕士,主要研究方向:航天器自动化测试、Web应用系统体系结构;余丹(1979-),女,湖北武汉人,博士,主要研究方向:航天器自动化测试、安全苛刻性系统、可信软件;李先军(1977-),男,山西晋城人,博士,主要研究方向:航天器自动化测试、软件体系结构;马世龙(1953-),男,北京人,教授,博士生导师,主要研究方向:海量信息处理。

试结果等。测试任务是工作任务的集合,工作任务是测试的最小单元,工作任务的执行流程由其对应的工作项目预先定义,航天器自动化测试任务的业务模型如图1所示。当前航天器测试数据查询应用发展的方向是构建智能化和灵活性强的系统,加强航天器自动化建设的电子化,更好地辅助研究人员分析测试情况。近几年来,相关的研究工作给出一些优化查询的方法,但存在一些问题。

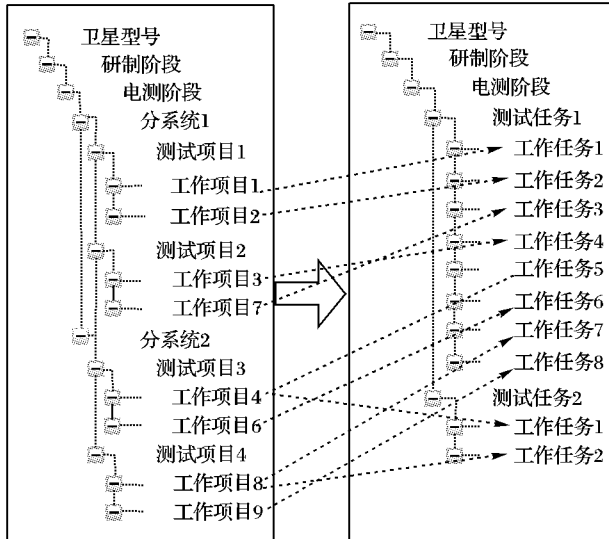


图1 航天器自动化测试任务的业务模型

文献[1]提出了一种基于权重的XML查询扩展方法。该方法首先进行查询词扩展,找到最能代表用户查询意图的权重扩展查询词,然后在扩展查询词的基础上进行结构查询扩展,最终形成完整的“内容+结构”的查询扩展表达式。该方法的特点在于扩展词包含了查询的语义信息,加入反馈环节可以不断修正扩展词的语义准确性。但由于该方法是对XML结构化文档的查询,而当前大多数待查询的原始数据不是以XML文档形式存储的。

文献[2]提出了一种用于数据聚合的灵活的数据查询和处理方式。该方法能够聚合网络上不同来源的数据,根据其建立的一套严整的标签体系理论,按照用户的要求对这些数据进行查询和处理。与此类似,参考文献[3-5]提出用于服务自动组合和企业数据总线建设的数据查询处理方法,但该类方法要求数据源以标准的RSS或Atom形式提供数据,特别适于与当前流行的Yahoo Pipes联合使用,却不适用于企业应用中对既定形式数据的查询与操作。

文献[6]提出一种利用RDF将网络上的图书信息语义化的方法。该方法可以代表当前采用语义知识提高查询的智能化程序的一类方法。RDF和Ontology技术为资源和数据提供丰富的语义信息。虽然基于语义的数据查询的逻辑基础相对比较完备,但本体的开发、合并和演化以及验证领域本体的完备性和正确性尚待进一步的研究^[7]。

2 基于语义的组件组合方法建模

结合对已有的各解决方案的优缺点分析,本文采用关键字描述的方式提供语义信息,运用灵活的组件组合方式为数据查询请求提供应答。虽然当前的航天器综合自动化测试处于研究阶段,但本实验室所在的研究小组在前期工作中对自动化测试各支持软件的建设已做出相关工作^[8]。本节将结合航天器试验与测试数据查询软件阐述基于语义的组件组合

方法的形式化建模过程,主要包括层级关键字建模、用户请求建模、组件建模、组件组合过程建模等。其中,以一个具体应用示例“查询某个工作项目对应的在某时间点后执行过的工作任务的集合”辅助说明以上不同类型的建模定义。

2.1 层级关键字建模

定义一组关键字为 $K = \{k_1, k_2, \dots, k_n\}$,即关键字集合,数据对象被一组关键字描述, $d(d) \subseteq K$ 。在航天器试验与测试数据查询系统中,被描述为 $\{\text{exeWorkTask}, \text{timeDesc}\}$ 的数据对象的含义是按时间降序排列的已执行的工作任务集合。层级关键字 K 的结构定义为一个有向无环图(Directed Acyclic Graph, DAG),图中顶点代表关键字,边代表关键字之间存在子-父关系。定义1和定义2给出层级关键字结构和关键字子-父关系的定义。

定义1 层级关键字结构。给定 $S = (K, E)$,其中 S 是层级关键字结构, K 表示关键字, $E \subseteq K \times K$ 表示关键字间子-父关系集合。

定义2 关键字子-父关系。如果关键字 $k_1 \in K$ 描述的数据对象同样可以被关键字 $k_2 \in K$ 描述,那么关键字 k_1 是关键字 k_2 的子关键字,记为 $k_1 < k_2$ 。

性质1 传递性。 $k_1 < k_2, k_2 < k_3 \Rightarrow k_1 < k_3, \forall k_1, k_2, k_3 \in K$ 。

性质2 自反性。 $k < k$ 。

性质3 反对称性。 $k_1 < k_2, k_2 < k_1 \Rightarrow k_1 = k_2$

子-父关系的定义具有关联语义的特性,举例说明,假设 k_2 是 k_1 的子关键字,那么用户容易将 k_2 描述的数据对象关联至 k_1 ,即由 k_2 描述的数据对象也一定在语义上适用于被 k_1 描述。例如,因为已执行的工作任务也是工作任务,所以能被“已执行的工作任务”描述的数据对象一定能被“工作任务”描述,那么“已执行的工作任务”是“工作任务”的子关键字,表述为 $\text{exeWorkTask} < \text{workTask}$ 。诸如此类,“时间降序”是“排序”的子关键字,表述为 $\text{timeDesc} < \text{sorted}$ 。此外,关键字的子-父关系还是一种偏序关系(引理1),可以用于组件搜索算法的剪枝策略的设计。与本章开始提出的示例相关的关键字包括:“工作任务”、“执行的工作任务”、“序列”、“工作项目”、“工作任务对应的工作项目”、“开始时间限制”,表述为 $\{\text{workTask}\}, \{\text{exeWorkTask}\}, \{\text{array}\}, \{\text{workProject}\}, \{\text{corWorkProject}\}, \{\text{StartTimeRestrict}\}$,其中“工作任务对应的工作项目”是“工作项目”的子关键字。

引理1 关键字子-父关系是偏序关系。

证明 由性质1,2,3可知,关键字子-父关系满足传递性,自反性和反对称性,所以关键字子-父关系是偏序关系。

2.2 用户请求的查询目标建模

用户请求由一组关键字描述,被定义为组件组合的执行目标,表述为 $g \subseteq K$ 。满足请求目标的数据对象定义为能够被目标请求中所有关键字描述的数据对象子集,其中考虑到关键字的层级结构。定义3给出满足用户请求的数据对象形式,即被满足的查询目标。

定义3 被满足的查询目标。满足用户请求的数据对象是数据对象集合 $D = \{d\}$ 的子集,其中描述数据对象的关键字能够被请求关键字集的所有关键字或其子关键字覆盖。

$Q = \{d \in D \mid \exists k' \in d(d) \quad \forall k \in q, \quad k' < k\}$

假定当前目标请求为“有序的工作任务集合”,表述为 $q = \{\text{workTask}, \text{sorted}\}$,有数据集“按时间降序排列的已执行的工作任务”,表述为 $d = \{\text{exeWorkTask}, \text{timeDesc}\}$ 。因为

$\text{exeWorkTask} < \text{workTask}$, $\text{timeDesc} < \text{sorted}$, 所以可以判定 $\forall k \in q, d = \{\text{exeWorkTask}, \text{timeDesc}\}$ 中每一个关键字的父关键字存在于 $q = \{\text{workTask}, \text{sorted}\}$ 中; 根据定义 3, d 是满足请求目标 q 的结果数据集之一。在本章开始提出的示例中, 用户请求的查询目标被描述为关键字集合“执行的工作任务”, “序列”, “对应的工作项目”, “开始时间限制”, 表述为 $q_1 = \{\text{exeWorkTask}, \text{array}, \text{corWorkProject}, \text{StartTimeRestrict}\}$; 有数据集“工作项目对应的在某时间点后执行的工作任务的升序集合”, 表述为 $\{\text{exeWorkTask}, \text{timeAscSeq}, \text{corWorkProject}, \text{StartTimeRestrict}\}$ 。因为“时间升序序列”是“集合”的子关键字, 即 $\text{timeAscSeq} < \text{array}$, 可以判定用 $\{\text{exeWorkTask}, \text{timeAscSeq}, \text{corWorkProject}, \text{StartTimeRestrict}\}$ 描述的数据集是满足目标 q_1 的结果数据集之一。

2.3 组件建模

本节给出组件的形式化定义。

定义 4 组件 $c = \langle \vec{i}, \vec{o} \rangle$ 被定义为在数据对象上的操作, $c(D, \vec{i}) = D \cup D'$, 其中 $D' = \{d_i' \mid d_i' \notin D\}_{i=1}^{n(c)}$ 是组件输出的新数据对象集合, 该数据对象被描述为 $d(d_i') = \vec{i} \cup \vec{a}(c) \setminus \vec{d}(c)$ 。各参数定义如下:

D : 数据对象集合;

\vec{i} : 描述组件 C 输入的关键字集合;

\vec{o} : 描述组件 C 输出的关键字集合;

$n(c)$: 组件输出中新数据对象的数量;

\vec{a} : 组件输出时添加的关键字集合;

\vec{d} : 组件输出时删除的关键字集合。

组件的语义通过关键字对其输入输出数据对象的描述而实现。

要完成本节初始提出的示例查询, 用到的组件之一的功能是根据工作项目查询该工作项目对应的工作任务, 包括已执行的和未执行的, 不妨表述为组件 c_1 : $\text{QueryWorkTaskByWorkProject} = \langle \langle (\text{workProject}), (\text{exeWorkTask}, \text{unexeWorkTask}, \text{array}) \rangle \rangle$, 在该示例组件中, 描述组件 c_1 的输入关键字集合为 $\vec{i} = \{\text{workProject}\}$, 即以“工作项目”作为输入, 描述组件 c_1 的输出关键字集合为 $\vec{o} = \{\text{exeWorkTask}, \text{unexeWorkTask}, \text{array}\}$, 即以“已执行和未执行的工作任务序列”作为输出, 组件 c_1 输出中新数据对象的数量 $n(c) = 2$, 即已执行的工作任务集合和未执行的工作任务集合, 其中集合的语义由关键字 array 描述, 组件输出时添加的关键字集合 $\vec{a} = \{\text{exeWorkTask}, \text{unexeWorkTask}, \text{array}\}$, 组件输出时删除的关键字集合 $\vec{d} = \{\text{workProject}\}$ 。

2.4 组件组合过程建模

本文的组件组合过程是以目标为驱动的形式。组件组合问题定义为找到一个组件序列最终产生满足目标描述的数据集合的问题。满足目标的组件序列是一个有向无环图(以下简称 DAG) $G(V, E)$: 顶点 $v \in V$ 是一个组件 C 的实例, 边 $(v_1, v_2) \in E$ 代表由顶点 v_1 到 v_2 的逻辑数据流, 边 $(v_1, v_2) \in E$ 的意义是组件实例 v_1 的输出可以作为 v_2 的输入, 并且 DAG 中至少有一条边代表的数据对象满足请求目标约束, 即 $Q \neq \emptyset$ 。

在组件组合的过程中, 组件的语义约束是对其结构约束的补充, 结构约束通过匹配组件的输出与输入参数类型实现, 语义约束通过前文提到的关键字定义, 即采用关键字集合描述组件的输入输出的数据对象语义和定义组件组合目标的语

义, 本文所述语义主要通过层级关键字结构描述, 其中层级关键字结构中的关键字子-父关系语义简洁易操作, 适用于本文所述的基于语义的组件组合方法。

3 基于语义的组件组合方法框架

根据第 2 章的方法建模思路, 设计基于语义的组件组合方法框架如图 2 所示。从顶层设计角度, 整个架构分为客户端表示层和服务端业务层。客户端与服务端的交互表现为客户端向服务端发送(数据查询)请求, 服务端向客户端返回应答集, 应答集定义为所有满足客户端查询条件的应答集合, 客户端用户可进一步选择查看某一具体应答。服务端业务层的处理流程描述为: 1) 接受来自客户端的请求后交由请求包装器包装成规划器可以识别的形式, 即由关键字描述的(数据查询)请求; 2) 规划器的核心部分是自动规划装置, 自动规划装置以反向搜索算法实现其自动规划功能, 它接收以关键字描述的请求后, 搜索组件池中的组件自动组合成满足请求目标的组件序列, 提交给执行引擎; 3) 执行引擎逐一调用组件序列中的组件, 合理连接组件的输出输入, 最后生成请求结果提交至结果包装器; 4) 结果包装器将请求结果包装成适合客户端表示层展示的形式, 返回至客户端。其中, 请求包装器提交给规划器的目标请求由关键字库中的关键字子集描述, 关键字库中的关键字子集还以描述组件输入输出数据对象的方式定义每个组件。

从方法设计原理图中可知, 基于语义的组件组合方法的核心是设计关键字库、组件库和自动规划装置。

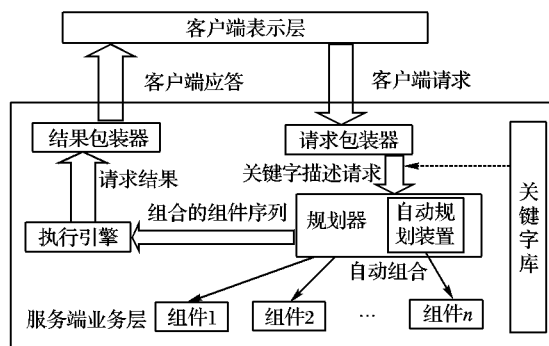


图2 基于语义的组件组合方法设计原理

4 基于语义的组件组合算法

经过上面的分析可知, 规划器在本文提出的方法中占有重要位置。客户端提交用关键字集合描述的请求给规划器, 规划器以自动规划理论为基础, 自动搜索输出适合请求的组件序列, 输出至组件执行引擎。

下面将详细介绍基于语义的组件组合算法, 即 SCC (Semantic-based Component Composition) 算法。该算法采用后向搜索策略, 如图 3 所示, 其基本思想是从目标开始, 反向规划操作, 产生子目标, 当算法产生的关键字集合被初始状态满足则停止。该算法可有效地解决组件自动组合的问题。

4.1 SCC 算法的设计与实现

首先介绍 SCC 算法, 该算法的具体伪代码如下:

Input: 组合目标 g ; 初始状态 S_0 ; 组件库 C
 Output: 一个满足既定目标的组件序列 σ
 $S \leftarrow g, \sigma \leftarrow \emptyset$
 loop
 if $S = S_0$, return σ ;

for each $c(\vec{i}, \vec{o}) \in C$
 if S_θ and \vec{o} satisfy the parital order relationship
 $S \leftarrow (S_\theta \setminus \vec{o}_\theta) \cup \vec{i}_\theta$
 $\sigma \leftarrow c_\theta \cup \sigma$

SCC 算法在搜索组件序列时采用后向搜索策略,通过在当前状态下运用具有合适输出关键字集的组件,从而产生新的状态。首先,将当前状态 S 初始化为指定的请求目标 g ,当前的规划序列 σ 为空;然后,对每一个状态分支 S_θ ,匹配组件集合 C 中每一个组件 $c(\vec{i}, \vec{o})$ 的输出关键字集合和 S_θ 关键字集合,若两者之间满足偏序关系(参见引理1),说明组件输出的数据对象满足当前状态的需求,即组件输出的语义包含当前状态表述的语义。当找到相应的匹配时,在当前状态的关键字集合中删除组件输出关键字集,增加组件输入关键字集合。同时将组件 c_θ 加入规划序列 σ 。在当前状态满足初始状态时,算法终止,并返回组件序列 σ 。

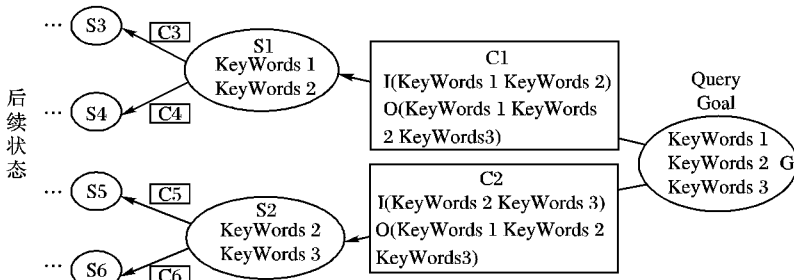


图3 后向搜索策略原理

4.2 SCC 算法复杂度分析

在人工智能规划领域,怎样找到一个最优规划仍然是一个难题。总的来说,并不存在最优或者接近常数因子的规划器可以保证所有规划任务在一个多项式时间内完成^[9]。如果每个组件的输入输出是唯一的,那么使用 SCC 算法找到一个合理的规划等同于在 STRIPS^[10] 领域找到一个合理的规划。而后者已经被证明是 PSPACE 完备的^[11],所以使用 SCC 算法找到一个最优或合理的规划至少是 PSPACE 完备的。另外,SCC 算法在剪枝策略的设计中引入关键字子-父关系约束(参见引理1),指数级地减少了不必要的搜索空间。

5 应用实例

为了加深理解,下面以航天器试验与测试系统为例,进一步阐明上述概念和模型含义。

将本文所述组件组合方法应用于实际系统时,从软件工程角度采用自顶向下的方式,遵循需求分析、应用建模、组件开发、单元/集合测试、软件部署等一系列步骤。特别地,本方法的构建过程又具有自底向上的特色,即在应用系统的测试和使用过程中,可以对组件进行改进或添加操作,另外,系统部署后,其功能并没有局限在既定的需求分析之中,而是支持使用规划器,通过已有的组件生成满足新功能的组件序列,这种特性是使用自动规划方法的优势之一。

以下将重点阐述关键字库设计、组件设计以及一个实例系统的工作过程。

5.1 层级关键字库设计

关键字是本文所述方法语义功能的核心。设计合理的关键字需要业务人员和组件开发人员的合作。设计的关键字能够准确描述组件的输入输出数据对象的意义,同时考虑和领域内术语保持一致。以航天器试验与测试数据查询系统为例,可建立层级关键字结构片段如图4所示。

符号 & 作为声明一个父关键字的记号。

```
keyword(satelliteID & ID)
keyword(subSysID & ID)
keyword(workTaskID & ID)
keyword(exeWorkTaskID & workTask)
```

图4 关键字定义片段

5.2 组件设计

用关键字描述组件输入输出的数据对象,为组件添加语义信息。

包含语义信息的组件的设计如图5所示。

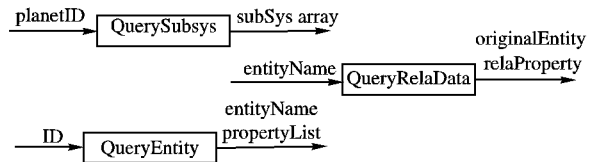


图5 组件描述片段

5.3 实例分析

系统原型采用 AJAX 技术实现本文所述方法应用的查询系统。在浏览器端,使用富客户端技术合理利用客户端计算能力,提供良好的用户体验;在服务端,采用 Apache Tomcat 作为应用服务器。组件库中的组件是对后台数据库进行的查询的 Java 类。规划器是一个可以接收用户请求描述,并为请求生成规划序列的程序模块。执行模块是根据规划器生成的组件序列逐一调用构成序列的基本组件,获得查询结果的程序模块。

分析一个查询案例如下:某用户需求为“查询某个工作项目对应的在某时间点后执行过的工作任务的集合”。首先,这个查询需求被描述成一组关键字集合:“执行的工作任务”,“序列”,“对应的工作项目”,“开始时间限制”,表述为 {exeWorkTask, array, corWorkProject, StartTimeRestrict}。然后这个查询请求被封装后提交至规划器,规划器采用后向搜索算法搜索组件库,给出组件组合序列。最后,将规划器给出的组件组合序列封装后提交至执行引擎。其中,规划器后向搜索算法的工作过程如图2所示,后向搜索算法从目标状态开始,搜索当前组件库中组件输出和当前状态的关键字匹配的组件如图6所示,由目标状态 G 出发,搜索组件库找到的组件 SelectByST 和 QueryByWorkProject,分别对应状态 S1 和 S2;转入状态 S1 后,继续搜索组件库,满足 S1 状态下关键字的组件 QueryWorkTask 和 SelectExeWorkTask 分别对应转入的状态 S3 和 S4;基于广度优先原则,当搜索转入 S3 后没有可用组件;S4 状态下搜索到满足条件的 QueryWorkTaskByWorkProject 组件,其输入关键字 WorkProject 满足提交的查询请求输入。最终,输出搜索得到的组件序列 SelectByST; SelectExeWorkTask; QueryWorkTaskByWorkProject。将该组件序列输出至执行引擎中,因为后向搜索的特点,执行引擎将逆向执行该组件序列。在本例中,首先根据数据的工作项目查询对应的工作任务,然后选出其中已执行的工作任务,最后选出开始时间在指定时间点后的工作任务。至此,查询过程结束。

6 结语

本文提出基于语义的组件组合的方法。该方法采用关键字描述的组件提供语义信息,以对组件组合方法过程的建模为基础,使用规划器自动搜索组件序列提供请求应答,很好地

解决了航天器数据查询与分析系统灵活性和扩展性差的问题。该方法已经在航天器试验与测试数据查询分析平台中得到了应用,并可方便地移植到其他企业应用系统中。

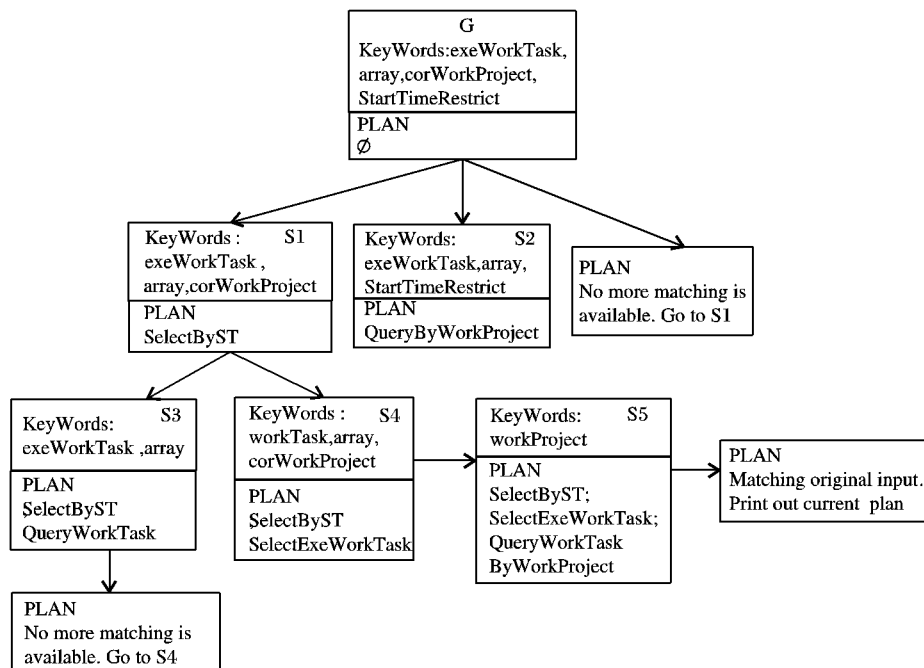


图6 后向搜索算法工作过程

参考文献:

- [1] 万常选, 鲁远. 基于权重查询词的 XML 结构查询扩展[J]. 软件学报, 2008, 19(10): 2611-2619.
- [2] RIABOV A V, R BOUILLET E, FEBLOWITZ M B, *et al.* Wishful search: Interactive composition of data mashups[C]// 17th International World Wide Web Conference. New York: ACM, 2008: 775-784.
- [3] BOUILLET E, FEBLOWITZ M, ZHEN L. *et al.* A tag-based approach for the design and composition of information processing applications[C]// Object Oriented Programming, Systems, Languages and Applications. New York: ACM, 2008: 585-602.
- [4] ZHEN L, RANGANATHAN A, RIABOV A, *et al.* A planning-based approach for the automated configuration of the enterprise service bus [C]// International Conference on Service-Oriented Computing. Berlin: Springer, 2008: 538-544.
- [5] BOUILLET E, FEBLOWITZ M, FENG H, *et al.* A folksonomy-based model of Web services for discovery and automatic composition [C]// Proceedings of the 2008 IEEE International Conference on Services Computing. Washington, DC: IEEE, 2008, 1: 389-396.
- [6] BIZER C, CYGANIAK R, GAUSS T. The RDF book mashup: From Web APIs to a Web of data[EB/OL]. [2009-06-01]. <http://www.dfki.uni-kl.de/~grimnes/2007/06/SFSW07Papers/6.pdf>
- [7] CHARIF Y, SABOURET N. An overview of semantic Web services composition approaches [EB/OL]. [2009-08-01]. <http://www.poleia.lip6.fr/~sabouret/ps/Charif-Sabouret-Context2005.pdf>.
- [8] YU D, LI Z W, YE G, *et al.* The workflow-based modeling method for spacecraft automatic testing process [C]// International Conference on Information and Knowledge Engineering. Las Vegas: CSREA Press, 2008: 267-271.
- [9] RIABOV A, LIU Z. Planning for stream processing systems[C]// Proceedings of the 20th National Conference on Artificial Intelligence. Pittsburgh: AAAI Press, 2005, 1: 1205-1210.
- [10] GHALLAB M, HOWE A, KNOBLOCK C, *et al.* PDDL: The planning domain definition language. [EB/OL]. [2008-10-10]. <http://www.cs.yale.edu/pub/mcdermott/software/pddl.tar.gz>.
- [11] BYLANDER T. The computational complexity of propositional STRIPS planning[J]. Artificial Intelligence, 1994, 69(1/2): 165-204.
- [5] 陈旺虎, 李婧. 基于抽象服务模型提高 Web 服务匹配的适应性[J]. 计算机科学, 2009, 36(9): 139-142.
- [6] HULL D, ZOLIN E, BOYVKIN A, *et al.* Deciding semantic matching of stateless services[C]// Proceedings of the Twenty-First National Conference on Artificial Intelligence. California: AAAI Press, 2006: 16-20.
- [7] MOTAHARI H R, BENATALLAH B, MARTENS A, *et al.* Semi-automated adaptation of service interactions [C]// WWW'07: Proceedings of the 16th International Conference on World Wide Web. New York: ACM Press, 2007: 993-1002.
- [8] BROGI R, POPESCU. Automated generation of BPEL adapters [C]// Proceedings of the 4th International Conference on Services Oriented Computing. New York: Springer-Verlag, 2006: 27-39.
- [9] KONGDENFHA W, SAINT-PAUL R, BENATALLAH B, *et al.* An aspect-oriented framework for service adaptation[C]// Proceedings of the 4th International Conference on Service Oriented Computing, Chicago, USA, 2006. New York: Springer-Verlag, 2006: 15-26.
- [10] HEMER D. A formal approach to component adaptation and composition[C]// Proceedings of the 28th Australasian Conference on Computer Science. Darlinghurst: Australian Computer Society, 2005: 259-266.
- [11] DUMAS M, SPORK M, WANG K. Adapt or perish: Algebra and visual notation for service interface adaptation[C]// Proceedings of the 4th International Conference on Business Process Management. Berlin: Springer, 2006: 65-80.
- [12] WANG X, CHEUNG S. C. Grammar based interface processing in Web service composition [C]// Proceedings of Conceptual Modeling for Advanced Application Domains. New York: Springer-Verlag, 2004: 323-334.
- [13] 王晓玲, 郭志懋, 周傲英. Web 服务组合的基于文法的消息处理. 计算机学报, 2005, 28(4): 478-485.
- [14] 房俊. 支持最终用户编程的服务虚拟化方法研究[D]. 北京: 中国科学院计算技术研究所, 2006.
- [15] CHIDLOVSKII B. Using regular tree automata as XML schemas [C]// Proceedings of IEEE Advances in Digital Libraries 2000. Washington, DC: IEEE Computer Society, 2000: 89-104.
- [16] 陈旺虎, 刘晨, 李厚福, 等. 支持虚拟组织的语义基础设施的动态构建方法研究[J]. 计算机学报, 2006, 29(7): 1127-1136.
- [17] 喻坚, 韩燕波. 面向服务的计算——原理和应用[M]. 北京: 清华大学出版社, 2006: 44-53.

(上接第393页)