# 一种语义为中心的分布构件系统故障诊断建模方法

黄 杰,陈 琳,贾 焰,邹 鹏
(国防科技大学计算机学院,湖南长沙 410073)

**摘 要**: 随着分布构件技术的发展,复杂的分布构件系统故障诊断问题越来越受到人们的重视.然而,现有的大多数分布构件系统故障诊断的研究没有充分考虑系统语义这一重要因素.文章提出了一种新颖的语义为中心的分布构件系统故障诊断思想.系统的语义主要由系统正常或者异常时的结构和行为特征构成.在给出基于模型的故障诊断问题定义之后,提出了具有复杂时序语义表达能力的系统逻辑模型编辑方法,通过在 Horn 短句中增加时序描述来增强诊断模型的表达能力.这种模型不仅对分布软构件系统具有较强的表达能力,还能够很好地利用现有的一阶逻辑定理证明器求解极小诊断问题,它并不需要具有高级时序逻辑处理能力的定理证明器的支持.另外,通过哲学家就餐问题还从直观上阐明了这种新颖的模型编辑方法.StarCCM 的诊断子系统由诊断引擎、诊断代理、诊断回调接口构成.它证明这种模型编辑方法能够有效地解决分布构件系统的语义诊断问题.

**关键词**: 故障诊断; 模型; 语义; 时序; 分布构件; CORBA
**中图分类号**: TP306 **文献标识码**: A **文章编号**: 0372-2112 (2004) 12A-243-04

# A Semantic Centric Fault Diagnostic Model Compiling Approach of Distributed Components Based System

HUANG Jie, CHEN Lin, JIA Yan, ZOU Peng
(*Dept. of Computer, National University of Defense Technology, Changsha, Hunan* 410073, *China*)

Abstract: With the development of distributed component technology, people set more and more focus on the diagnosis problem in such systems. Whereas, the current research on the diagnosis problem about distributed component systems often omits an important factor the system semantics. The system semantics are composed of the structure and behavior information under the normal and abnormal conditions. The paper proposes a novel ideal of diagnosis, which is a semantic centric method. After introducing principle of model based diagnosis, the paper puts forward a novel model compiling approach by providing the time series information in the Horn clauses, which can express complicated temporal semantics and compute the minimal diagnosis by the theorem provers of first order logic. Further more, the paper sets forth the model compiling method by a philosopher dinner problem. In the end, the diagnosis subsystem implemented in StarCCM is introduced, which consists of the diagnosis engines, diagnosis agents and diagnosis callback interfaces. The novel diagnosis model compiling approach is proved to be applied in the distributed component system effectively.

Key words: fault diagnosis; model; semantic; time series; distributed components; common object request broker architecture (CORBA)

## 1 引言

从上个世纪 90 年代中期直至今日,中间件技术的快速发展使得分布构件技术日趋成熟.分布构件技术以降低中间件平台的复杂性,支持分布式企业应用和代码的二进制重用为目标,已广泛的应用到大型的企业计算之中.目前,主流的企业级分布构件平台有 SUN 公司的 EJB, OMG 组织的 CCM (CORBA Component Model), Microsoft 的 COM+.

随着应用系统越来越复杂,系统的测试、调试、维护成本越来越高,软件的故障诊断问题越来越多的受到人们的重视.然而,软件固有的复杂性和应用业务逻辑的复杂性交织在一起使得基于软件的系统故障诊断问题十分复杂,相关的研究进展十分缓慢.分布构件技术的成功给软件系统的故障诊断问题带来了新的机遇和挑战.国外许多著名大学已经开始重新关注和研究软构件平台下的诊断方法,但目前国内在该领域的研究工作还比较少.

本文将在分析软构件故障诊断研究现状的基础上,提出"语义为中心"诊断的新思想.通过基于模型的故障诊断方法的定义,阐述一种新颖的模型编辑方法,这种方法不仅能够可以表达软构件系统中的复杂时序语义逻辑,而且利用现有的一阶定理证明器求解问题.它不依赖于更高级的定理证明器,使得这种模型编辑方法更加有效和实用.

## 2 研究进展和语义为中心的软构件故障诊断思想

分布构件技术、软件工程技术在上个世纪末和这个世纪初得到了长足的发展,这些技术本身已经广为人们接受和使用.与此同时,分布构件技术和软件工程技术的巨大进步也给基于软件系统故障诊断问题的解决带来了新的机遇.系统的模块化、对象化、构件化,已经成为今后软件开发的必然趋势.基于分布构件的系统有明确的"软件边界".这使得基于软构件系统的行为在一定粒度上能够被观测,同时也为软件系统的故障诊断问题打开了新的思路.

收稿日期:2004-09-02;修回日期:2004-11-16
基金项目:国家自然科学基金(No. 90104020);国家高技术研究发展计划(863)(No. 2001AA113020);国家重点基础研究发展规划(973)(No. G1999032703)

目前,国外的著名大学已经在软构件系统的故障诊断问题上作了一些富有成效的研究工作.例如,Berkeley 分校 ROC 小组的 Pinpoint[1]系统基于 EJB 平台给出一种通过跟踪分布调用请求的路径进行故障诊断的方法,并提出通过所有的失效路径来确定根故障原因.文献[2]提出通过故障注入来挖掘系统中潜在的异常,通过建立异常知识库对系统故障进行管理来支持诊断.麻省理工大学提出一种软构件平台中的自动异常处理体系结构[3],这种思想同目前研究较热的"自主计算"思想不谋而合.这些研究成果即具有一定的先进性,同时也有一定的局限性.如 Pinpoint 系统提出以"路径"为核心的诊断思想,但解决故障诊断问题的前提是单故障假设,或者是将多故障问题当作单故障处理.在这种前提下,由于每次只能定位一个最主要的故障,因此会导致诊断效率的下降.在更严重的情况下,也可能由于多故障间的互相干扰致使诊断结果严重失误.另外 Pinpoint 以"路径"为核心诊断和文献[2,3]中以"异常"为核心诊断方法没有考虑应用系统的语义.尤其是[3]中提到的异常自动处理技术,如果没有语义的支持就会使得文中提到的方法在继续深入研究时,陷入进退两难的境地.

伴随着软件的普及化和复杂化,软构件系统的诊断问题正越来越多地受到人们的重视.现有的方法还缺乏相应的理论指导.大量的应用表明,分布构件的失效常常是因为分布构件或由分布构件组成的系统偏离了原先规定的行为而造成的.这种预期的行为常常称作"系统规范".现有的故障诊断方法对这种分布构件系统的实际行为偏离了预期规范的情况考虑得较少.既然大量的软件故障同应用的语义相关,这就使应用的语义在处理软件的故障诊断问题时变得十分重要.

基于以上分析,我们提出一种"语义为中心"(Semantic Centric)的故障诊断方法.系统的"语义"即系统的逻辑结构和行为特征,它反映了系统最重要的逻辑要素.系统的"语义"既可以反映正常时的结构和行为特征,也可以反映异常时的结构和行为特征."语义为中心"的故障诊断方法以基于模型的故障诊断理论为基础,充分考虑了软件具有复杂的"时序"特性这一事实."时序"的引入增强了模型的表达能力,尤其是对于具有复杂时序特征的分布构件系统.以"语义为中心"的诊断方法能够有效刻画复杂时序下的系统行为,并以此作为分布构件系统运行时故障诊断的参考依据.

以语义为中心的故障诊断方法关键技术包括系统的模型编辑方法和故障诊断算法.由于篇幅关系,本文将集中讨论我们提出的这种具有"时序"特性的软构件系统模型编辑方法,而相关的故障诊断算法已经另文给出[4].首先我们将给出基于模型的故障诊断方法描述.

## 3 基于模型的故障诊断问题描述

基于模型的故障诊断基本思想是,通过为系统建立一种诊断模型来预测系统的行为.如果系统观测到的行为同预期的行为不符,我们就利用已经建立起来的模型和观测的结果导出候选诊断[5].以下给出基于模型的故障诊断技术的基本概念.

**定义 1** 一个系统被定义为一个序偶$\langle SD, COMP\rangle$.
$SD$ 代表软构件系统的逻辑描述,它是一个一阶语句的集合.分布构件系统由一组软构件构成,将这组软构件的集合表示为 $CMP$.

**定义 2** $\langle SD, COMP, OBS\rangle$的诊断是一个关于构件集合 $\Delta \subseteq COMP$, $SD \cup OBS \cup \{Ab(c) \mid c \in \Delta\} \cup \{\neg Ab(c) \mid c \in COMP-\Delta\}$,其中 $OBS$ 代表对系统状态和行为的一次观察,它也是一个一阶语句集合.$Ab$ 是表示某个构件发生异常的谓词.

**定义 3** $\Delta$ 是$\langle SD, COMP, OBS\rangle$的一个极小诊断,当且仅当 $\Delta$ 关于 $\subseteq$极小.

**定义 4** 给定模型 $M_1$ 和 $M_2$,我们定义 $<_{Ab}$,并且 $M_1 <_{Ab} M_2$, iff $M_1|Ab| \subset M_2|Ab|$,其中 $M_1$ 和 $M_2$ 是 $SD \cup OBS$ 的模型.$M|Ab|$ 代表模型包含的故障构件.

**定理 1**[6] 一组构件集合 $\Delta$ 是$\langle SD, COMP, OBS\rangle$的极小诊断,当且仅当存在一个关于 $SD \cup OBS$ 的 $<_{Ab}$极小模型 $M$,且 $\Delta = M|Ab|$.

极小诊断的求解依赖于系统的语义.如何表达软构件系统复杂的时序逻辑,使得可以通过成熟的工具求解极小诊断,就对软构件系统的模型编辑方法提出了较高的要求.

## 4 语义为中心的分布构件系统诊断模型编辑方法

为了对系统偏离"规范"的行为进行精确的诊断,首先需要能够对系统的有效语义进行描述.文献[7]中的 livestone 引擎为航天器建立系统模型,通过一阶逻辑诊断可能的故障.文献[8]中提出了一种类似 Clips 语法的模型编辑方法 MPL.但是,这两种模型描述相对比较简单,他们只能用于建立简单的静态系统模型.以前的方法将系统看作是相对静止的,它按照系统的观察值和静态的系统描述来建立诊断模型.然而,复杂的软构件系统具很强的时序特性,可以用一定的时序逻辑来表示.如何为分布构件系统建立系统模型来反映软构件系统的语义行为和这种时序特性是故障诊断的关键.同时,编辑出的模型要能够尽可能利用现有的成熟的工具求解.

因此,在模型编辑方法的设计上,我们认为应该保留一阶逻辑的描述方法,同时通过加入时序变量和常量表达软构件系统中的复杂时序关系.此外,我们认为分布构件通过方法的组合来完成其业务逻辑功能.方法是业务逻辑的宿主,要对分布构件组成的系统进行逻辑诊断就首先需要对方法和其运行环境进行建模.在以方法为基本单位的诊断模型基础上,通过以下一系列描述和定义给出系统语义模型 $SD$ 的编辑方法.

**定义 5** $MS$ 代表所有方法的集合.$ARGS$ 代表所有传入和传出参数的集合.$ENV$ 代表方法外部环境变量的集合.

**定义 6** 方法到其输入参数集的映射为 $\tau: M \rightarrow \rho(ARGS)$ 其中 $M \in MS$,$\rho(X)$ 表示 $X$ 的幂集.

**定义 7** 方法到被其调用方法集的映射为 $\sigma: M \rightarrow \rho(M)$,其中 $M \in MS$.

**定义 8** 方法到对应的外部环境变量集的映射为 $\tilde{\omega}: M \rightarrow \rho(ENV)$,其中 $M \in MS$,$\rho(ENV)$ 是 $ENV$ 的幂集.

**定义 9** 一组方法的输出变量以映射 $\vartheta: \rho(M) -> \rho(ARGS)$ 表示.

**定义 10** 方法到被其调用的方法输出参数的映射 $\mu: M \rightarrow \vartheta(\rho(M))$.

**定义 11** 设 $X$ 为一个实体集合,这些实体可以是方法、

参数、环境.定义 $P(X)$为关于 $x \in X$ 的所有谓词集合. $P(X, S)$为关于 $x \in X$ 在时序标号 $S$ 时的所有谓词集合.

**定义 12**　系统描述 $SD$ 是一组 Horn 短句的集合 $SD = \{h \mid h \in \rho(P(\{M\}, S)) \times \rho(P(\tau(M), S)) \times \rho(P(\omega(M), S)) \times P(\sigma) \times \rho(P(\mu(M), T)) \rightarrow \vartheta(P(\{M\}, S))\}$,其中 $S$ 和 $T$ 均为一个时序标号,且 $T$ 代表的时刻晚于 $S$.

**定义 13**　诊断构件的集合被定义为 $COMP = \{\rho(P(MS \rightarrow SS))\}$,其中 $SS$ 是所有调用序列标号的集合.

**定义 14**　系统的观察值被定义为事实的集合 $OBS = \{f \mid f \in P(\{M\}, S) \cup P(\tau(M), S) \cup P(\omega(M), S) \cup P(\sigma) \cup P(\mu(M), T) \cup P(\{M\}, S)\}$. 我们以分布构件平台 CCM(CORBA 构件模型)为例,从直观上阐述这种以一阶逻辑的 Horn 短句为基础的,并具有时序特性的软构件系统语义模型编辑方法. CCM 的研究者常常以经典的哲学家就餐问题展开相关的研究.文献[9]中给出了我们自主开发的 StarCCM 平台实现,并提供了哲学家就餐问题的描述[9]. 这里我们从一个简化的 IDL3 描述开始来解释本文提出的模型编辑方法.

```
module Dinner{
interface Fork{
    Cookie obtain_fork()
    raises (ForkNotAvailabel);
    void release_fork(in Cookie ck)
    raises (NotTheEater);
};interface State{
   void goto_state(State state)
    raise(ChangeStateFailed);
};
component Philosopher{
    uses Fork left_hand;
    uses Fork right_hand;
      provides State the_state;
};
component Cutlery{
    provides Fork the_fork;
};
…
};
```

上述 IDL 定义了哲学家构件和餐具构件.哲学家可能会有思考、饥饿、就餐、睡眠等几种状态,并在这几种状态之间进行转换.每个哲学家的左右手分别有一个餐具,只有在双手都获得了餐具时才可以进入就餐状态,否则就需要在饥饿状态进行等待.这是一个典型的竞争问题.

按照上述的具有时序特征的语义模型编辑方法,我们对 Cutlery 构件 Fork 刻面的 obtain_fork 和 Philosopher 构件的 goto_state 方法进行建模.

首先用 Horn 短句对系统行为进行描述. obtain_fork 的行为如下:

out(Fun, not_available, Seq) :-obtain_fork(Fun, Seq), has_owner(Fun, Seq), not(ab(Fun, Seq)).

out(Fun, no_exception, Seq) :-obtain_fork(Fun, Seq), not(has_owner(Fun, Seq)), not(ab(Fun, Seq)).

上面的描述说明,在调用序列的某个时刻,如果方法 obtain_fork 行为是预期的,那么在餐具已经被某个调用者拥有时输出 ForkNotAvailable 异常,否则没有异常产生.其中 obtain_fork/2 是关于方法类型的论断,has_owner/2 是方法执行时环境的论断.

相应的可以给出关于 Philosopher 构件 goto_state 方法的部分 Horn 短句描述:

out(Fun, state_change_failed, Seq) :-goto_state(Fun, Seq), arg1_huangry(Fun, Seq), depend(Fun, F1, F2), out(F1, not_available, [-|Seq]), out(F2, _, [_|Seq]), not(ab(Fun, Seq)).

out(Fun, state_change_failed, Seq) :-goto_state(Fun, Seq), arg1_huangry(Fun, Seq), depend(Fun, F1, F2), out(F1, _, [-|Seq]), out(F2, not_available, [_|Seq]), not(ab(Fun, Seq)). out(Fun, no_exception, Seq) :-goto_state(Fun, Seq), arg1_huangry(Fun, Seq), depend(Fun, F1, F2), out(F1, no_exception, [_|Seq]), out(F2, no_exception, [_|Seq]), not(ab(Fun, Seq)).

头两个 Horn 短句描述了在 goto_state 的某次调用中,如果被 goto_state 调用的两个 obtain_fork 方法中有一个抛出异常,goto_state 也要抛出相应的异常.最后一个 Horn 短句论述了 goto_state 执行成功的情况.其中 arg1_huangry/2 是关于 goto_state 方法输入参数值的论断. depend/3 描述了方法之间的调用关系. [_|Seq]序列标号表明 obtain_fork 的调用存在于 goto_state 的序列之中.

此外我们还需要定义一些短句来简化关于观察结果和系统构件故障假设的描述.

out(Fun, _, Seq) :-ab(Fun, Seq).　　ab(Fun, _) :-ab(Fun).

obs(Fun, Value, _) :-obs(Fun, Value).

第一个短句说明如果方法的执行偏离预期的行为,什么样的输出结果都是有可能的.系统的观察 OBS 被描述为一系列的实事.

depend($p$, $f1$, $f2$). goto_state($p$, [1]). obtain_fork($f1$, [1, 1]). obtain_fork($f2$, [2, 1]). has_owner($f1$, [1, 1]). arg1_hungry($p$, [1]).

obs($p$, no_exception).

我们可以从 CCM 的组装描述符中提取 depend/3 所需要的系统组装知识.在系统运行时获取系统的其他观察值.最后根据如下谓词来判断系统的运行是否偏离预期行为.

contradiction :- obs($p$, Value, [1]), not(out($p$, Value, [1])).

尽管一阶逻辑可满足性的判定是一个 NP 难题,但是对于一组简单的 Horn 短句(在项的深度和参数元组个数都是有界时)来说并不困难.如果出现故障,可以通过给出系统的故障假设和 contradiction/0 谓词来判定系统可能的故障.

例如,这个哲学家问题的分布构件系统在上述观察中出现了与预期行为的偏差.系统可以自动的提出一系列的故障假设,并根据该诊断建模方法验证假设是否正确.可以看出在假设 ab($f1$)或者 ab($p$)时,谓词 contradiction/0 的值是假.这说明在该次观察中产生了两个极小诊断,一个是仅构件 Cutlery 构件的 obtain_fork 方法偏离了行为规范,另一个是仅 Philosopher 构件的 goto_state 方法行为不合行为规范.当然,在复杂的系统中极少诊断的基数可能会大于 1.

本节提出这种分布构件系统模型编辑方法不仅能够精确表达系统的时序语义,同时简单的 Horn 短句描述方式使得诊断引擎可以使用已有的成熟定理证明器.尽管需要相应的背景知识,但这种建模和诊断方法却能够在复杂时序下更精确定位故障和快速查明故障原因,同时它支持多故障假设.

## 5　StarCCM 中在线语义故障诊断子系统

在上述系统时序语义建模方法的基础上,在我们已有的

CORBA 构件模型平台 StarCCM[8] 中,我们实现了一个分布构件时序语义诊断子系统.

图 1 是 CCM 容器系统结构,为支持这种基于模型的诊断技术,在容器中增加了一个诊断代理对象(DA). DA 位于 servant 将请求派发给构件实现路径之中,故它可以截获请求的参数和执行结果. DA 通过对调用参数和结果做出相应的判断产生诊断需要的部分"事实"或"观察".

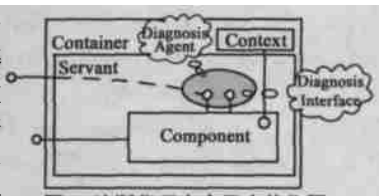

图 1 诊断代理在容器中的位置

同时在构件原有接口上,为构件的"执行体"增加了一个称作诊断接口(DI)的本地接口(local interface). DA 使用 DI 接口提供的功能访问构件状态或间接通过 DI 接口使用上下文接口来访问方法调用时的容器环境. 有了 DI 接口支持,DA 就可在运行时产生方法调用时的环境"实事"或对环境的"观察".

图 2 为多个分布构件服务器构成的结构图. 一个域包含多个分布的构件服务器(CS). 域管理器上管理着运行于 CS 上的所有的"组装件". 对应每个组装件,为其设计了一个组装件的诊断引擎(ADE),它由逻辑定理证明器和组装件系统描述组成. DA 主动的按照 ADE 的指示观察系统的状态,并将观察的结果发送至 ADE. ADE 根据观察到的事实和已有的系统描述进行故障推理. 在发生故障时,按照一定的规则做出故障假设. 通过对假设的评估,可以求出组装件系统的极小诊断. 从而判断分布构件系统在运行时的行为与行为规范上的偏离.

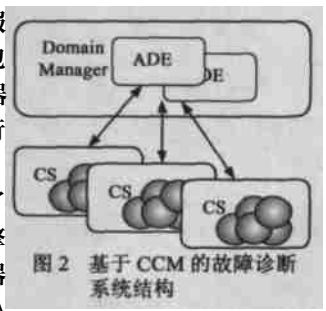

图 2 基于 CCM 的故障诊断系统结构

此外,为了获得请求的调用序列,需要为构件服务器实现一个截获器. 该截获器跟踪请求的调用路径,为分布调用产生类似 UML 时序图的时序标号. DA 可以根据 CORBA 线程和请求级的上下文来获取这个时序标号. 这个时序标号用于刻画观察的发生时刻和先后顺序. 如果没有时序标号就很难刻画出复杂的分布构件系统的交互关系.

## 6 结束语

本文从分析分布构件故障诊断方法入手,提出了一种新颖的、以"语义"为中心的软构件故障诊断思想. 针对基于模型的诊断方法,构造了具有描述复杂时序系统的语义模型编辑方法. 这种方法不仅具有强大的语义表达范围,而且可以使用已有的定理证明器求解极小诊断. 这使得本文提出的模型编辑方法可以很容易的在现实系统中使用和实现. 同时,我们讨论了 StarCCM 构件系统中诊断子系统的若干问题. 这种模型编辑方法,将会成为一种十分有潜力的语义诊断技术应用于分布构件系统的故障诊断领域.

## 参考文献:

[1] M Chen, E Kiciman, E Fratkin, A Fox, E Brewer. Pinpoint: Problem Determination in Large Dynamic Internet Services [DB/OL]. http://www.cs.berkeley.edu/~mikechen/publications/ipds2002/ipds2002.pinpoint.pdf, 2002.

[2] Chrysanthos Dellarocas, Mark Klein. A knowledge-based approach for designing robust business processes[J]. Business Process Management, 2000:50 - 65.

[3] Chrysanthos Dellarocas. Toward Exception Handling Infrastructures for Component based Software[DB/OL]. http://ccs.mit.edu/dell/papers/icse98.pdf, 1998.

[4] 黄杰,陈琳,邹鹏. 一种求解极小诊断的遗传模拟退火算法[J]. 软件学报,2004,15(9):1345 - 1350.
Huang Jie, Chen Lin, Zou Peng. Computing minimal diagnosis by compounded genetic and simulated annealing algorithm[J]. Journal of Software, 2004, 15(9):1345 - 1350.

[5] 李占山,姜云飞. 对基于模型诊断测试理论的修正与扩充[J]. 软件学报,2000,11(7):979 - 983.
Li Z S, Jiang Y F. A Correction and extension to the testing theory for model-based diagnosis[J]. Journal of Software, 2000, 11(7):979 - 983.

[6] Peter Frohlich, Wolfgang Nejdl. A static model-based engine for model-based reasoning[A]. Proceedings 15th International Joint Conference[C]. Nagoya, Japan, Artificial Intelligence, 1997. 466 - 473.

[7] Brian C Willians, P Pandurang Nayak. A model based approach to reactive self-configuring systems[A]. Workshop on Logic-Based Artificial Intelligence[C]. Washington, June, 1999. 14 - 16.

[8] Seung H Chung, Anthony Barrett. Distributed Real - time Model based Diagnosis[DB/OL]. http://ase.jpl.nasa.gov/public/planning/papers/barrett - aero03.pdf, IEEE AC #1116, 2003.

[9] StarCCM is a CORBA Component Model implementation in C++ language[DB/OL]. http://starccm.sourceforge.net, 2004.

## 作者简介:

黄 杰 男,1976 出生于陕西西安,博士生,主要研究领域为分布计算,构件系统与智能诊断. E-mail:huangjie@nudt.edu.cn.

陈 琳 女,1976 出生于福建厦门,博士生,主要研究领域为网络系统管理和智能诊断.