国产基础软件应用平台的兼容性测试研究

郭 军 刘 强 王云升 李宪莉 张 斌
(东北大学信息科学与工程学院 沈阳 110004)

摘 要 近年来,随着我国基础软件产业的蓬勃发展,操作系统、数据库软件、服务中间件、基础办公软件等国产基础软件[1]已得到广泛的应用。然而,国产基础软件产品组合关系的复杂性以及引起国产基础软件应用平台兼容性问题的不确定性,给国产基础软件应用平台兼容性测试带来了一定的困难。对此提出了一种基于依赖关系的国产基础软件应用平台的兼容性测试方法DRBA(Dependency Relationship Based Approach),给出了DRBA方法的具体描述和测试执行策略,并以C/S的方式设计实现了基于DRBA的兼容性测试工具,应用一个实例验证了DRBA的可用性和有效性。

关键词 国产基础软件,兼容性测试,软件测试
中图法分类号 TP302.7 **文献标识码** A

Research on the Compatibility Testing of the Domestic Foundational Software Application Platform

GUO Jun LIU Qiang WANG Yun-sheng LI Xian-li ZHANG Bin
(College of Information Science & Technology Engineering, Northeastern University, Shenyang 110004, China)

Abstract In recent years, with the continuous advance of the domestic foundational software, operating system, database software, service middleware, office software and other domestic foundational software have been used widely in many social fields. However, it is hard to test domestic basic software application platform compatibility due to combination relations complexity of domestic basic software and uncertainty of causing application platform compatibility problems. The paper presented a dependency-based domestic foundational software application platform compatibility testing method DRBA (dependency relationship based approach) and its policies. The method was defined and encoded following the execution tree policy. And a compatibility testing tool based on DRBA by the C/S way was implemented to verify the method's availability and effectiveness.

Keywords Domestic foundational software, Compatibility testing, Software testing

1 引言

国产基础软件兼容性测试是基础软件测试领域的一个重要方向,涉及到两两软件的接口规范和整个基础平台的兼容性等问题[2]。通过开展相关研究形成并完善国产基础软件兼容性测试标准、测试模型和测试用例集,为基础平台产品选型和国产基础软件的普及提供良好的保障。所以,对于国产基础软件的兼容性测试是具有现实意义的。关于组件版本兼容性测试,I.-C. Yoon 等人在文献[3]中提出了一种基于直接依赖关系的组件版本兼容测试模型,其基本思想是测试具有直接依赖关系的组件的兼容性。关于组件COTS的兼容性测试和回归测试方面,Leonardo Mariani 等人在文献[4]中针对COTS组件的频繁更新和多种不同的组件供应商的选择,提出了一种基于组件交互的行为表示模型,在测试执行的过程中自动生成原始行为模型的测试套件。目前提到的方法仍存在着一些不足:(1)对于组件而言,版本不易变动,组合空间相对较小,操作性较简单;而国产基础软件存在组合爆炸性问题,组合方式不确定;(2)组件的兼容性比对接口相对固定,依赖性单一;而国产基础软件间的依赖关系不仅存在直接的依赖关系,而且存在间接的依赖关系,组合关系复杂。

对此本文提出了一种运用依赖关系的思想对国产基础软件兼容性进行测试的方法,以提高国产基础软件的质量和互操作性。本文第2节分析国产基础软件关联关系,并介绍相关概念;第3节设计国产基础软件组合空间和构造执行树,并给出执行树执行策略;最后设计并实现了基于DRBA的系统原型。

2 国产基础软件关联关系分析

2.1 基本概念

国产基础软件生产商产品众多,相关的基础软件存在多个版本,而且版本组合关系复杂,依赖关系易变动,所以国产基础软件版本组合数目较大,需要建立一个更易于操作的国产基础软件组合空间。DRBA利用国产基础软件间的依赖关系和前驱关系确定国产基础软件应用平台组合空间。为了让

到稿日期:2011-08-10 返修日期:2011-11-24 本文受国家高技术研究发展计划863计划(2009AA01Z122),核高基科技重大专项(2010ZX01045-001-008),辽宁省工业攻关计划(2010216005)资助。

郭 军(1974—),男,博士,副教授;刘 强(1987—),男,硕士,E-mail:sosoliveyes@163.com(通信作者)。

组合空间中的组合之间按照一定的数据结构组织，本文把已确定的组合空间归并为执行树，将每一个国产基础软件组合的创建抽象为执行树结点任务，依据一定的执行策略执行每个结点任务。DRBA 在生成国产基础软件组合空间的时候根据约束集中的约束裁剪多余的国产基础软件组合，缩减组合空间。DRBA 主要包括应用平台、软件依赖图[5]、组合空间、约束集、执行树（前缀树[6]）等元素，如图 1 所示。其中：

1）一个国产基础软件应用平台包含一个软件依赖关系图，软件依赖关系图用来刻画国产基础软件之间的依赖关系；

2）依赖关系图可刻画国产操作系统、国产数据库、国产中间件和国产办公套件之间的关系；

3）参考软件依赖关系图，在 0 个或者多个约束集作用下，一个软件依赖关系图对应多个组合空间；

4）执行树是由组合空间中的国产基础软件组合按照某种顺序构造的树型数据结构，一个组合空间能构造多个执行树。

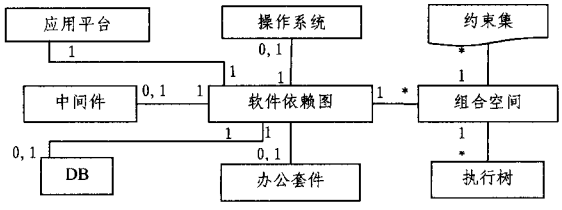

图 1　DRBA 元素关系图

软件依赖图是 DRBA 的顶层元素，它是一个有向无环图，是国产基础软件间的依赖关系和前驱关系的形式化表示，用它能够生成国产基础软件的组合空间。

定义 1（软件依赖图，Software Dependency Graph，SDG）$SDG=(N,E)$。$N=S\cup R$，其中 S 是国产基础软件结点集，R 是一组关系结点集，$R=\{r|+\cup^*\}$，关系结点是一个逻辑谓词（AND，OR）函数。E 是依赖关系边的集合，连接两种不同的结点。$E=\{(s,r)|s\in S,r\in R\}$。SDG 具有如下 3 种类型的结点：(1) 根结点，即 SDG 中入度（以结点为头的弧的数目）为 0 的结点，一般根结点是应用平台。(2)叶结点，即 SDG 中出度（以结点为尾的弧的数目）为 0 的结点。(3)非根、叶结点，即 SDG 中出度和入度均不为 0 的结点。

参考 SDG 建立国产基础软件的组合空间，不同的规则和限制条件产生不同的组合空间。为了按照一定的方式产生合理的组合空间，需要定义约束集来限定国产基础软件之间的组合方式。

定义 2（约束集，Constraint Set，CS）　本文中提到的 CS 包括 3 部分：(1)对于每一个 $s\in S$ 的国产基础软件，初始化 s 的版本范围，给出国产基础软件版本集；(2)覆盖准则（Coverage Criteria），记为 CC，它是生成组合空间的方式；(3)国产基础软件版本之间的限制和国产基础软件在国产基础软件组合中的约束。国产基础软件间的限制和约束具体是用关系算子（$\leqslant,\geqslant,<,>,==$）和布尔算子（$\wedge,\vee,\rightarrow,\neg$）表示的。

采取一定的策略，在约束集的作用下，创建国产基础软件应用平台组合空间。组合空间由国产基础软件组合和约束集组成。

定义 3（组合空间，Combination Space，CSP）　$CSP=\{cs_i|cs_i=(s_{1v},s_{2v},\cdots,s_{nv})\bigcap c_i,c_i\in CC\}$，$s_{iv}$ 表示国产基础软件 s_i 的版本 v，它是软件版本的所有组合构成的集合。组合空间的大小直接关系到测试的复杂程度。

组合空间中的元素是离散分布、无结构的，故对国产基础软件的兼容性测试需要构造数据结构，调优兼容性测试。本文所建立的数据结构是执行树，其将每个国产基础软件组合映射到执行树的结点。

定义 4（执行树（前缀树），Execution Tree，ET）　ET 是根据字串的前缀组织而成的树形结构。它与二叉排序树不同，在树里没有结点存储与结点相关联的关键字，树中结点是用它在树中的位置展示与它相关的关键字，根结点与空字符相关。执行树第 i 层的每个结点都包含了一个长度为 i 的字串以及该字串在文中出现的频率。父结点中的字串是子结点中字串的最大前缀，互为兄弟结点的字串仅最后一个字不同。

2.2　关系分析

不同的国产基础软件之间存在着各种关联关系，关联关系可以通过形式化的模型描述，例如基于基数的特征模型[7]和基于规则的形式化模型[8]。本文通过形式化的图形 SDG 分析国产基础软件之间的依赖关系和前驱关系。

依赖关系是记录不同国产基础软件间的依存关系，依赖关系有强依赖和弱依赖。

定义 5（依赖关系，Dependency Relationship，DR）　若 $s_{vi}\in S,s_{vk}\in S,i\neq k$，且 s_{vi} 依赖于 s_{vk}，则序偶 $\langle s_{vi},s_{vk}\rangle$ 称为一个 DR。

国产基础软件之间的依赖关系就是国产基础软件间的依存关系，所以依赖关系具有传递性：若 $\langle s_{vi},s_{vk}\rangle\in DR$ 且 $\langle s_{vk},s_{vj}\rangle\in DR$，则 $\langle s_{vi},s_{vj}\rangle\in DR$。

强依赖就是直接依赖（Direct Dependency，记为 DD），它是指序偶 $\langle s_{vi},s_{vj}\rangle\in DR$ 的 s_{vi} 与 s_{vj} 是直接可达的。$DD=(S_v,Dep)$，S_v 表示软件 S 的版本 v，Dep 是 S 直接依赖的软件版本集合。弱依赖就是间接依赖（Indirect Dependency，记为 ID），它是具有传递性质的依赖。$ID=(S_v,IDep)$，S_v 表示软件 S 的版本 v，$IDep$ 是 S 直接依赖和间接依赖的软件版本集合。兼容性测试就是测试 S_v 与 Dep 或 $IDep$ 中的软件版本的兼容性。

国产基础软件之间还会存在前驱-后继关系，其用来反映国产基础软件的创建顺序。

定义 6（前驱关系，Precursor Relationship，PR）　在 SDG 中，结点间的有向边表示前驱或偏序关系“$\succ$”，$\succ=\{(s_{vi},s_{vj})|s_{vi}$ 必须在 s_{vj} 创建之前创建$\}$。$(s_{vi},s_{vj})\in\succ$，记为 $s_{vi}\succ s_{vj}$，s_{vj} 是 s_{vi} 的直接后继。由定义知前驱关系同样具有传递性。

图 2 所示的是一个国产基础软件应用平台的 SDG 示例。从图中可知国产基础软件应用平台 A 直接依赖于国产基础软件 B 和 C，所以国产基础软件 B 的各个版本同 A 的各个版本存在前驱关系 $\langle B_v,A_v\rangle$，同理 C 的各个版本同 A 的各个版本也构成前驱关系 $\langle C_v,A_v\rangle$；国产基础软件 B 直接依赖于国产基础软件 D，则国产基础软件的各个版本同 B 的各个版本存在前驱关系 $\langle D_v,B_v\rangle$；国产基础软件 C 和 D 直接依赖于 E，则国产基础软件 E 的各个版本同 C 的各个版本和 D 的各个版本存在前驱关系 $\langle E_v,C_v\rangle$、$\langle E_v,D_v\rangle$。根据依赖关系和前驱关系的传递性得国产基础软件应用平台 A 间接依赖于国产

基础软件 D 和 E，国产基础软件 B 间接依赖于国产基础软件 E，国产基础软件 D、E 是 A 的前驱，国产基础软件 E 是 B、C 的前驱。

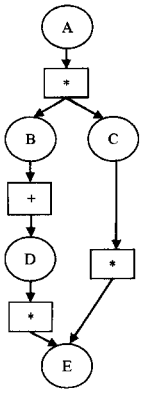

图 2　国产基础软件 SDG 示例

3　组合空间设计与执行树构造

3.1　组合空间生成

本文中组合空间生成的覆盖准则采用的是国产基础软件间的依赖关系。利用国产基础软件之间的依赖关系和前驱关系生成国产基础软件组合空间，能够缩减国产基础软件的组合空间，提高测试效率。根据定义 5 知，直接依赖关系和间接依赖关系均可以生成软件组合空间。直接依赖关系只是两个软件间的直接关系，所以只能测试二者的兼容性，而间接依赖关系能够测试多个软件之间的兼容性。下面介绍依赖关系的两种组合空间生成策略。

依赖关系包含于软件组合，就称该软件组合覆盖该依赖关系，并且有关软件创建时所必需的软件版本信息在依赖关系中能够得到，软件之间的兼容性通过测试软件组合获得。

基于直接依赖关系的组合空间生成策略：对于一个正要创建的软件组合，建立直接依赖 $DD=(S_v,Dep)$ 就是建立 S_v 与 Dep 中的软件之间的直接依赖关系，最初它只有一个元素，就是直接依赖本身。下面获取 S_v 直接依赖的软件版本 Dep。在约束集的作用下广度优先遍历 SDG，选择满足约束集中条件的软件版本，将与 S_v 具有直接依赖关系的软件版本存放于 Dep，遍历终止于叶节点。广度优先遍历过程是：遍历未访问过的结点 n，访问该结点；从 n 出发，将与 n 邻接的所有结点 $w_1,w_2,w_3,\cdots,w_n$ 中的 $w_1.S_v,w_2.S_v,\cdots,w_n.S_v$ 软件放入 $n.Dep$，从约束集获取 $n.S_v$ 及 $w_1.S_v,w_2.S_v,\cdots,w_n.S_v$ 的所有版本，执行组合算法[9]，获取软件版本组合。然后，w_1，$w_2,w_3,\cdots,w_n$ 依次执行上述过程。

基于间接依赖的组合空间生成策略：根据间接依赖 $ID=(S_v,IDep)$ 生成软件组合空间就是创建所有与 S_v 直接依赖和间接依赖的软件版本，在满足约束集中条件的前提下，深度优先遍历 SDG，将与 S_v 依赖的软件存放于 $IDep$，直到 SDG 的所有结点遍历完为止。深度优先遍历的过程是：遍历未访问过的结点 n，对 n 赋访问标记，选择一条从 n 出发的边 (n,r)（若遇到 R 集的结点，则选择与 R 集结点连接的边）。若结点 r 已标记为访问过，则重新选择另一条从 n 出发的边，否则沿边 (n,r) 到达未曾访问过的 r；把 $r.S_v$ 放入 $n.IDep$，然后从 r 开始搜索，重复上述步骤；访问完所有从 r 出发可达的结点之后，回溯到结点 n，再选择一条从 n 出发的边，执行上述步骤，到所有 n 出发的边搜索结束。检查 $n.S_v$ 和 $n.IDep$ 中软件的所有版本范围，执行组合算法，得到软件版本组合。

3.2　执行树构造

已生成的所有国产基础软件组合可能会按照一定的顺序线性地在机器上进行测试，这样有可能会重复创建某些国产基础软件版本，降低测试执行效率。多个国产基础软件组合可能会包含相同的国产基础软件版本依赖关系，例如两个国产基础软件组合的第一个依赖关系相同，这样就可以复用这个依赖关系。所以本文将国产基础软件组合空间中的组合归并为执行树，这样相同的国产基础软件前缀可以被共享，减少国产基础软件版本重复创建的数量。执行树也就是国产基础软件的兼容性测试方案，其构造过程如下：

开始，执行树只有一个结点即树根结点，每个结点表示直接依赖 $DD=(S_v,Dep)$ 或者间接依赖 $ID=(S_v,IDep)$，按顺序将依赖关系放入前缀树结点中，这就形成了一个树形结构——执行树。结点 n 中的 S_v 用 $n.S_v$ 表示，Dep 和 $IDep$ 分别用 $n.Dep$ 和 $n.IDep$ 表示，放入顺序采用 SDG 中结点字母的字典序（用 $A,B,\cdots,Z$ 对 SDG 中的结点按照自顶向下编号）。按照树型结构，由前驱关系得，具有相同前驱的结点可以共享该前缀，这就减少了组合空间中的结点创建数目。

3.3　执行树执行策略

执行测试方案，就是执行前缀树中的每一个结点的任务。对于每一个前缀树结点中的 S_v，它必须在 Dep 或者 $IDep$ 中的软件创建之前创建。测试结点 n，因为 $n.Dep$ 和 $n.IDep$ 中的软件版本可能依赖于其它较底层的软件版本，所以必须保证 n 的祖先结点 $n.Dep$ 和 $n.IDep$ 中的软件版本已经创建。因此，从树根结点到结点 n 的路径中以结点表示的软件版本序列必须在测试结点 n 之前完成创建，该序列就是每个结点的任务。如果一个结点的任务执行成功，则说明与之依赖的所有版本的基础软件创建是成功的。这时采用缓存机制[10]存储该结点的执行序列的断点，该结点的子结点就可能复用父结点的执行任务，从而子结点就只需执行剩余的任务，节省了结点任务执行时间。

利用深度优先执行策略使得每个执行结点任务的客户端可以最大化地复用本地缓存的结点任务[11]。深度优先执行策略具体执行步骤如下：采用深度优先搜索顺序，客户端执行结点 n 的任务后请求下一个新任务；如果在测试方案中结点 n 是非叶结点，服务器将新任务分配给 n 的子结点，客户端先检查本地缓存的断点设置值，若可以复用，则该子结点只需创建该任务中与其直接依赖的软件版本；如果在测试方案中结点 n 是叶节点，则对应该分支的结点任务执行完毕，接着执行缓存中最近任务的结点的其余子结点。由于结点任务的执行是分配到多个客户端，因此对执行和未执行的结点要进行标记。接着按照以上的规则继续深度优先搜索测试方案中未测试的结点，直到将所有结点的任务执行完毕。

执行测试方案需要将每个结点的任务分配到多个客户端上运行，收集各个客户端的执行结果。国产基础软件在虚拟机下创建并运行于各个机器之上，这样可以使得每个机器的

状态不会被修改，以保证各个机器的可靠性[12]。对于软件的创建采用监测机制[13]，即对每个软件的安装进程进行错误监测，以防止安装错误。

在测试期间，如果 $n.S_v$ 不能在 $n.Dep$ 或 $n.IDep$ 之上正确地创建和运行，就说明 $n.S_v$ 与 $n.Dep$ 和 $n.IDep$ 中的软件版本不兼容，把不兼容信息作为进一步测试的依据[14]。基于依赖关系生成的组合空间，在测试过程中由于 $n.S_v$ 不能正确地创建可能会阻止 $n.S_v$ 的子结点的兼容性测试，因此要保证在测试每个结点之前创建它所有的祖先结点，且错误地创建 $n.S_v$ 不一定会对其它的国产基础软件兼容性测试造成影响。综上所述，选择不同的方式动态地创建额外的国产基础软件 $n.S_v$ 的组合来测试，可使组合空间最大化地覆盖具有依赖关系的国产基础软件间的组合。

4 工具支持和示例分析

4.1 工具支持

SCTSA(Software Compatibility Testing System Archetype)是基于 DRBA 建立的系统原型，它主要是按照基于依赖关系的国产基础软件兼容性测试方法的流程构建的系统，具体的流程包括：绘制 SDG、生成国产基础软件组合空间、构建执行树和运行执行树结点任务。SCTSA 系统的整体框架如图 3 所示。

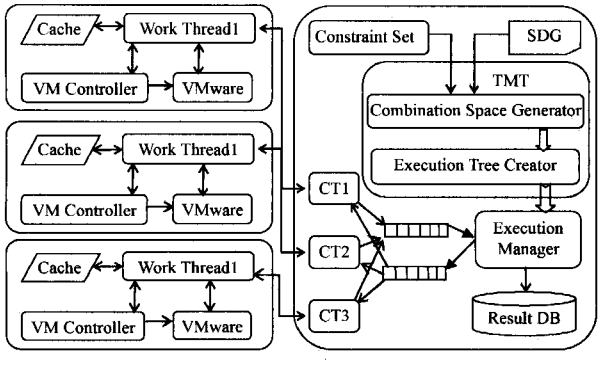

图 3 SCTSA 框架图

客户端负责执行测试方案和结果反馈。服务器端生成国产基础软件组合空间、构造执行树，然后将执行树结点的任务分配到各个客户端。框架具体的组成介绍如下：

1)服务器端(Server)是测试过程中的主要控制者。服务器在测试启动时依据 SDG 和约束集生成满足约束条件的国产基础软件组合空间，在所生成的国产基础软件组合空间的基础上构造执行树。服务器动态地控制分配测试任务给客户端，依据第 3.3 节提到的执行策略执行整个测试方案。

服务器会创建与客户端虚拟机进行交互的客户端线程 CT(Client Thread)和一个测试管理线程 TMT(Test Manager Thread)，TMT 的职责是生成国产基础软件的组合空间和构造执行树，CT 完成所有服务器与客户端的通信任务。当客户端发出第一个任务请求时，服务器为其创建 CT。当一个客户端执行完一个任务时，CT 将测试结果送入请求队列中，执行管理器(Execution Manager)抽取测试结果并存入 Result DB 中，以便最后分析数据。

2)客户端(Client)执行服务器分配的任务，每一个客户端都有一个 Work Thread 线程控制局部任务执行，同时与服务器对应的 CT 进行交互，接收分配的任务，返回测试结果。根据之前的描述，客户端的任务是根据依赖关系，利用虚拟硬件技术[15]建立基础软件的各个版本，即在虚拟机环境下，虚拟机提供虚拟的硬件层，基础软件运行于一个虚拟硬件环境之中。

虚拟机控制器(VM Controller)的主要功能是将建立软件版本的信息转换为机器命令，传送给 VM 建立虚拟基础软件。VM Controller 与 Work Thread 进行信息交换，Work Thread 把创建基础软件的信息传送到 VM Controller，VM Controller 将该信息转换为虚拟机能够接收的命令，传送到 VM，VM 创建相应的软件版本。

虚拟机缓存(VM Cache)用于缓存虚拟机执行任务时的状态，为其子结点复用，减少执行的时间，提高测试效率。

4.2 实例分析

结合本文提出的兼容性测试方法，将该方法应用到社保领域国产基础软件应用平台的兼容性测试中。其中，国产基础软件应用平台由操作系统(OS)、数据库软件(DB)、服务中间件(MW)、基础办公软件(Office) 4 类国产基础软件组成。按照不同品牌，每类国产基础软件构成的集合为 OS={Red Flag Linux, KylinOS, CS2CLinux}, DB={HUABASE, OpenBase, KingbaseES, DM}, MW={Tong-Tech, Infors, Kingdee}, $Office$={CS2C Office, YOZO Office, WPS Office}。操作系统 Red Flag Linux 有 5 个版本，KylinOS 有 3 个版本，标普华 Linux 有 4 个版本。数据库 KingbaseES 有 3 个版本，OpenBASE 有 5 个版本，数据库软件 DM 有 4 个版本，HUABASE 有 3 个版本。服务中间件东方通中间件有 2 个版本，中创中间件有 3 个版本，金蝶中间件有 4 个版本。基础办公软件中标普华 Office 有 3 个版本，永中 Office 有 4 个版本，WPS Office 有 5 个版本。若按照任意方式组合，需要测试的国产基础软件应用平台的组合数为 $C_3^3\times C_3^1\times(C_3^1+C_4^1+C_5^1)\times C_4^1\times(C_3^1+C_5^1+C_4^1+C_3^1)\times C_3^1\times(C_2^1+C_3^1+C_4^1)\times C_3^1\times(C_3^1+C_4^1+C_5^1)=2099520$ 种。若采用 DRBA，按照直接依赖关系需要测试的国产基础软件应用平台的组合数为(36+36+60)+(60+36+48)+(24+36+48)=388 种；按照间接依赖(假设中间件传递依赖于操作系统)关系，需测试的国产基础软件应用平台的组合数为(36+36+60)+(60+36+48)+(24+36+48)+108=496 种。从以上的示例可以看出，采用本文提出的 DRBA 方法，在保证覆盖每个国产基础软件的所有版本的前提下，有效地减少了国产基础软件应用平台的组合数目。

结束语 本文提出了一种利用国产基础软件之间的依赖关系的兼容性测试方法 DRBA，即通过国产基础软件间的依赖关系和前驱关系绘制国产基础软件 SDG，参考 SDG，通过覆盖准则确定国产基础软件应用平台组合空间，并在生成的组合空间基础上构造执行树，选取执行策略，按照一定的执行算法执行结点的任务。鉴于 DRBA 方法具有线性、易实现的特点，本文基于 DRBA 方法搭建了 SCTSA 系统原型，其可从国产基础软件间的直接依赖和间接依赖实现国产基础软件两两兼容性测试和集成后的兼容性测试。同时应用一个实例说明了 DRBA 方法的有效性。基于前面的工作，在以后的实验

和研究中需要设计更有效的执行方案执行策略，同时还需将研究工作扩展到国产基础软件的性能测试和可靠性测试等方面。

参 考 文 献

[1] 兰雨晴，赵同，高静，等. 基础软件平台质量评估[J]. 软件学报，2009，20(3)：567-582

[2] 高静，兰雨晴，金茂忠. 基础软件平台集成测试组合选择方法[J]. 北京航空航天大学学报，2010，36(3)：265-269

[3] Yoon I-C, Sussman A, Memon A, et al. Direct-dependency-based software compatibility testing[C]// Proc. of the 22st Int'l Conf. on Automated Software Engineering. Nov. 2007

[4] Mariani L, Papagiannakis S, Pezze M. Compatibility and Regression Testing of COTS-Component-Based Software, icse[C]// 29th International Conference on Software Engineering (ICSE'07). 2007：85-95

[5] Yoon I-C. Automating Software Compatibility Testing [C]// Proc. of the Int'l Symposium on Software Testing and Analysis. July 2008

[6] Van T-T, Vo B, Le B. Mining Sequential Rules Based on Prefix Tree[C]// New Challenges for Intelligent Information and Database Systems. 2011, SCI 351：147-156

[7] Czarnecki K, Helsen S, Eisenecker U. Formalizing cardinality-based feature models and their specialization[J]. Software Process：Improvement and Practice, 2005, 10(1)：7-29

[8] Syrjanen T. A rule-based formal model for software configuration[R]. A55. Helsinki University of Technology, Dec. 1999

[9] 温嘉佳，陈俊亮，彭泳. 基于目标距离评估的启发式Web Services组合算法[J]. 软件学报，2007(01)：85-93

[10] Lin Yi-bing, Lai Wei-ru, Chen J-J. Effects of Cache Mechanism on Wireless Data Access[J]. IEEE Transactions on Wireless Communications, 2003, 2(6)：1247-1258

[11] Schmidt T, Kuhn L, Price B, et al. A depth-first approach to target-value search[Z]. Palo Alto Research Center, 2009

[12] Yoon I-C, Sussman A, Memon A, et al: Effective and Scalable Software Compatibility Testing[C]// Proc. of the Int'l Symposium on Software Testing and Analysis. July 2008

[13] Oh N, Mitra S, McCluskey E J. Error Detection by Diverse Data and Duplicated Instructions[J]. IEEE Transaction on Computers, 2002, 51(2)：180-199

[14] VMware Inc. Streamlining software testing with IBM Rational and VMware[M]. Test lab automation solution-whitepaper, 2003

[15] Duarte A, Wagner G, Brasileiro F, et al. Multi-environment software testing on the Grid[C]// Proc. of the Workshop on Parallel and Distributed Systems：Testing and Debugging. July 2006

(上接第 106 页)

随着时间推移，可以发现网络可靠性渐近0.1，这是由于RWP移动模型中节点的非一致分布。RWP移动模型导致节点机终止于仿真区域的中间，因此源和目的节点间的可用路径减少。

结束语 移动自组织网络可靠性是影响物联网应用的数据采集和设备控制的关键因素之一。而移动自组织网络拓扑结构的动态变化和无线终端设备的能量限制等因素又直接影响着移动自组织网络的可靠性。要让物联网在更广泛的领域得到普及性应用，物联网感知层移动自组织网络可靠性理论与评估方面还存在很多技术难题需要突破。本文的研究成果对提升移动自组织网络节点的性能以及合理部署自组织网络节点具有指导意义，将为物联网信息全面感知奠定理论基础。

参 考 文 献

[1] 高飞，张少中，王光兴. 计算无线通信网络2-终点可靠性的快速算法[J]. 计算机学报，2007，30(6)：1035-1039

[2] 闵军，张海呈，朱桂斌. 自组网可靠性评价方法[J]. 电子科技大学学报，2008，37(3)：436-438

[3] Chen X, Lyu M R. Reliability analysis for various communication schemes in wireless CORBA[J]. IEEE Transactions on Reliability, 2005, 54(2)：232-242

[4] AboElFotoh H M F, Iyengar S S, Chakrabarty K. Computing Reliability and Message Delay for Cooperative Wireless Distributed Sensor Networks Subject to Random Failures[J]. IEEE Transaction on Reliability, 2005, 54(1)：145-155

[5] Andrew A D M, Snow P. Upkar Varshney. Reliability and Survivability of Wireless and Mobile Networks[J]. Computer, 2000, 33(7)：49-55

[6] Zhen Yan, Wu Mu-qing, Wu Da-peng, et al. Toward path reliability by using adaptive multi-path routing mechanism for multimedia service in mobile Ad-hoc network[J]. The Journal of China Universities of Posts and Telecommunications, 2010, 17(1)：93-100

[7] 何明，裘杭萍，鲍广宇，等. 基于多径路由的无线Mesh网可靠性评估[J]. 应用科学学报，2009，27(5)：441-445

[8] 赵蕴龙，单宝龙，高振国，等. 无线Mesh网骨干层2-终端可靠性计算策略[J]. 计算机学报，2009，32(3)：424-431

[9] Abdule S M, Hassan S, Ghazali O, et al. The effect of speed on AODV protocol performance using RWP in Ad hoc network[C]// Proceedings of 2010 International Conference on Broadcast Technology and Multimedia Communication (Volume5). Wuhan：IITA, 2010

[10] Kalmanek C R, Ge I, Lee S. Using exploratory data mining to raise the bar on network reliability and performance[C]// Design of Reliable Communication Networks, 2009. Washington：IEEE Press, 2009：1-10

[11] 李海波，蔡一兵，李忠诚，等. 一种基于节点间距离提高Ad hoc路由稳定性的方法[J]. 系统仿真学报，2007，19(10)：2374-2378