

Reachability analysis of web service compositions via NWA

Du Xutao¹ Xing Chunxiao^{2,3} Zhou Lizhu¹

(¹Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

(²Research Institute of Information Technology, Tsinghua University, Beijing 100084, China)

(³Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: In order to improve the design and implementation quality of web service compositions, formal methods are used to model them and certain properties are verified. WCFA (web service interface control flow automata) is used to model web services, especially the control flow and possible interactions with other web services. A web service composition consists of a set of interacting WCFA. The global behavior of web service compositions is captured by NWA (nested word automata). A variation of the depth-first search algorithm is used to transform a set of WCFA into an NWA. State formulae and call stacks at each node of NWA are computed by a path-sensitive reachability analysis. Safety properties, call stack inspection properties and pre/post-conditions of service invocations are described by assertions. Then verification of these assertions is carried out by an automated SAT tool.

Key words: web service composition; formalism; nested word automata (NWA); web service interface control flow automata (WCFA); verification

Web services are intended to be independently developed software components, which are to collaborated to accomplish complex tasks. Since the quality of web service compositions is vital for serious applications, we need to ensure general properties (e. g., safety properties) of web service compositions.

For such purposes, inspired by web service interface formalism^[1], we developed the concept of WCFA (web service interface control flow automata)^[2] as an interface-level description of web services. WCFA supports sequential composition, parallel composition and branching structure.

NWA (nested word automata)^[3] is able to describe both linear and hierarchical program structures without using an unbound stack. We use NWA to model the global behavior of web service compositions. Safety properties, call stack inspection properties and pre/post-conditions of certain web service invocations are declared as assertions (quantifier-free propositional predicates) in states of NWA. Then a SAT solver (CVC3)^[4] is used to check the validity of the assertions under the assumptions of the state formula (for safety property) or the call stack (for call stack inspection property) in the same state.

The difference between our method and other methods^[5-7]

is that our formalism does not look at details of implementation but focuses on web service invocation behavior. Our method can be used at the very beginning of the design stage of web service compositions and can help to detect design errors as early as possible. Therefore, our method can be beneficial to the designers.

1 Web Service Interface Control Flow Automata

1.1 The concept of WCFA

We introduce the concept of WCFA through the example of the TripRequest web service. Fig. 1 gives the WCFA of a travel agency's TripRequest service. A WCFA is a directed acyclic graph. Nodes are control points of the web service, which have two fields and are depicted by a tuple $\langle n, t \rangle$. The number field n is a unique number for the control point. The type field $t \in NT = \{\text{start}, \text{sq}, \text{test}, ++, +-, *, -, \text{rt}\}$ describes the type of the control point.

A start node is the start point of a web service, and an rt node is the return point. An sq node has a sequential property and has one input edge and one output edge. A test node has one input edge and possibly several output edges, only one of which will be chosen according to the labels on the edges. A ++ (+) node is the start node for a wait-for-one (wait-for-all) parallel composition. Every *+ (++) node has a corresponding collecting node whose type is *- (+-). A *- (+-) node has several input edges and one output edge. Edges are web service invocations (possibly with pre/post-conditions), predicate tests or assignments.

The travel agency provides the TripRequest service which actively accepts a customer's request to order a trip. Fig. 1 is rather self-explainable. We want to stress that at $\langle T\#3, *+ \rangle$, two parallel threads are created to invoke two external web services, FlightReserve and HotelReserve, respectively. The returned values of them are stored in O_FlightReserve and O_HotelReserve, respectively. After both of them return, the control flows to $\langle T\#8, *- \rangle$.

1.2 A design error

We assume that the HotelReserve service needs to invoke the BankPay service for a successful reservation of rooms. However, the TripRequest service will only return OK when both HotelReserve and the FlightReserve services succeed. Therefore, when HotelReserve succeeds (i. e., has paid for the room reservation) but FlightReserve fails, the TripRequest service will return FAIL. This design error can be easily corrected once detected. We are going to develop a method to detect such errors.

2 Constructing NWA from WCFA

For the verification of web service compositions, we need

Received 2008-04-15.

Biographies: Du Xutao (1977—), male, graduate; Xing Chunxiao (corresponding author), male, doctor, professor, xingcx@tsinghua.edu.cn.

Foundation items: The National Key Technology R&D Program of China during the 11th Five-Year Plan Period (No. 2006BAH02A12), the National High Technology Research and Development Program of China (863 Program) (No. 2006AA010101).

Citation: Du Xutao, Xing Chunxiao, Zhou Lizhu. Reachability analysis of web service compositions via NWA [J]. Journal of Southeast University (English Edition), 2008, 24(3): 293 – 295.

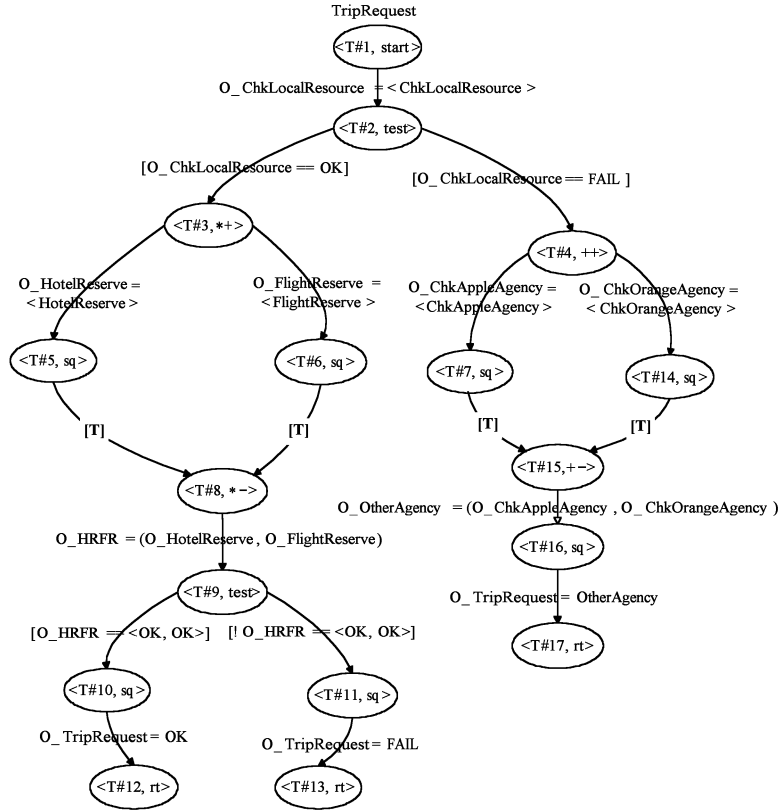


Fig. 1 WCFA for TripRequest

to understand their global behavior. NWA is used for such purposes. When we design a web service composition, some of the involved web services may have their WCFA, and others may not have it. Depending on the availability of WCFA, we can build an NWA of a web service composition. If some invoked service has its WCFA, it will be expanded in the NWA. Every state (node) in the NWA has a state formula describing the current running state of the web service. The state formulae are obtained by a path-sensitive static analysis.

2.1 Nested word automata

An NWA^[3] is a powerful formula that can be used for software verification. An NWA can capture both linear and hierarchical structures. Linear states and internal-transitions in NWA can model the variable states and normal (e. g. assignment) activities smoothly. Hierarchical states associated with call-transitions and return-transitions can model the web service calling activities and the call stack of a web service composition.

2.2 Algorithm WCFA2NWA

The algorithm WCFA2NWA, that transforms a web service composition (described by a set of the WCFA) into an NWA, starts from the root of the WCFA of the composition. Other information such as the current call stack and state formulae also serves as parameters.

When computing the NWA, we need to deal with every outward edge from the node being processed (say m) to another node (say m'). If m is not a parallel composition node (i. e. the type is either $*+$ or $++$), we consider two cases. If the outgoing edge is a service call $VA = \langle EMT \rangle$ which

does not have a WCFA, or a predicate test $[p]$, or a normal assignment $VN = ST$, then we recursively call WCFA2NWA to compute $NWA_{m'}$. If the outgoing edge is a service call $VB = \langle IMT \rangle$ with a corresponding $WCFA_{IMT}$, then we need to expand it by recursively calling WCFA2NWA to obtain the NWA_i . Then the linear call-transition is created. For each return nodes (say rn) in the NWA_i , we recursively call WCFA2NWA to compute $NWA_{m'-rn}$, and the return-transitions are created. Then hierarchical call-transitions are added.

When dealing with parallel composition nodes (i. e., the type is $*+$ or $++$), we take the interleaving semantics. Each permutation of the outward edges is viewed as a sequential composition and processed by our algorithm in order to construct the NWA. During this process, the state formula and call stack of the current state should be updated and passed to WCFA2NWA as parameters.

3 Verification Based on NWA

When the NWA for a web service composition is constructed, we can use it to perform the verification. Assertions are used to describe certain properties. The structure of the NWA allows us to naturally express such properties at the states of the NWA. We use CVC3^[4] as our SAT solver. CVC3 supports several built-in base theories.

The verification process is as follows:

- 1) Take the state formula (and/or call stack) at the node as Γ ;
- 2) Take the property assertion (e. g. safety property assertion) at that node as α ;
- 3) Use the SAT solver to check whether $\Gamma \models \alpha$.

Since there is no quantifier, CVC3 is a decision procedure which means that CVC3 can correctly answer whether α is

valid under the assumption Γ or not. If CVC3 returns valid, it means that the assertion is a logical consequence of the state formula (or the call stack). If CVC3 returns invalid, it means that the assertion cannot be verified at this node, and we can find an error in the design.

For example, we use the assertion $[\neg O_RoomPay == OK]$ at $\langle T\#13, rt \rangle$ (a return node in $NWA_{TripRequest}$) to describe the safety property that on a FAIL return node, hotel reservations should not be paid. The state formula at that node contains the path formula:

$$\{ ([O_RoomPay == OK], O_HotelReserve == OK), \\ ([O_RoomPay == FAIL], O_HotelReserve == FAIL) \}$$

This path formula says that $\langle T\#13, rt \rangle$ can be reached through at least two paths. One of them takes both $O_RoomPay$ and $O_HotelReserve$ to be OK and the other one takes both of the two variables to be FAIL. When checking whether the assertion is the logical consequence of the state formula in CVC3, the answer will be “invalid”. Therefore, we have found an error in our design of TripRequest.

4 Conclusion

NWA^[3] is used for the modeling and verification of web service compositions. Web services are described by WCFA^[2]. An algorithm for transforming a set of WCFA to an NWA is presented. States in NWA are equipped with a state formula and a call stack. They describe relevant information of the web service composition when control flows to certain states.

Safety properties, call stack inspection properties and pre/post-conditions are described by quantifier-free assertions on states of NWA. Then a SAT solver is used to check whether the assertion is the logical consequence of the state formula

(or call stack). Since the SAT solver is a decision procedure for quantifier-free validity checking, the result can help the designers decide whether their design of the web service composition is correct or not.

References

- [1] Beyer D, Chakrabarti A, Henzinger T A. Web service interfaces [C]//*Proceedings of the 14th International World Wide Web Conference*. New York: ACM, 2005: 148 – 159.
- [2] Du Xutao, Xing Chunxiao, Zhou Lizhu. Abstract reachability graph for verifying web service interfaces [C]//*Proceedings of the 10th International Conference on Software Reuse*. Beijing, China, 2008: 262 – 265.
- [3] Alur R, Madhusudan P. Adding nesting structure to words [C]//*Proceedings of the 10th International Conference on Developments in Language Theory*. Santa Barbara, 2006: 1 – 13.
- [4] Barrett C, Berezin S. CVC lite: a new implementation of the cooperating validity checker [C]//*Proceedings of the 16th International Conference on Computer Aided Verification*. Boston, 2004: 515 – 518.
- [5] Fu X, Bultan T, Su J. Analysis of interacting BPEL web services [C]//*Proceedings of the 13th International World Wide Web Conference*. New York: ACM, 2004: 621 – 630.
- [6] Foster H, Uchitel S, Magee J, et al. LTSA-WS: a tool for model-based verification of web service compositions and choreography [C]//*Proceedings of the 28th International Conference on Software Engineering*. Shanghai, China, 2006: 771 – 774.
- [7] Ouyang C, Verbeek H, van der Aalst W, et al. Wofbpel: a tool for automated analysis of BPEL processes [C]//*Proceedings of the 3rd International Conference on Service-Oriented Computing (ICSOC)*. Amsterdam, Netherlands, 2005: 484 – 489.

使用 NWA 对组合 web 服务进行可达性分析

杜旭涛¹ 邢春晓^{2,3} 周立柱¹

(¹ 清华大学计算机科学与技术系, 北京 100084)

(² 清华大学信息技术研究院, 北京 100084)

(³ 清华大学清华信息科学与技术国家实验室, 北京 100084)

摘要: 为了提高组合 web 服务的设计和实现质量, 使用形式化方法对其进行建模并对其关键性质进行验证. 使用 web 服务接口控制流自动机 (WCFA) 对 web 服务进行建模, 主要描述其控制流及与其他 web 服务的交互关系. 组合 web 服务由一组交互的 WCFA 组成. 使用嵌套字自动机 (NWA) 对组合 web 服务的整体行为进行建模. 将一组 WCFA 转换为嵌套字自动机 (NWA) 的算法是深度优先搜索算法的变种, 算法中使用路径相关的可达性分析计算 NWA 的每个节点的状态公式和调用栈. 安全性相关性质、调用栈相关性质及服务调用的前置和后置条件都可以用断言来描述, 然后使用一个自动的可满足性 (SAT) 求解工具对这些断言进行验证.

关键词: 组合 web 服务; 形式化方法; 嵌套字自动机; web 服务接口控制流自动机; 验证

中图分类号: TP311