Transaction Orchestration

Jayadev Misra
The University of Texas at Austin, Austin, Texas, USA

The internet promises a truly global computer where all services and data are available to all users at all time. True internet computing, which we call wide-area computing, will combine multiple available services in much the same way as we compose functions by combining individual statements in a sequential program. However, it is astoundingly difficult to program the "internet computer" for seemingly simple tasks. Consider a university department in which the calendars of all professors are available online. Also available are the room reservation charts. It is still a major software en-gineering task to automatically schedule a meeting among a group of professors and reserve a room where the meeting is to be held. Add to this the reasonable requirements that (1) the staff should be contacted to provide the appropriate audio-visual equipment for the meeting, (2) the meeting should be rescheduled if more than one third of the members later decide that they can not meet at that time, and (3) a new room should be found if the assigned room is later preempted for a meeting by the chairman of the department.

This small example embodies the computational pattern inherent in many wide-area applications: acquire data from one or more remote services, calculate with these data, and invoke yet other remote services with the results. Additionally, it should be possible to invoke alternate services for the same computation to guard against failure of one service, repeatedly poll a service until it supplies the results which meet certain desired criteria, or ask a service to notify the user when it acquires the appropriate data. It should be possible to download an application and invoke it locally, or have a service call another service on behalf of the user. Cardelli identifies "concurrent flow", or orchestration, as one of the two outstanding problems in internet computing (the other being management of semistructured data).

We propose a novel approach for combining different web services, as well as gen-eral distributed transactions, to form complex services over the internet. The resulting services may be regarded as general purpose programs for the internet, which would acquire data from remote locations, apply (remote) services to them, pipe the results to yet other remote services, poll services and data until some desired criterion is met, or pursue alternative computation strategies to select an appropriate result. The notions of data and service migration, application discovery, and downloading of services for local executions are totally transparent to the users and other applications.

The proposal is based on a strong theoretical foundation; a computation is specified by an expression of Kleene algebra (the algebra of regular expressions). The primitive symbols are web services and transactions, which are combined using the operators of the algebra. Atomicity of transactions justify treating each symbol as an (atomic) in-struction in a larger program. The laws of Kleene algebra and its relationship to finite state machines provide the basis for efficient implementation and program optimiza-tion. The programming model exploits the power of transaction processing, though the complexities of transactions are hidden from the users. An implementation can be hosted easily in traditional programming languages like Java and C++.

References

1. Jim Gray and Andreas Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993.
2. C.A.R. Hoare. Communicating Sequential Processes. Prentice Hall International, 1984.
3. Dexter Kozen. On Kleene algebras and closed semirings. In Proceedings, Math. Found. of Comput. Sci., volume 452 of Lecture Notes in Computer Science, pages 26–47. Springer-Verlag, 1990.
4. Bob Laskey and James Parker. Building a reverse auction with biztalk orches-tration. `http://msdn.microsoft.com/xml/articles/biztalk/bizorchestr.asp` , Aug 2000.
5. R. Milner. Communication and Concurrency. International Series in Computer Science, C.A.R. Hoare, series editor. Prentice-Hall International, 1989.
6. A web site on BizTalk. Available for download at `http://www-106.ibm.com/developerworks/webservices/library/ws-bpel`
7. A web site on BPEL. Available for download at `http://www-106.ibm.com/developerworks/webservices/library/ws-autobp`
8. A web site on BPEL. Available for download at `http://www-106.ibm.com/developerworks/webservices/library/ws-bpel`