Composing Personalised Services on top of Abstract State Services

(Extended Abstract)

Hui Ma[1] Klaus-Dieter Schewe[2] Bernhard Thalheim[3] Qing Wang[4]

[1] Victoria University Wellington, School of Mathematics, Statistics and Computer Science, Wellington, New Zealand, Email: huimanz@gmail.com
[2] Information Science Research Centre,Palmerston North, New Zealand
Email: kdschewe@acm.org
[3] Christian-Albrechts-University Kiel, Institute of Computer Science, Kiel, Germany
Email: thalheim@is.informatik.uni-kiel.de
[4] Massey University, Palmerston North, New Zealand
Email: q.q.wang@massey.ac.nz

Since its introduction the role of the world-wide web has shifted from enabling access to a pool of documents to the provision of services. Such web services can in fact be anything: a simple function, a data warehouse, or a fully functional Web Information System. The unifying characteristic is that content and functionality are made available for use by human users or other services. Therefore, web service integration has become a highly relevant research topic, and a lot of work has been investigated into it.

In this paper we take a more abstract, conceptual approach to service integration, composition and personalisation. We adopt the fundamental idea from the area of dialogue systems that a service can be described by two layers: a hidden database layer consisting of a database schema and transactions, and a visible view layer on top of it providing views and functions based on them. This idea has already been mirrored in development methods for Web Information Systems, and also appears as a natural choice for component-based systems development.

However, we want to go one step further and develop a theory of Abstract State Services (ASSs) following the line of thought of the ASM thesis. Gurevich and Blass formalised sequential and parallel algorithms by requiring a small set of intuitive, abstract postulates to be satisfied, then proved that these postulates are always satisfiable by (sequential) ASMs, i.e. ASMs capture algorithms in the most general sense in a natural way. This has been picked up by Wang and Schewe and refined for database transformations exploiting states as meta-finite logical structures. This research can be used as a basis for the model of transactions on the database level, and thus forms the basis of the formal definition of ASSs. In doing so the web as the medium through which a service may become available is of no importance; the notion of ASS can also be applied to enterprise services that are only available to selected clients.

We then address the problem of service integration and composition. Integration means to replace two or more ASSs by a single new one that offers all the

Dagstuhl Seminar Proceedings 08181
The Evolution of Conceptual Modeling
http://drops.dagstuhl.de/opus/volltexte/2008/1597

functionality of the individual services. We show that this problem can actually be reduced to database schema and view integration. The second one requires the extraction of service components from existing ASSs that feed a new service without replacing the original ones. For this we can adopt ideas from component composition .

In a second step we address the problem of service personalisation according to preference rules. For this we pick up the idea from WIS personalisation to compose personalised tasks, where the preference rules indicate, which choices will be preferred.