Extensions of Event Based B for Development of Grid Systems

Pontus Boström and Marina Waldén

Åbo Akademi University, department of Computer Science
Turku Centre for Computer Science (TUCS)
Lemminkäisenkatu 14 A, 20520 Turku, Finland
E-mail:{Pontus.Bostrom, Marina.Walden}@abo.fi

1 Introduction

Computational grids have become widespread in organizations for handling the vast amount of available information and their need for computational resources. These grid systems are often complex and special care need to be taken in order to ensure their correctness. We propose an extension of Event based B [2], in order to enable their formal implementation. The extensions provide new constructs to take into account the client-server architecture of grid systems, as well as important features like communication and synchronisation.

2 A language for distributed systems

Organizations need the ability to efficiently utilise existing hardware and be able to effectively share information with each other. Computational grids have become a popular approach to enable organizations to handle the vast amount of available information. Grid computing is a distributed computing paradigm that differ from traditional distributed computing in that it is aimed toward large scale systems that even span organizational boundaries. Recently standards such as Open Grid Services Infrastructure (OGSI) [5] and toolkits such as Globus toolkit [4] for implementing those standards have been developed in order to simplify the construction of complex grid systems.

The standards above do, however, not help in designing and implementing *correct* grid systems. The use of formal methods are needed in order to ensure their correctness. The Action System formalism [3] is a formal method that is well suited for developing large distributed systems, since it support stepwise development. However, it lacks tool support. The B

Method [1] is a formal method originally developed for construction of sequential programs and has good tool support. The B Method can be combined with Action Systems in order to formally reason about distributed systems as in the related methods B Action Systems [6] and Event based B [2]. B Action Systems models Action Systems in ordinary B, while the later Event based B also extends traditional B with some new constructs. With generic formal languages it is, however, possible to construct abstract models and specifications that are not implementable or very difficult to implement efficiently. The problem becomes especially apparent when developing distributed systems with complicated synchronization and communication patterns. Therefore, we propose new extensions to Event based B in order to be able to implement grid systems and verify their correctness.

The language extensions we propose for Event based B are targeted towards Grid systems using the Globus toolkit [4] middleware. Grid systems usually have a client-server architecture. This means that there is a client that initiates communication with the server, which only responds to the clients request. A client may access several servers simultaneously. Therefore, the language has to support client-server architectures with multiple concurrent accesses by the same client to several servers. The main communication mechanism of the grid middleware is remote procedure calls. However, the grid middleware also supports asynchronous notifications sent from a server to a client. Hence, the language should support both these communication primitives.

In grid systems a server is usually referred to as a grid service, since it provides services to other grid components. In order to meet the language requirements above we propose to extend Event based B with two types of machines, a *grid service machine* modelling abstract grid service features and a *grid refinement machine* for refining grid service machines or for introducing grid features to ordinary Event based B models. A grid service machine is a template of which the clients can obtain instances. The client can control several instances of the same grid service machine as a master can control several identical worker nodes. A grid service machine contains specifications of remote procedures, actions and notifications. The grid refinement machine on the other hand has clauses for refined remote procedures and actions, as well as statements for handling notifications. The clients and servers, i.e. grid services, use remote procedure calls and notifications to communicate and synchronize with each other. A client makes a requests to a grid service with a remote procedure call. When the request has been carried out a notification is sent back to the client. While the client waits for a notification from a grid service, it cannot make a new remote procedure call to the same grid service.

The development of the grid system, shown in Figure1 starts with an initial specification in Event B, *D0,* that is refined in a number of steps, *D1*. The specification is then split up into a client, *D2*, and a number of grid services, *E0*. The grid services can in turn be independently refined further, *E1*, and reference new grid services, *F0*. Throughout the development of the system the grid constructs are translated to ordinary B machines for verification purposes.

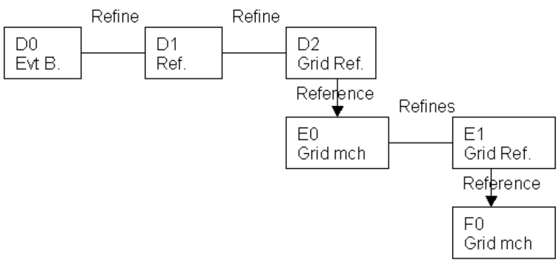

Figure 1: The structure of the grid system development

3 Conclusions

We have proposed an extension of Event based B to facilitate the development of correct grid systems. Grid systems are large distributed systems and standard development tools cannot guarantee their correct implementation. Our extension can provide a convenient formal development process for such distributed systems. The systems will by construction have an architecture that is implementable. Furthermore, specifications of grid systems constructed in this language will be clear to understand, since the systems are modeled in terms of grid primitives with a precise meaning. We believe that this approach to adapt B to the Globus Toolkit middleware can also be useful for other types of middleware for distributed systems.

References

[1] Abrial, J. -R. 1996. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press.

[2] Abrial, J. -R. and Mussat, L. 2001. (06.09.2004) *Event B Reference Manual*. http://www.atelierb.societe.com/ressources/evt2b/eventb_reference_manual.pdf

[3] Back, R. J. R. and Kurki-Suonio, R. 1983. Decentralization of process nets with centralized control. In *Proceedings of the 2nd ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 131-142.

[4] The Globus Alliance. 2003. (06.09.2004) Globus Toolkit. http://www.globus.org/

[5] Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Snelling, D. and Vanderbilt, P. 2003. (06.09.2004) Open Grid Services Infrastructure. http://www.ggf.org/documents/final.htm

[6] Waldén, M. and Sere, K. 1998. Reasoning About Action Systems Using the B-Method. *Formal Methods in Systems Design* 13, 5-35. Kluwer Academic Publishers.