The Hierarchical Evaluation Tool HIT

H. Beilner, J. Mäter, C. Wysocki
Universität Dortmund, Informatik IV
D-44221 Dortmund, Germany
e-mail: hit@ls4.informatik.uni-dortmund.de

Abstract: The software tool HIT provides for model-based performance evaluation of computing and communication systems during all phases of their life cycle. Specification of (models of) dynamic, discrete-event, stochastic systems is achieved by particular language- and graphics-based description options. Performance evaluation of accordingly specified models is supported by a variety of techniques of the simulative and analytical types.

Keywords: Hierarchical model structure, homogeneous model description, different solution methods, heterogeneous modelling, modelling environment

1 Properties/Model World

HIT is a comprehensive software tool supporting the model-based evaluation of computing system performance. HIT models exhibit a highly structured view of the systems to be assessed, based on (vertical) functional hierarchies and (horizontal) modularization as employed in modern software engineering and hardware architecture approaches. The HIT model world is tailored upon the prevailing view of computing system structures as partitioning a system

- vertically, into a sequence of layers and levels, communicating via function calls, and jointly representing a hierarchy of virtual machines;
- horizontally, into independent, mutually well-protected, information-hiding modules, each realizing some subset of functions to be provided at a particular level.

Although originally developed for evaluating computing system performance, HIT also lends itself to the analysis of "similar" systems such as communication and office systems, transport and logistic systems and others, all of the dynamic, discrete-event, stochastic type.

2 Specification

The specification of models is supported by a particular model description language, HI-SLANG, and/or a graphical interface, HITGRAPHIC.

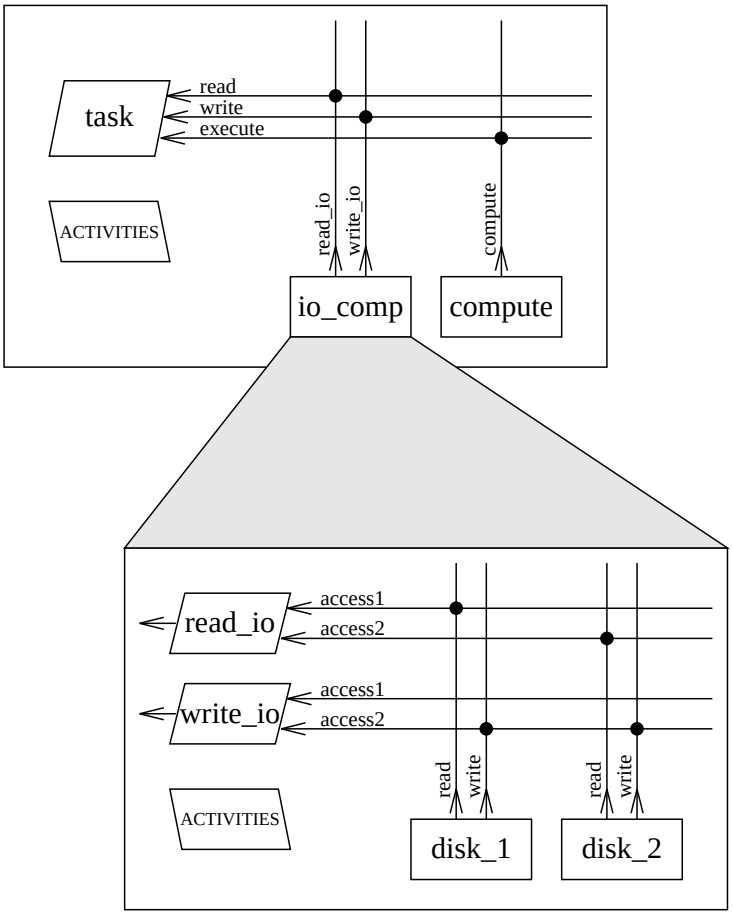

Figure 1: HITGRAPHIC User Interface

The HI-SLANG specification maintains, as far as possible, a high-level language (HLL) approach, well-known to and convenient for the perceived user community of the tool:

- Conventional functions (termed SERVICEs) serve as "patterns" for processes "to-be-run". They are described in terms of HLL control and data structures.

- SERVICEs can be packaged into and exported by modules (termed COMPONENTs). Services can be called upon by other services, from within higher layer COMPONENTs.

- Options for initiating processes in time-controlled or event-controlled mode complete the specification capabilities for describing systems of parallel processes.

Within a COMPONENT, the image of a LOAD to be executed on a MACHINE is enforced, with all SERVICEs contained in this COMPONENT describing the particular load (behaviour) pattern, and all COMPONENTs being used by this COMPONENT (via using their exported SERVICEs) constituting the particular machine base.

The HITGRAPHIC specification (cf. Figure 1) relieves the user of detailed work when

- specifying the static structure of hierarchical models,
- creating LOADs and MACHINEs consisting of SERVICEs and COMPONENTs, respectively,
- binding LOADs to MACHINEs,
- specifying hierarchies of LOAD pathes, and
- specifying selective evaluating/measuring options for performance metrics.

Disjoint specifications of models (to be analysed) and experiments (to be performed with these models) greatly increase the flexibility of use. Additionally, a model bank (OMA) supports storage and retrieval of (parts of) models and analysis results.

3 Analysis

HIT evaluation techniques include the following approaches:

- stochastic discrete-event simulation with appropriate statistical result evaluation;
- exact result evaluation for "separable networks" (with product-form solution) and approximate evaluation techniques for both "large" separable and certain "non-product-form" networks;
- numerical evaluation of Markov chain representations of more general models;
- submodel analysis and aggregation with the objective of generating "equivalent" higher level representations, to be used in structured and/or heterogeneous (total model) evaluation.

A HIT model specification is not directly influenced by the particular evaluation technique to be employed. There does, of course, exist an indirect influence whereby certain models will turn out not to be tractable by one or the other analysis technique, with simulation clearly offering the largest spectrum.

4 Environment

HIT is implemented in SIMULA. Given an appropriate SIMULA compiler, HIT/HIT-GRAPHIC is currently operational for the following systems (cf. Figure 2): Sun (Sun3 and SPARC), Apollo, DEC workstations, Siemens mainframes (BS2000), most PC-'386 (ix, SCO/UNIX, SINIX). HITGRAPHIC, the graphical user interface, is implemented in C on top of the X Window System.

HIT has been developed at the University of Dortmund. The development was initially supported by Nixdorf Computer AG and BMFT.

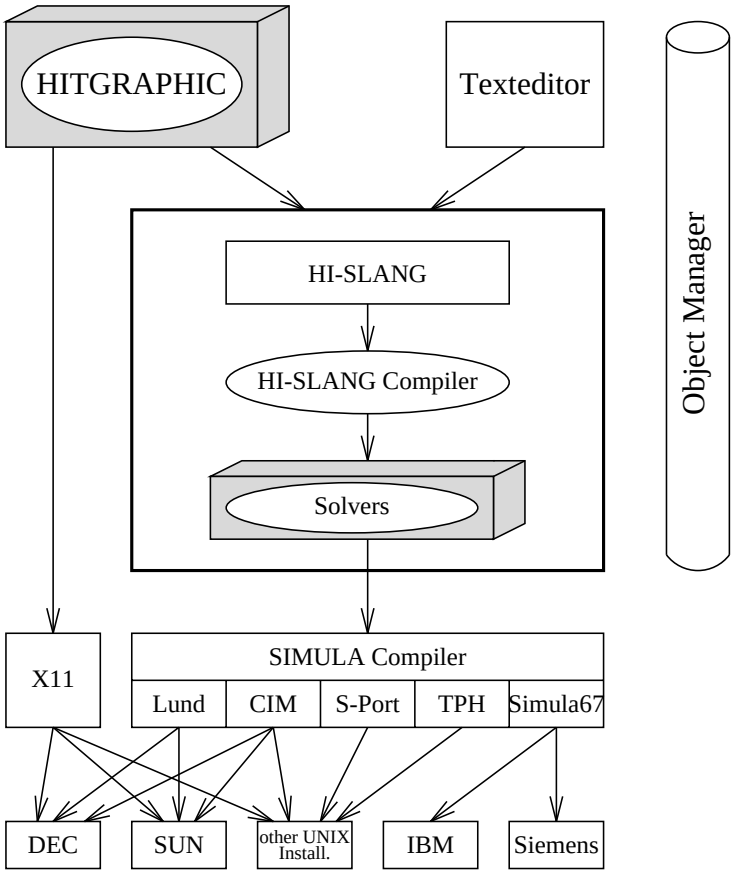

Figure 2: HIT Architecture

References

[1] H. Beilner, J. Mäter, and N. Weißenberg, *Towards a Performance Modelling Environment: News on HIT*, Proc. 4th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation, Palma de Mallorca, 1988.

[2] H. Beilner and F.-J. Stewing, *Concepts and Techniques of the Performance Modelling Tool HIT*, Proc. European Simulation Multiconference, Vienna, 1987.

[3] M. Büttner (ed.), *HIT and HI-SLANG, An Introduction*, Universität Dortmund, Lehrstuhl Informatik IV, 1992.

[4] M. Büttner (ed.), *HI-SLANG Reference Manual*, Universität Dortmund, Lehrstuhl Informatik IV, 1993.

[5] M. Sczittnick (ed.), *HITGRAPHIC User's Guide*, Universität Dortmund, Lehrstuhl Informatik IV, 1993.

[6] N. Weißenberg (ed.), *HIT-OMA User's Guide, Standard Object Manager of the Hierarchical Evaluation Tool HIT*, Universität Dortmund, Lehrstuhl Informatik IV, 1992.