A Graphical Environment for the Semantic Validation of a Plan Execution Language

(Extended Abstract)

Héctor Cadavid
Escuela Colombiana de Ingeniería
AK. 45 No. 2005–59, Bogotá, Colombia
Email: hcadavid@escuelaing.edu.co

César Muñoz
National Institute of Aerospace
100 Exploration Way, Hampton, VA, 23666, USA
Email: munoz@nianet.org

Camilo Rocha
University of Illinois at Urbana-Champaign
2111 Siebel Center, 201 N Goodwin Ave, Urbana, IL 61801, USA
Email: hrochan2@cs.uiuc.edu

***Abstract*—This paper presents a graphical environment that provides a user-friendly interface to the formal operational semantics of PLEXIL, a plan execution language developed by NASA to support autonomous space operations. This environment serves as a testbed for developers of a PLEXIL's executive system to validate, maintain, and debug the implementation of the system against the formal semantics of the language.**

The Plan Execution Interchange Language (PLEXIL) [1] is a high-level plan execution language developed by NASA that supports autonomous spacecraft operations. The *Universal Executive* (UE) is a system that interprets and executes PLEXIL plans.[1] An executive, such as UE, is a complex piece of software. It has been designed to be deployed in multiple platforms, usually with limited computational resources and under uncertain physical conditions.

Given the critical nature of spacecraft operations, PLEXIL's operational semantics has been formally defined [2] and several properties of the language have been mechanically verified [3] in the Program Verification System (PVS) [4]. This semantics has been also implemented in the formal notation of Maude [5], a high-performance implementation of the rewriting logic framework [6]. Although the development of the executive was guided by the formal semantics of the language, it has not been formally verified that it correctly implements the semantics.

The formal semantics of PLEXIL is organized as a stack of five abstract relations, which range from an atomic relation describing the evolution of a single computational element of PLEXIL to an execution relation describing the evolution of the whole plan after the occurrence of a series of external events. For efficiency reasons, an executive system may profit from properties of the language, such as determinism and compositionality, to implement these relations in a different way. Therefore, there may not be a one-to-one relation between the formal semantics and the executive implementation. Furthermore, a discrepancy between the executive and the formal semantics does not necessarily mean that the executive is *incorrect*. After all, the language is still evolving and the executive serves as an implementation of the intended semantics.

We have developed a graphical environment where PLEXIL developers can validate the formal semantics of the language against an intended semantics, such as an executive. The graphical environment consists of visualization software, written in Java, the formal operational semantics of PLEXIL, written in Maude, and bidirectional translator from Maude syntax to Java objects. The environment provides a user-friendly interface to the step-by-step evaluation of a PLEXIL plan for a recorded sequence of external events. Furthermore, it allows for inspection of the internal state and execution status, backtracking, traceability, and cross-reference to the formal semantics.

Our current implementation is a proof of concept. Although it does not support yet all the syntactic elements of the language, we have already discovered, and fixed, a semantic rule in the language that deals with assignment of local variables. In addition to extending the support to full PLEXIL, we also plan to integrate into the environment formal analysis capabilities provided by Maude, such as model checking and theorem proving. This will enable the formal verification of properties for a particular plan under a given sequence of events.

During the initial evolution of the language, we expect that this environment will become an important tool for PLEXIL's designers. In the long term, we expect that this tool will become a formal debugging environment of PLEXIL.

[1]The PLEXIL executive system is electronically available at http://plexil.wiki.sourceforge.net.

REFERENCES

[1] V. Verma, A. Jónsson, C. S. Păsăreanu, and M. Iatauro, "Universal executive and PLEXIL: Engine and language for robust spacecraft control and operations," in *American Institute of Aeronautics and Astronautics Space 2006 Conference*, 2006.

[2] G. Dowek, C. Muñoz, and C. Păsăreanu, "A small-step semantics of PLEXIL," National Institute of Aerospace, Hampton, VA, Technical Report 2008-11, 2008.

[3] ——, "A formal analysis framework for PLEXIL," in *Proceedings of 3rd Workshop on Planning and Plan Execution for Real-World Systems*, September 2007.

[4] S. Owre, J. Rushby, and N. Shankar, "PVS: A prototype verification system," in *11th International Conference on Automated Deduction (CADE)*, ser. Lecture Notes in Artificial Intelligence, D. Kapur, Ed., vol. 607. Saratoga, NY: Springer-Verlag, Jun. 1992, pp. 748–752.

[5] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott, Eds., *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, ser. Lecture Notes in Computer Science, vol. 4350. Springer, 2007.

[6] J. Meseguer, "Conditional rewriting logic as a unified model of concurrency," *Theoretical Computer Science*, vol. 96, no. 1, pp. 73–155, 1992.