

Towards *Hermes/dl*, a Design Language for Interactive Systems

Alexandre R. J. François
Viterbi School of Engineering
University of Southern California

I. OVERVIEW

This poster introduces the principles that underly *Hermes/dl*, a design language for interactive systems. This ongoing effort is part of *Hermes*, a broader cross-cutting integrative project initiated and lead by the author. *Hermes* aims to create a common infrastructure for all parties involved to communicate during the creation, design, realization and evolution of complex, dynamic, interactive software systems. *Hermes/dl* establishes a theoretically grounded design language in which to develop *Hermes*' vocabulary and idioms for digital interaction. As a design language, *Hermes/dl* comprises of a collection of primitives, a set of organizing principles, and collections of qualifying situations. A graphical notation shields users from traditional algebraic notation, making *Hermes/dl* more accessible and appealing to potential users in creative and scientific fields. A rigorous graph theoretic formulation will enable mathematical manipulations, and underlie productivity and collaboration tools that assist users in the design and creation process.

II. MOTIVATION

Interactive systems are *human-centered* and should be *designed* as such; they are *dynamic* and should be *modeled* as such; they are *integrated* and should be *engineered* as such. While the ultimate engineering objective is a software system running on computing hardware, the design and creation activities center around communication between humans, with diverse levels of technical literacy (and interest). *Hermes* reconsiders and challenges past assumptions to create a computing environment that answers the human-oriented interaction challenge in an era of concurrent architectures and cross-disciplinary integration. *Hermes/dl* provides the theoretical means to describe complex dynamic systems in terms of components and their structural and temporal relationships. The language is designed for humans, and offers scalability of detail and composition.

III. OBJECTIVES

Hermes builds on the author's past and ongoing research in software architecture for interactive systems [1], and extensive personal experience in the collaborative design and implementation of cross-disciplinary interactive systems that integrate technologies ranging from audio and music processing [2], to vision and graphics [3], to networking [1], [3].

The Software Architecture for Immersipresence (SAI) framework created by the author [1] shares some of the design objectives identified for *Hermes/dl*, and will provide a valuable starting point in the iterative development of the design language. In particular, the SAI architectural style allows a continuum of intermediate-level representations from conceptual to logical to physical specifications (through the use of an appropriate architectural middleware). Furthermore, SAI promotes the encoding of system logic in the structural organization of simple computing components rather than in the complexity of the computations carried by individual components. SAI designs exhibit a rich variety of structural and functional *architectural patterns*, suitable for systematic study and reuse.

The migration from a software architecture framework to a design language framework broadens the scope of application for the principles that gave rise to SAI, and emphasizes the human communication component of the software system design activity. *Hermes/dl* elements and constructs will exist in three interchangeable formulations: a human-oriented graphical formulation, a proof-oriented graph-theoretic mathematical formulation, and a machine-oriented code middleware.

IV. BROADER IMPACT

The impact of *Hermes/dl* extends beyond research and development. *Hermes/dl* and derived tools will be instrumental in the realization of, and directly applied to, a new course under development by the author, titled "Collaborative Development of Interactive Software Systems." The course will feature a class project in which all the students will collaborate in the design and implementation of a complex, cross-disciplinary software system. The author pioneered this innovative project concept successfully in a previous graduate level course, and is introducing a similar concept in an undergraduate course titled "Principles of Software Development" which he is currently teaching at the University of southern California.

REFERENCES

- [1] A. R. François, "A hybrid architectural style for distributed parallel processing of generic data streams," in *Int. Conf. Software Engineering*, Edinburgh, Scotland, UK, 2004, pp. 367–376.
- [2] A. R. François and E. Chew, "An architectural framework for interactive music systems," in *Int. Conf. New Interfaces for Musical Expression*, Paris, France, 2006, pp. 150–155.
- [3] A. R. François, "Software Architecture for Computer Vision," in *Emerging Topics in Computer Vision*, G. Medioni and S. Kang, Eds. Prentice Hall, 2004, pp. 585–654.