From Contexts to Negotiation Forums

Carl Hewitt
M.I.T.

From 1967-1972, various PLANNER-like programming languages (like POPLER, QA-4, Conniver, etc.) explored a notion called “contexts” which is remarkably similar to the contexts more recently explored by Guha, McCarthy, Buvac, Lenat, etc. In these languages, a program was always “in” a default context and could add assertions and goals to other contexts which might in turn invoke more pattern-directed procedures. By 1972 it became clear that this notion of context was not going to work well in the networked multi-participant systems that we believed to be important in the future. So we abandoned work on this notion of context and began work on “Actors” as the universal primitives of concurrent systems. The work on Actors has led us to the notion of “negotiation forums” to provide contexts for discourse in multi-participant concurrent systems. This paper discusses the paradigm shift from contexts to negotiation forums.

We are developing negotiation forums to enable multiple participants to “plug-in” to software services. Like a hardware bus, the negotiation forum provides a protocol for software services (like hardware boards) to cooperate using a common resource. Unlike early hardware buses (such as the ISA bus of the PC world), the negotiation forum should allow other software services to be added easily without knowledge (e.g., dip switch settings) of the other participants on the forum. Like more recent busses such as the PCMCIA bus, it should automatically perform any reconfiguration needed to enable participants to be plugged in or removed while the system is in operation.

A hardware bus enables cooperation by requiring each board to make requests and receive permission before taking any bus action. A negotiation forum can also require participant systems to request permission by making proposals and requesting the approval of other systems before taking action under the authority of the forum. However, hardware bus protocols manage a simple resource (access to the bus to transfer a short data or signal burst in the immediate future) and thus can get by with protocols composed of a few request, priority, grant, acknowledge, etc. signals. A negotiation forum can be used to manage resources and resource use patterns with much greater complexity by providing participants with semantic methods for managing resources, their relationships, constraints, behavior, and interactions.

Software services plugged into a negotiation forum modularize and encapsulate the policies and preferences from multiple organizations. Negotiation semantics can make use of the following semantic methods:

* Constraint propagation to propagate the consequences of actions and to detect conflicts. * Case-based reasoning to generate proposals and support claims on the basis of precedents. * Explanation-based learning and
statistical learning to capture and generalize experience.

A negotiation forum can increase robustness by referring problematic cases to higher authority. If the negotiation peters out or times out without a clear agreement, then the negotiation so far may be summarized and passed on to higher authority (e.g., a human, or another Forum) for more in-depth processing. Thus, some unresolved conflicts are detected and brought to the attention of higher authority.

Negotiation forums can be monitored. Higher authorities can also monitor performance of the forum, making adjustments to system behavior to change how messages are processed. Initially this function can be performed by humans, but automated systems which monitor resources and adjust their behavior accordingly can also be developed.