A Visual Language for Specifying and Programming
Reality-Based Interfaces

Orit Shaer
Computer Science Department
Tufts University
oshaer@cs.tufts.edu
http://www.cs.tufts.edu/~oshaer

Research Goal: Simplify the development of reality-based interfaces by providing a technology independent high-level description language for specifying and programming these interfaces.

Research Structure: Iterative cycle with four stages: identifying development challenges, defining a high-level description language, evaluating the language, and images of three example reality-based interfaces that were developed throughout this research.

A high-level description language for TUIs

To lay the foundation for a high-level description language for RBIs we identified a set of core constructs sufficient for describing the structure of tangible user interfaces (TUIs) [3], a key interaction style included in the reality-based interaction category. We describe a TUI as a set of relationships between two types of physical objects – tokens and constraints. Tokens represent digital information or computational function. Constraints limit the behavior of tokens by suggesting how to manipulate a token, physically constraining it or providing a frame of reference for interpreting combinations of tokens and constraints.

TAC is the relationship between a token and a set of constraints. Similar to widgets in traditional graphic UIs, TAC objects encapsulate the set of manipulation actions that can be performed upon a physical object in a TUI.

Describing a TUI using TUIML

TUIML is a visual high-level description language for TUIs. It provides the following diagrams for describing the structure and behavior of TUI:

- **The TAC Palette**
  - Describes the structure of a TUI as a set of TAC relations. It uses a special visual notation, the TAC notation, to capture the physical relations between tokens and constraints. For each TAC it describes its token, constraints, what information it represents and how it can be manipulated.

- **The Dialogue Diagram**
  - Describes the behavior of a TUI in terms of high-level states and transitions. It draws upon the Statechart[1] notation and depicts the various states a TUI may be in, the tasks a user can complete within each state as well as the transition between these states. A high-level state represents a context in which a certain set of tasks can be completed. While the system is in a certain high-level state, users can perform these tasks sequentially or in parallel depending on the pre-conditions of each task.

- **The Task Diagram**
  - Depicts the decomposition of a task into a set of interactions and specifies the temporal relations between these interactions. For each interaction (discrete or continuous) the diagram specifies a pre-condition that describes what interaction objects and how many instances of each it requires. Furthermore, for each interaction it describes the physical and digital states of the system before and after it takes place.

Summary

We develop a visual formalism, TUIML, for specifying and programming the structure and behavior of TUIs. The TUIML notation explicitly captures the unique characteristics of TUIs such as parallel interaction and output channels. We expect that applying TUIML to the development of TUIs will simplify their development.

References