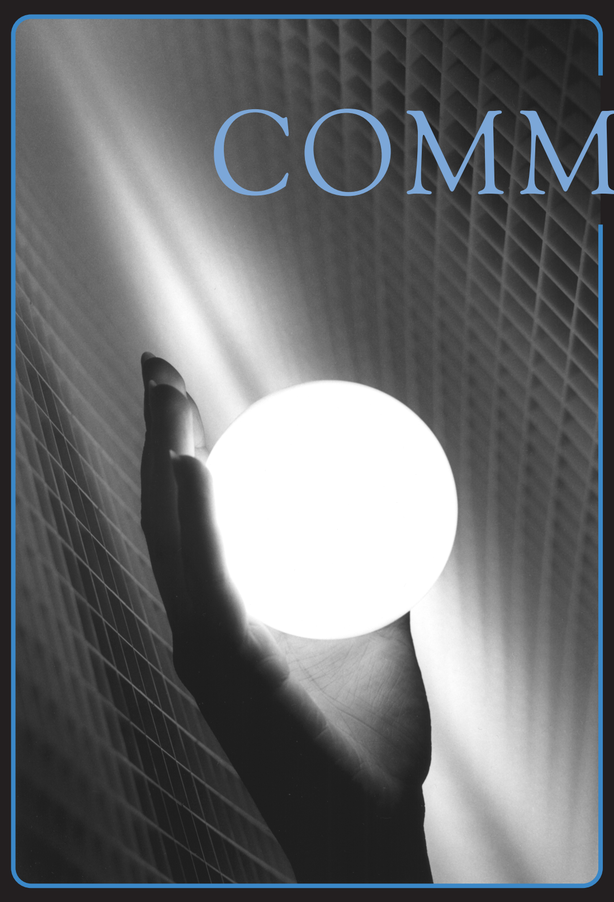
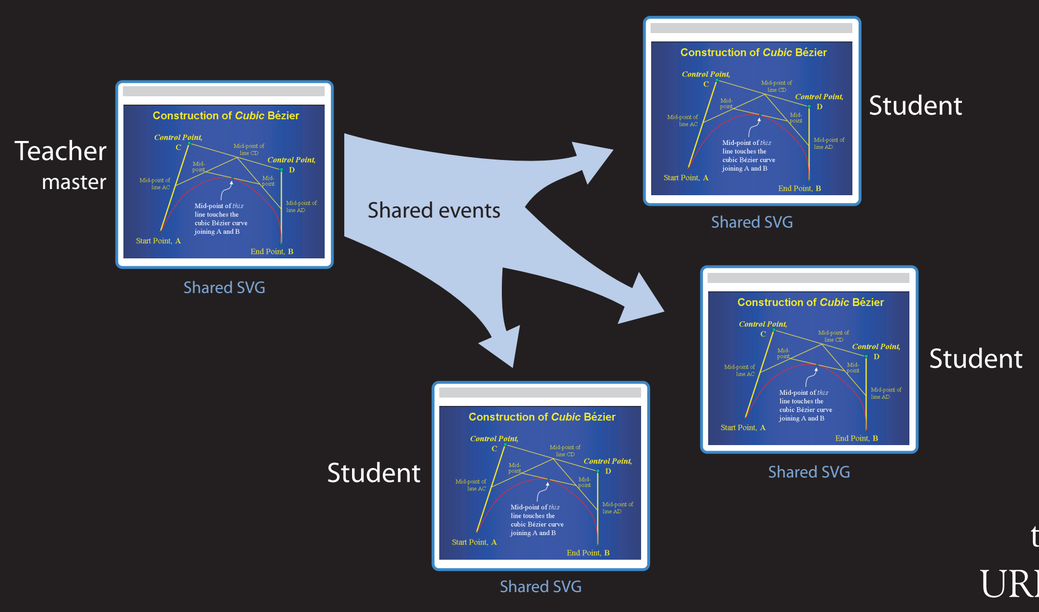


COMMUNITY GRIDS LAB

www.pervasive.iu.edu
www.svgarena.org

pervasive technology labs
AT INDIANA UNIVERSITY

Applications of a Collaborative SVG Browser



Teacher-Student

A simple but important application for a shared-event, collaborative browser is to display graphical materials in a synchronous distance-learning setting – typically, in a lecture given over the Web. A teacher runs the collaborative browser in master role, while the students are participating clients. The teacher can load graphics from a suitable URL (which may be a local file URL, if lecture materials have been preinstalled by students). The "document-load" event is propagated to the students' browsers, and they automatically load the same graphic. The teacher may also zoom and pan on points of interest. These events are likewise propagated to students, whose browsers follow the teacher's. Because only events are propagated (not whole screen captures), this is a potentially very effective use of network bandwidth.

In this application, the SVG browser may be a standalone application available to all participants, or it may be a plugin to a more general collaborative environment, providing more comprehensive support for distance learning (with audio, chat, whiteboard services, etc).

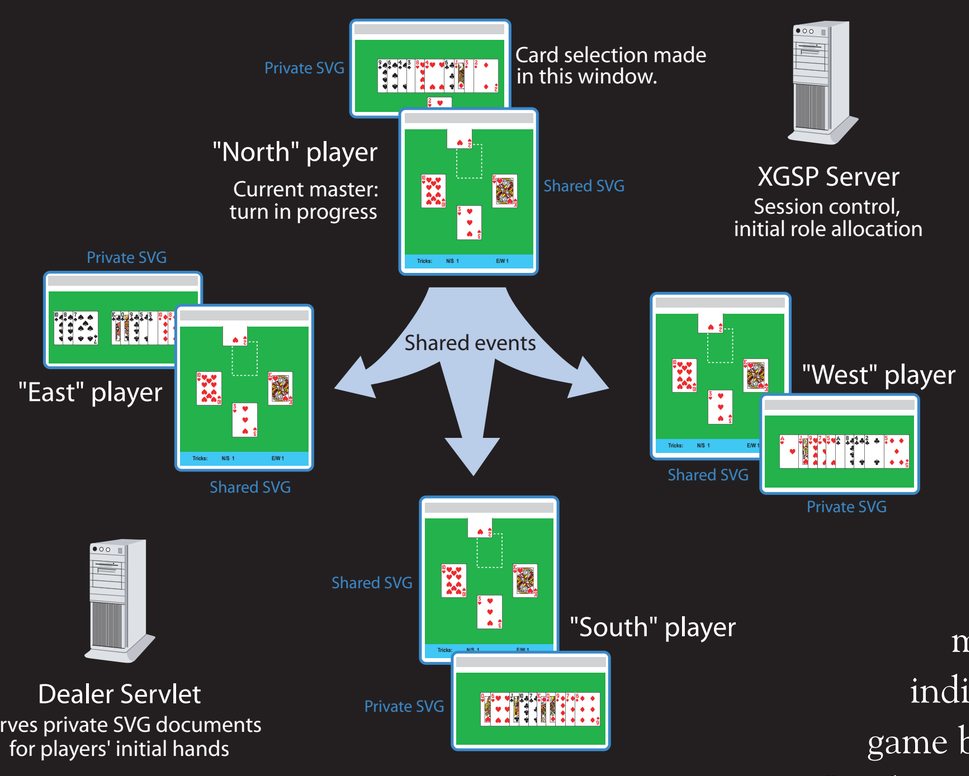
Simple Game Playing Patterns

More advanced applications can exploit on the remote propagation of DOM-level events. These include the familiar "onclick", "onmouse-move", etc events defined by JavaScript programmers when creating interactive Web pages and documents.

By propagating these events from master to participating client, richer patterns of interactivity and response are possible. If some participants are also able to acquire or hand on the master role dynamically, applications like game-playing become straightforward.

As a demonstration, we implemented an Internet chess-playing program as an SVG document with JavaScript event handlers. In this application the game is coded almost exactly as it would be for a "local" interactive SVG document. Events are automatically forwarded from current master to other participants by the collaborative browser. Identical event handlers on all participants validate the move and update the display in lockstep. The only modification necessary for Internet game-playing is logic to exchange of roles after a move completes successfully.

In this example there are essentially three roles: white player, black player, and observer (audience members in a tournament, say). The first two participants exchange the "master token" between moves.



More General Patterns

Games like chess have a simple "round-robin" interchange of roles. This can be generalized. In the future more advanced session control protocols (XGSP) will be added to allow finer control over roles.

Another simplifying feature of games like chess is that all information is public, so a single shared SVG document can capture the game state. In many games some information is private to individual players. Here we envisage a card game based on our shared browser infrastructure. It relies on the feature that a single instance of the collaborative browser can display several windows: some of these windows can be collaborative (shared) while others may be "local" (private). We assume that a separate dealer process generates the initial hands for the players (as SVG documents, of course!) and these are downloaded into the *private* windows of the respective browsers. This application also requires that JavaScript event handlers can program-matically generate shared events. This allows information (the selected cards) to be passed from a local private window to the shared window, and thus seen by all players.

Gaming is a useful model of collaborative behavior, and many patterns observed in games carry over to general collaborative applications.