

Exploratory Authoring of Interactive Content in a Live Environment



Bachelor Project Proposal, WS 2013/2014 – SS 2014
Software Architecture Group, Prof. Dr. Robert Hirschfeld

Lively Kernel, a Live Programming System in the Web

The Lively Kernel (<http://lively-kernel.org>) is a Web-based runtime and development environment for Web applications. It incorporates tools and techniques to be completely self-sufficient. Development in Lively is notably different from development in other environments. Instead of working on text files containing source code, programmers interact with and modify live objects. Lively offers various programming tools similar to other programming systems such as Squeak/Smalltalk, but also provides tools for editing texts, graphics, and even presentations. It allows for Wiki-like editing of Web pages, which might also include or be Web applications. Typical programming sessions in Lively often yield active essays, resembling the creative development process with embedded idea descriptions, initial design sketches, running prototypes, and applicable changes to the code base. Once finished, users can share these artifacts by publishing their page, individual live objects, or changes to the code base.

The screenshot displays the Lively Kernel web interface. At the top, a browser window shows the URL `www.lively-kernel.org/repository/webwerkstatt/webwerkstatt.xhtml`. The main content area is divided into several sections:

- Welcome !**: A paragraph describing Webwerkstatt as an interactive and programmable wiki environment for experimenting with different approaches to End-user Web Development.
- Documentation**: A list of links including [Lively2 Overview](#), [FAQ](#), and [PartsBin](#).
- Video Tutorials**: A list of video links such as [110419_ManipulateMorphs.mov](#), [110421_CreatingVideoTutorials.mov](#), and [110429_TextAttributes.mov](#).
- PartsBinBrowser**: A grid of various visualization examples, including `d3ForceLayoutEx...`, `BubbleVisExample`, `ConnectionsInE...`, `MorphDrawerSun...`, `d3BubbleExample...`, `d3ForceLayoutEx...`, `URLCirclePacking`, `ScatterplotD3Ex...`, `ProtovisDrawing`, `ProtovisStackOr...`, `TreeMap`, `MorphHierarchyV...`, `URLSunbursts`, `URLTree`, and `URLTreeMap`.
- Visualization**: A scatter plot showing `Sepal Length (cm)` on the y-axis (ranging from 4 to 8) and `Sepal Width (cm)` on the x-axis (ranging from 3.6 to 4.4). The data points are colored by species: `versicolor` (orange) and `virginica` (green).
- ObjectEditor**: A window for editing the visualization code. The code includes a `draw()` function that uses `d3.js` to create a scatter plot with a margin and a range for the x-axis.

Project Objectives

Open and flexible environments such as Lively are particularly interesting for authoring interactive tools that combine data from various sources into dynamic representations. For example, Lively's strong focus on direct manipulation of live objects allows users to start with an initial sketch, but also to successively refine that sketch into a polished Web application. Lively already incorporates basic data visualizations and a set of scripting tools, but the creative process of building interactive and visual applications needs dedicated tools for authoring such applications. Project participants will design and implement tools and frameworks for exploratory programming, which might include support for:

- scripting interactive data representations
- direct manipulation of visual components
- program comprehension through concrete data
- program creation by gradually abstracting from concrete examples
- withdrawing changes that turn out to be inappropriate

Organization

A group of six to eight (6-8) students may participate in the project. Organization and tasks will be determined by the project participants, following an Agile development process. The project will be carried out at the Hasso Plattner Institute in Potsdam. Project participants are expected to communicate with our partner on a regular basis. In WS 2013/2014, participants will work on initial design sketches and prototypes. Main steps in design and implementation are to be executed in SS 2014.

Partner & Contact

Dan Ingalls, Robert Krahn
SAP Labs, Palo Alto, CA, USA
<http://sap.com/research>, robert.krahn@sap.com

Prof. Dr. Robert Hirschfeld, Bastian Steinert, Lauritz Thamsen, Jens Lincke
Software Architecture Group, Hasso Plattner Institute, Potsdam
<http://www.hpi.uni-potsdam.de/swa>, hirschfeld@hpi.uni-potsdam.de