Although some frameworks support the design and development of uncommon input devices and post-WIMP interaction concepts, this keeps a highly challenging task and demands expertise on all layers ranging from hardware prototyping over signal processing to application programming. Furthermore, the relevant tools and techniques are spread over diverse heterogeneous frameworks and toolkits. We therefore introduce the interaction library Squidy which eases the design of natural user interfaces by unifying relevant frameworks and toolkits in a common library and providing a central design environment based on high-level visual dataflow programming combined with zoomable user interface concepts. Squidy thereby hides the complexity of the technical implementation from the user by providing a simple visual language and a collection of ready-to-use devices, filters and interaction techniques. However, if more functionality is needed by the user, Squidy provides the capability to adjust parameters of a certain node interactively. By switching to a different IDE or rebuilding the corresponding node without the need of access to the source code of the functionality. Modifications are applied as soon as the user zooms out and knowledge about the functionality of the corresponding node without the need of switching to a different IDE or rebuilding the node or pipe and provides direct feedback about failures and their corresponding impact.

Key Benefits:
- Adjusts complexity: semantic zooming enables access to advanced functionality on demand (e.g. parameter configuration, documentation, visual debugging, embedded source code)
- Unifies diverse toolkits, drivers and frameworks (e.g. for Surface Computing, gestural interfaces, pen input, touch interaction, gaze-based input)
- Supports whole life cycle: interactive design, implementation, evaluation & redesign

Squidy data types hierarchy based on primitive virtual devices introduced by [Wallace, V. L., 1999]. The interaction data consists of single instances or compositions of these predefined data types.

In order to improve user understanding and interpretation of the interaction data, Squidy provides such a data flow visualization which maps the values of the carried data according to its occurrence over time. This approach offers the opportunity to identify unique interaction patterns and provides the possibility for visual debugging at run-time.