Domain-specific languages for pervasive systems

Görel Hedin, Boris Magnusson
Department of Computer Science
Lund University
Sweden
{gorel | boris}@cs.lth.se

Position paper
Sept 16, 2004

We are currently taking part in the EU project on Palpable Computing that aims at providing a new perspective on ambient computing [palcom2004]. One key issue in this project is that it must be possible for users to construct new systems from existing components, and for deconstructing existing systems into their constituents. In this position paper we present some ideas we have about using domain-specific languages for programming such palpable systems.

We envision a future where pervasive systems can be constructed dynamically, on an ad-hoc basis, by combining the functionality of several individual communicating devices. These devices might be stationary equipment like TV sets, audio systems, refridgerators, etc., as well as smaller mobile devices such as cellular phones, laptops, cameras, mp3 players, etc. The combination of a set of devices in a home or in an office may allow new applications to be developed on an ad-hoc basis, simply by configuring the devices in a suitable way. We might call such new applications *logical devices*.

An example of a logical device could be a device that records a phone call as an mp3 file and stores it in the mp3 player under a folder "recent calls". As software engineers, we can easily imagine several ways of programming such an application. For example, we could write the application as two communicating pieces of software that are installed on the mobile phone and the mp3 player, respectively. This solution would require quite an amount of programming expertise, and it is not something that could be done by a normal end user. How could this problem be solved so that the end user that suddenly got this idea, could somehow easily construct this logical device?

We are currently working on an infrastructure framework where user interfaces can migrate between nearby (mobile) devices over radio communication [SvMa2004]. Such a framework can be used as the basis for allowing the user to control one device from another, e.g., controlling the mp3 player from the mobile phone. The framework also allows the user to connect devices and set up communication channels for streamed data, e.g., audio. For example, if the mp3 player has a recording function, the user could redirect the phone audio output to the mp3 recording function, and in this way

record the phone call. The framework thus allows the user to emulate a logical device by performing all the basic configuration and controlling steps manually. However, we would like to explore techniques for allowing the user to easily program the complete logical device so that it becomes easy to use. The programming must also be very easy, so that end users can perform it.

To accomplish such easy programming of logical devices, we envision the use of a domain-specific language, probably partly visual. We are looking for different ideas in this area. The language will be an area of exploration, and one idea is to organize the language in layers, each giving more programming power. A normal end user would use the top layers whereas an experienced software engineer could make use of additional layers that include a complete general programming language to program sophisticated logical devices.

As a technology for implementing domain-specific languages we are using a generative transformation technique called ReRAGs [EkHe2004a]. This technique is based on object-oriented attributed grammars, in-place declarative transformations, and aspect-oriented language implementation modules. The technique allows domain-specific languages to be built in a modular way on top of general-purpose languages, supporting a flexible way of experimenting with domain-specific languages [EkHe2004b].

References

[palcom2004] Palpable Computing - a new perspective on Ambient Computing, www.palcom.dk

[SvMa2004] D. Svensson, B. Magnusson: An Architecture for Migrating User Interfaces, Nordic Workshop on Programming Environment Research, NWPER 2004, Turku, Finland

[EkHe2004a] T. Ekman, G. Hedin: Rewritable Reference Attributed Grammars. Proceedings of ECOOP 2004, LNCS 3086, Springer.

[EkHe2004b] T. Ekman and G. Hedin. Reusable Language Specification Modules in JastAdd II. Workshop on Evolution and Reuse of Language Specifications for DSLs, ERLS 2004.