Virtual Product Design Case Study: The Nokia RFID Tag Reader

Pertti Repo, Mikko Kerttula, Marko Salmela, and Heikki Huomo

EDITORS' INTRODUCTION

Pervasive systems are used in the milieu of other conceptually demanding social and task-oriented interactions—and they will fail if they demand the system user's full attention. In January 2003, we discussed a usability methodology applied to a highly successful restaurant order system (V. Stanford, "Pervasive Computing Puts Food on the Table"). In April 2004, we published an initial usability evaluation framework that systematized the concepts discussed in the earlier work (J. Scholtz and S. Consolvo, "Toward a Framework for Evaluating Ubiquitous Computing Applications"). This issue further extends techniques for producing usable pervasive applications: *virtual product design* allows users to test simulated versions of pervasive devices before the hardware becomes available. VPD adds a new chapter to the usability and design story we've been covering. I hope you enjoy its contribution to our ongoing discussion of how to create usable pervasive systems. I think it will be important. —*Vince Stanford*

The rise of the pervasive computing industry as a whole has been slow. We see at least two significant reasons. First, the technological challenges posed by pervasive computing systems are not simple. Furthermore, solutions must emerge from a constant battlefield of technologies, representing competing hardware and software platforms, wireless technologies, programming languages, and so on. Second, pervasive system developers lack methodologies for estimating a planned product's success among end users. Do users need this kind of product or service? If they do, are they willing to pay for it?

Forecasting success is difficult even with nonpervasive products. It becomes even more difficult with the new multitechnology systems that pervasive computing introduces. Companies can't simply rely on market forecasts. They must go to customers and demonstrate new product features with concrete examples. And they must do so early in the development process.

Virtual product design is a methodology that supports heterogeneous, hardware-software prototyping.[1] Using virtual prototypes, VPD can work with realistic product concepts throughout the product life cycle. Virtual prototypes are especially effective in facilitating communication among stakeholders along the value chain and in validating product concepts with early end-user testing. These features make VPD a promising development tool for companies trying to cope with the inherent complexities of pervasive systems.

In 2003, Nokia adopted this methodology to develop an RFID kit for the Nokia 3220 mobile phone. Cybelius Maestro, a virtual prototyping tool developed by Cybelius Software in cooperation with Nokia Mobile Phones and the VTT Technical Research Centre, Finland, supported VPD development of the product concept and its features. In March 2004, Nokia launched the first RFID-enabled mobile phone at CeBIT 2004.

VIRTUAL PROTOTYPING REQUIREMENTS

Pervasive computing's multidisciplinary and heterogeneous nature makes it a demanding area for prototyping. Indeed, it's a challenge just to get different system parts working together while using tools from specific design disciplines. However, the greatest challenge is not necessarily in the technology integration but in earlier phases when designers are conceptualizing and validating system features and specifications. As the target systems become more complex, it becomes more difficult to justify features through traditional methods, such as textual descriptions, technical drawings, use cases, and operational concepts.

By combining software and hardware platforms, virtual prototyping offers a better way to illuminate system characteristics in a product. Virtual prototypes are realistic, supporting user testing of new product concepts with functionality and appearance comparable to that tar-

APPLICATIONS

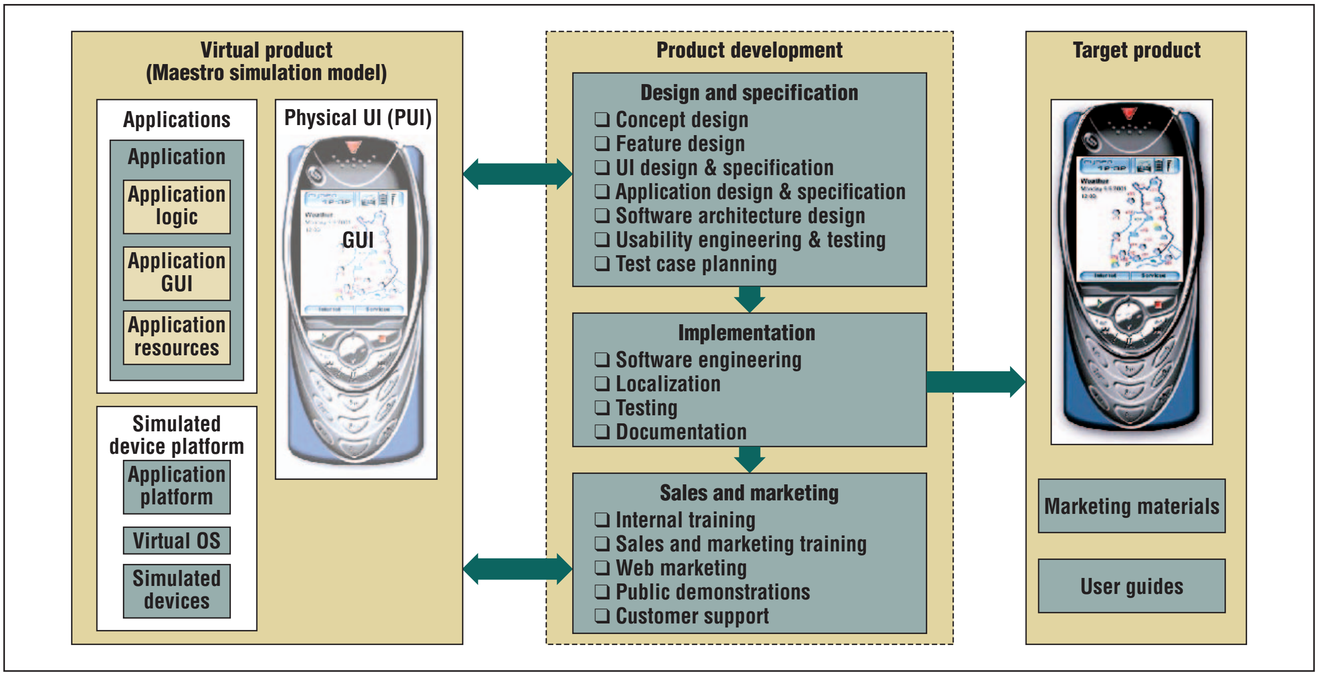

Figure 1. Virtual product design. A virtual product simulates prototype components. In an iterative process, it evolves throughout the design/specification and implementation phases as well as the development of sales and marketing programs for customer support.

geted for the final product implementation. They can integrate different design ideas behind the target object, and they aim for not only cross-discipline integration but also the integration of parts and components from different maturity levels. This heterogeneous approach enhances the prototype's accuracy and also saves time and effort needed to realize it. Designers can realize product ideas faster by using not only simulated parts but also existing products, parts, and components. For example, you can test and demonstrate a new wrist computer and its services by prototyping the user interface software using existing software algorithms and hardware. On the service side, you can prototype new service concepts using existing network connections and services.

When prototyping is part of industrial product development, the tool requirements focus on issues aimed at reducing development time and effort. Virtual prototyping tools must go further. They must support effective component reuse as well as fluent transition from prototyping and specification phases to the implementation.

VIRTUAL PRODUCT DESIGN

We have applied VPD to electronic devices, where it provides both a process and a technique for discovering a product concept, features, and behavior. As a process, VPD covers whole product development from initial concept design through virtual prototypes to marketing and end-user support. As a technique, VPD is a product development and business process framework that applies virtual prototyping and virtual prototypes in selected process phases and tasks.[1,2] Figure 1 depicts the virtual product concept and its use in product development.

Electronic device design involves a lot of communication between people representing different fields of expertise, such as industrial and mechanical design, software development, user interface design, usability, and marketing. Each field has its own modeling techniques and jargon, which makes it difficult to find a common understanding of the product concept. The virtual prototype provides a common model for specifying, validating, and testing the product concept, thus facilitating communication among all stakeholders.

CYBELIUS MAESTRO: VPD TOOL SUPPORT

The Cybelius Maestro tool platform meets the industrial requirements of VPD rapid prototyping. Developed in conjunction with Nokia Mobile Phones and VTT, the tool has its origins in virtual reality prototyping research that VTT and the University of Oulu, Finland, carried out in the 1990s. It targets development for digital convergence appliances such as mobile phones, PDAs, game consoles, and set-top boxes. It is also well suited for developing devices and services for the multidevice wireless environments that pervasive computing involves.

The decision to develop Maestro was motivated by the diverse set of closed tools that companies were using for specific tasks. These tools lacked a fluent means to work cooperatively, which made it difficult to integrate domains such as industrial design and mechanical, software, hardware, and usability engineering. Building heterogeneous system simulations was also difficult. And finally, no tool support existed for new technology areas, such as perva-

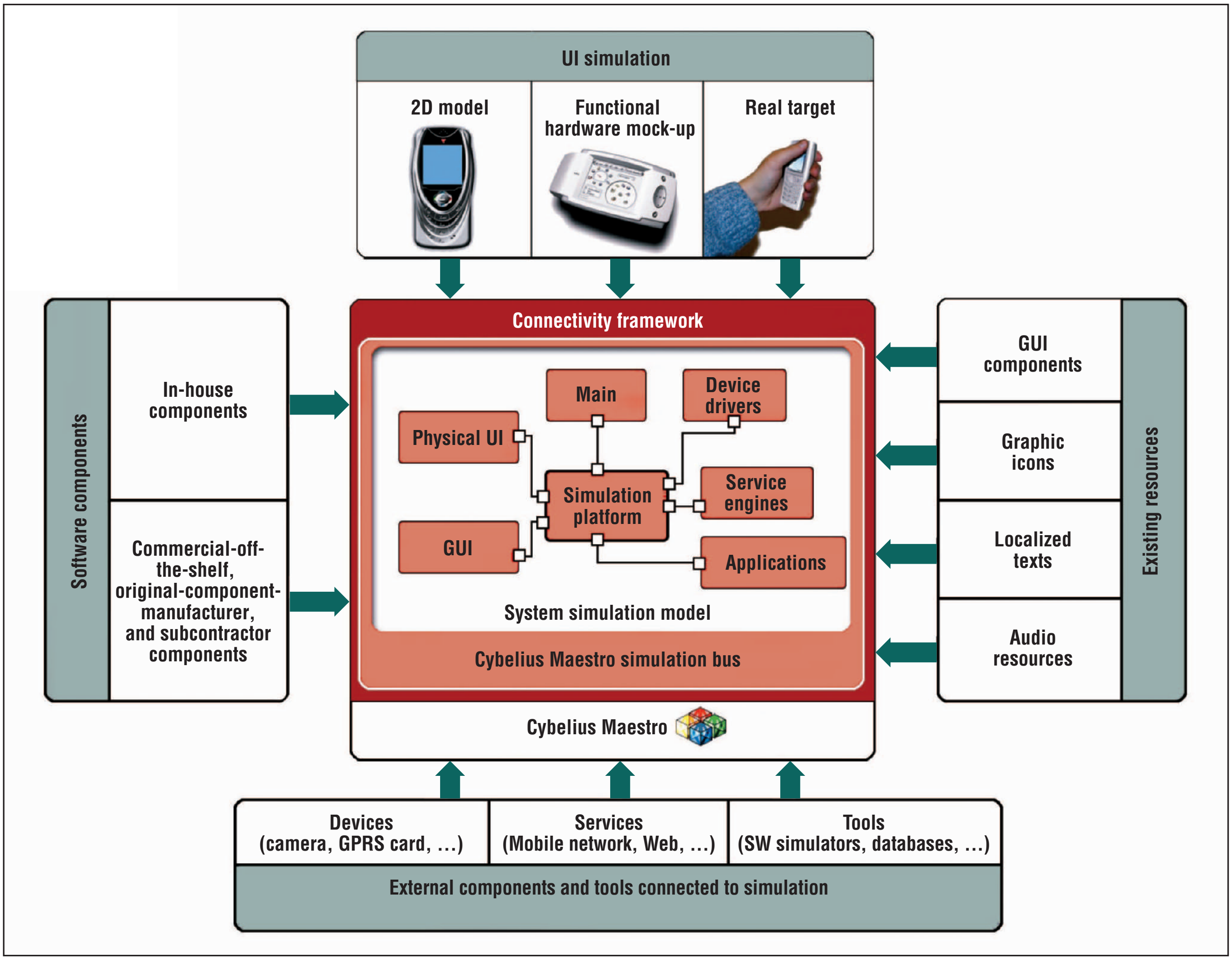

Figure 2. The Cybelius Maestro development platform.

sive computing, or for new tasks such as usability engineering and multimodal user interface development.

Cybelius Maestro promotes concurrent engineering and component reuse. It is an open, integrated development platform (see figure 2), with a tool plug-in architecture that supports extensions for new applications and design tasks. An open simulation bus supports various simulation components: software and hardware components as well as external simulation systems and tools. Cybelius Maestro is based on open standards, such as Java and XML. Other features include extensive user interface simulation and visualization properties and code-generation support.

CASE STUDY: NOKIA RFID TAG READER

The Nokia Ventures Organization generates new business ideas that fall outside the scope of Nokia's core business units. The organization recently proposed using mobile phones as an RFID tag reader. This functionality could give consumers new interaction modalities and service discovery methods by capitalizing on the promise of RFID tag technology.[3] The broad market penetration of mobile phones in the Nordic countries has achieved the first truly pervasive computing platform, so it's only natural to add pervasive consumer services and capabilities to it. In addition, mobile phone capabilities have developed rapidly, making them good candidates for the universal remote controllers and sensors that RFID enables.

Nokia began the tag reader design early in 2003 with a search for the product concept. The first ideas focused on a separate device for reading data from and writing data to the RFID tags. The concept further evolved to integrate the reader into the mobile phone. Other concepts included the kinds of services that the device could offer consumers.

Nokia used the Cybelius Maestro tool to build the virtual prototypes. It was already using Maestro for building simulations on Nokia's Series 60 mobile phone platform. For the RFID tag reader product, the Cybelius Maestro tool allowed Nokia to study usable product concepts given the RFID tech-

APPLICATIONS

Figure 3. The RFID read-write-send virtual prototype as seen on a PC screen: on the left, a functional simulation of a mobile phone; on the right, the user interface of the RFID RWS prototype.

niques and Series 60 application model. The basic idea of reading from and writing to an RFID tag is not detailed enough for conceptualizing a new product until it is put into the context of an actual implementation environment. Simulations with virtual prototypes offer this detail.

Conceptualizing the device and services

When work began on the RFID prototype, Nokia phones lacked hardware extension mechanisms. The prototype therefore reflects the initial idea of a tag reader and writer as a separate device—a key chain, in this case. Users would touch an RFID tag with the read-write-send (RWS) device, which would read the tag data and send it to the phone over a Bluetooth connection. In a write operation, the phone would send data to the RWS device, the user would then touch an appropriate RFID tag, and the RWS device would write the data to the tag.

The first problem to solve was enabling the user to enter the content that would be written to a tag in the Series 60 mobile phone platform. We opted for the Notepad application as an editor for programming and writing custom data to be sent to the RFID tags. The RFID RWS prototype thus combined the specified software and hardware. We built the prototype phone simulation using Cybelius Maestro with the Series 60 Extension Package. We built additional functionality and the prototype user interface using the Java language and its Swing components.

Figure 3 depicts the prototype's user interface. On the left side is the functional simulation of a Nokia Series 60 mobile phone; on the right side is the user interface of the RFID RWS prototype. The phone simulation includes a GUI (the phone display) and a physical user interface (the phone keys). To use the phone's features, the user clicks the appropriate key. The prototype responds with an animation indicating that the phone key was pressed, and the appropriate GUI for the corresponding application appears on the phone display.

To use the RFID prototype's features, the user clicks the appropriate icons on the right side and corresponding results will appear on the phone's display—for example, notification of an incoming short-message service (SMS) message. The prototype included a real GSM card (Nokia D211), so it could send a real SMS message to a real phone and vice versa.

Figure 3 shows the simulated RWS device in the lower-right corner. It has only one button and two indicator lights: one shows the device's state (read/write/show/off), while the other blinks when the device is either reading data from or writing data to a tag.

Operation depends on the feature the user chooses for simulation. For example, if the choice is to read an RFID tag, the user must first select the RWS device's read mode and then use the PC mouse to select the icon representing RFID tag types (see figure 3). This operation sends notification of a received message to the phone's display, as shown in figure 4. The user can now read the message in the simulated phone in same way it would appear in the corresponding real phone.

The prototype tested many features. Users could read an RFID tag to get a weather forecast, open a URL to music services, buy a bus ticket, send a postcard via an electronic postcard service, get location information for selected services (such as restaurants), download a Java game, and check the available payment method for a service (the wallet application). The virtual product simulation lets you show how to combine an existing technology (mobile phone) with external techniques (RFID tags and the read/write device) to offer useful consumer services. In addition, you can connect the virtual simulation to real-world services through a GSM card on the hosting PC.

Extended VPD benefits

The prototype's realistic-looking functionality enhances understanding of a

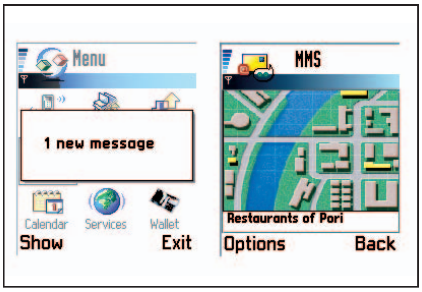

Figure 4. Simulated phone displays. On the left, the PC phone view shows the notification of a received message on the phone display after the user selected the “restaurants” RFID tag. On the right, the user sees the actual multimedia message received—a map highlighting restaurants in Pori town.

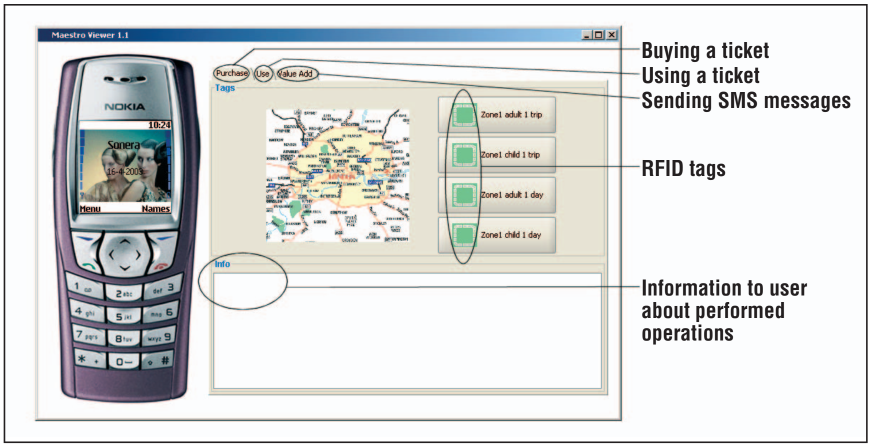

Figure 5. Ticketing service virtual prototype with the RFID tag reader embedded in the mobile phone: on the left, the mobile phone simulation; on the right, the ticketing service’s user interface panel.

product concept, its possibilities, and its limitations. Nokia first used the prototype to support the decision-making process and to sell the product concept within the organization. When Nokia decided to go ahead with the concept and build a hardware prototype, the company gave the job a subcontractor and used the RFID RWS prototype for communicating the product concept. The simulation served as the specification; no formal descriptions were required. The targeted hardware prototype design differed somewhat from the design depicted in Figure 3, but it was also a key chain and its functionality was the same.

During the RFID tag reader’s conceptualization phase, designers also developed other virtual prototypes for further study and concept refinement. These prototypes led Nokia to embed the RFID tag reader into a phone. Figure 5 shows a ticketing service prototype built to address not only the idea of embedding the RWS device in the phone but also the security concerns inherent in using RFID tags.[3] The prototype application paid a subway trip fare with a mobile phone (Nokia Series 40 this time) equipped with an RFID tag reader. Again, the prototype embedded the GSM card, so you could actually purchase a Helsinki subway ticket by sending a text message to the appropriate phone number.

With fewer features than the RFID RWS prototype, the ticketing service prototype made it easier to study how to design a usable interface that could accommodate the security features required for this kind of service. The virtual prototype discloses usability problems early. Developers can start end-user testing immediately, getting feedback from real customers to determine whether they find the concept easy to use and valuable enough to pay for.

Our case study shows how virtual prototypes can mediate the product concept through the whole value chain. Applying VPD to the development of pervasive products can yield generally the same benefits as in any product development process, but it might prove crucial to addressing the multidimensional complexities of products for pervasive computing markets.

REFERENCES

1. M. Kerttula and T. Tokkonen, “Virtual Design of Multi-Engineering Electronics Systems,” *Computer*, Nov. 2001, pp. 71–79.
2. K. Leppänen, M. Kerttula, and T. Tuikka, *Virtual Design of Smart Products*, Edita, IT Press, 2003.
3. V. Stanford, “Pervasive Computing Goes the Last Hundred Feet with RFID Systems,” *IEEE Pervasive Computing*, vol. 2, no. 2, 2003, pp. 9–14.

Pertti Repo is R&D manager at Cybelius Software, part of the CCC Group. Contact him at pertti.repo@cybelius.com.

Mikko Kerttula is business development manager at Cybelius Software, part of the CCC Group. Contact him at mikko.kerttula@cybelius.com.

Marko Salmela is the chief software architect at Cybelius Software, part of the CCC Group. Contact him at marko.salmela@cybelius.com.

Heikki Huomo is the director at Nokia Corporate Strategy Organization. Contact him at heikki.huomo@nokia.com.