Virtual Design of Multiengineering Electronics Systems

A virtual design and prototyping environment offers a solution for companies marketing products that depend on a fast development cycle, user satisfaction, and localized customization.

Mikko Kerttula
Cybelius Software

Timo Tokkonen
Nokia Mobile Phones

Corporations that create high-technology applications or operate in the consumer electronics and telecommunications markets must be particularly adept at reacting to changing business circumstances. Tightening global competition, distributed organizations, advancing technology, shortened product life cycles, active study of user preferences, and increasing product complexity all require new methods for conducting product development and related business operations. We can classify these general business demands under three main trends that also affect the development of computer-based systems.

First, in many product areas, user satisfaction has become increasingly important to staying competitive. User satisfaction is important not only in consumer electronics products, but also in original equipment manufacturing and industrial electronics. Given the close competition in many technological areas, which laws and standards sometimes further impact, companies have increasingly shifted their focus to user interface design, usability, and an appealing appearance. For example, in a mobile terminal business, implementing a new product or features may no longer be the problem. Often, the problem now is how to form an optimal product variation for different customer segments in international markets.

Second, effective communication has become increasingly important. Modern computer-based systems result from a multidomain development process that incorporates several engineering disciplines and sciences. Successful organization and management of this work requires developers to communicate, cooperate, and coordinate effectively. Given that product development increasingly involves nonengineering teams that undertake usability-engineering and industrial-design activities that support early linking of product development to other company operations, the need for more general, nontechnical communication skills also increases.

Third, global and distributed business operations place new demands on organizations and product development. Geographically dispersed development teams must communicate effectively. Distance and time differences provide the first obstacles, but businesses also must overcome linguistic and cultural challenges. Multinational markets create special needs for internationalization from the product marketing and development viewpoints. In addition to taking the local language into account, specific designs and product features must accommodate cultural and geographical preferences. Businesses also must configure their products to fit with local regulations and laws. To address these concerns, companies must implement methods and processes to efficiently develop product variants.

Our work focuses on developing consumer electronics and telecommunications products that reflect the design decisions these three trends inspire. We base these efforts on solutions and practices derived from systems engineering and software engineering research, integrated with the results of several case studies conducted between 1996 and 2001 by VTT Electronics, a unit of the Technical Research Center of Finland. We conducted these case studies in several joint research and contract research projects, in which our partners were both international companies and small- to medium-

0018-9162/01/$17.00

sized enterprises operating as subcontracting partners. Research in these projects focused primarily on small-sized products such as cellular phones, other personal terminals, set-top boxes, and wrist computers.

SOFTWARE-INTENSIVE MULTITECHNOLOGY PRODUCTS

The role of software has increased dramatically in electronics products since the 1990s. In mobile phones, for example, the size of software has risen from a few kilobytes in analog phones to about 1 Mbyte in digital GSM phones.[1] Significantly, the weighting of this software has shifted from lower-level system software to user interface components: In some consumer products, the UI can account for more than 50 percent of all software.

To develop a successful, high-quality, desirable product, however, developers must integrate software development with expertise from other engineering and expert domains. This focus must extend beyond more traditional disciplines—such as mechanical and electrical engineering—to include the increasingly important areas of industrial design, usability engineering, and other disciplines.

Further, the entire product development process must be efficiently integrated with other company functions, such as marketing, manufacturing, and customer support. Developing modern, software-intensive electronics and telecommunications products in highly competitive markets equates with demanding systems engineering tasks such as developing a complex process-control or advanced-weapons system. Only the technologies used and their weightings differ. The "Focused Development" sidebar summarizes the characteristics of software-intensive multitechnology electronics products.

Product development requirements

Product development strives to increase customer satisfaction and achieve better quality. Other business- and company-specific objectives can be defined, such as shorter product development times, support for global operations, and so on. Our industrial-partner cases indicate that the requirements for implementing better product development in software-intensive multiengineering products fall into three categories, as described in the "Efficient Product Development" sidebar: product development process, tool support, and simulation and prototyping support.

These requirements resemble many basic concurrent engineering requirements.[2] Concurrent engineering is synonymously known as simultaneous engineering, integrated product and process development, or integrated product development. These approaches, supported by various standards,[3,4] have been used successfully in military systems, shipbuilding, automobile manufacturing, and the aviation and space industries. They address issues such as total product visibility, multiengineering cooperation, concurrent work, and effective communication. Likewise, they also set demands for the tools they use and recognize the value of prototyping technologies.[5]

However, using these approaches as base practices for the development of software-intensive multitechnology electronics products presents two challenges. First, the approaches usually focus on large systems in relatively stable business areas, where established standards and national or international regulations typically guide the work. We must determine how to implement them in areas of rapidly changing technology and market dynamics.

Second, concurrent engineering and systems engineering do not focus on software development—they

Focused Development

The software-intensive multitechnology consumer electronics products we develop exhibit the following characteristics.

Software development focus

Usually, software development takes a major part of the product development effort and integrates other technologies and sciences. These include the traditional technologies—hardware, mechanical, and telecommunications design—along with newer ones, such as industrial design and usability engineering.

We also incorporate application-specific technologies in our development process: For example, value-added service development for a mobile business application, or physiology for a wrist-computer business. As developing technologies mature, we will incorporate them as well, including ubiquitous computing, intelligent environments, and context-awareness applications.

Business area focus

To fully integrate technology and business requirements, our development process maintains a strong business focus. Business factors that demand attention include developing for a global and highly dynamic market, succeeding against strong competition, maintaining an innovative high-technology edge, establishing highly dynamic organizations and processes, and emphasizing a user-centered design process.

regard software as only one component of a system. We must decide how to make these approaches recognize the software's dominant role in the system's final features, as is the case with software-intensive products.

Process support

Because we focus on software-intensive products, we cannot disregard the role of software development processes. Software process models such as the classic waterfall model and its extensions, software prototyping and incremental and evolutionary development models,[6-8] serve well in their specific application or technology areas. These models inherit several objectives from concurrent engineering and systems engineering practices and thus meet many of our requirements. The challenge in applying these models to our needs lies mostly in the early integration of software engineering with other engineering areas and business operations. Recognizing the importance of user interface design and usability, we must also emphasize the role of user-centered design within the models and throughout the design process.

Efficient Product Development

We have found that products can be developed efficiently by meeting the following requirements.

Core development

Efficient development begins with these activities.

- *Early validation*. Validating product concepts, versions, and features early is vital. You can achieve better customer satisfaction by validating design alternatives efficiently at the beginning of the design cycle. Reducing the number of costly mistakes through efficient early validation can result in better product quality and shorter development times as well.
- *Efficient communication*. A process model should support effective communication at both the technical and human level. In practice, this means that product development goals and particulars must be communicated clearly between technical teams, other company operations, and partners and customers.
- *Multidisciplinary cooperation*. A process model should support multidisciplinary cooperation by providing common and integrated views of the design target. The model should also provide updates on the design's status.
- *Concurrent operations*. Integrating design disciplines during the development cycle's early phases can achieve support for concurrency in operations. The interfaces between different technology areas in a product must be defined as soon as possible. Concurrency should extend beyond the design itself. For example, market tests and protoseries manufacturing can be implemented in parallel with detailed product design.
- *Component and document reuse*. The development process should explicitly support component reuse. Pretested and preused components normally exhibit higher quality than ad hoc components. Further, component reuse decreases the amount of work and thus shortens development time.
- *Adaptivity and flexibility*. The development process model should be flexible enough to accept the adoption of new tools, methods, and technologies. Likewise, it should allow changes to the project's organization and business strategy.

Tool support

Implementing the following measures can increase the efficiency of development tools.

- Tools should use standardized interfaces, both when accessing data and interfacing with other tools. If no standardized or de facto standard interfaces exist, tools should provide flexible and open interface components that allow customization of the tool to different data formats and other tools.
- If possible, tools should be customizable to new tasks. They should also have some programmable capabilities that permit building new features into them. This capability lets existing tools adapt to new design tasks until specific tools can be developed for them.
- Tools should be built upon flexible base technologies. Doing so helps them adapt to different design methods and notations easily, without undergoing extensive updates.

Simulation and prototyping support

Building the following capabilities into simulations and prototypes can make this phase of the design process more efficient:

- More realistic simulation of software-intensive multiengineering products and new technology areas is a core design goal. To achieve realistic simulation, prototyping should combine different technologies and simulate various product features in an extensive product model presentation.
- Prototyping tools should support heterogeneous simulations so that different simulation tools and components with different maturity levels can combine to create an extensive product or system simulation.
- It should be possible to use prototypes and their components in real product components. No extra effort should be needed to build simulations for different purposes. Instead, simulations should evolve incrementally toward real product components.

Figure 1. Virtual reality prototyping. VRP can use virtual and physical models in combination to provide the most effective simulation of the prototyped product's features. (CyPhone model courtesy of JP Metsävainio Design Oy, Finland.)

As the ongoing standardization work with the ISO/IEC 15288 standard[9] for System Life Cycle Processes—scheduled for publication in October 2002—shows, the special nature of software-intensive multitechnology products has been widely recognized. This work strives to provide a new standard suited to modern systems, including digital computers and software, by combining project management, systems engineering, and software engineering approaches. The standard also emphasizes the role of human interfaces in component-based software.

VIRTUAL REALITY PROTOTYPING

We base our approach to meeting the requirements for developing software-intensive multitechnology electronics and telecommunications products on a virtual product design process and virtual reality prototyping technology. Virtual reality prototyping (VRP) can be viewed as a process or technology description.

Process and technology

A VRP process combines simulation models and virtual reality techniques to simulate a product or a product concept and its behavior. In a technological approach to VRP, the focus usually shifts to realizing the virtual reality prototype. The prototype simulates a real target object and strives to portray its physical and logical functionality and appearance adequately. It achieves this goal by combining different simulation models and virtual reality techniques.

We base our definition of VRP on the work of Edward J. Haug and colleagues[10] on virtual prototyping simulation (VPS) of mechanical systems. Our approach conforms with many of their ideas and findings, but it has a different focus. When concentrating on software-intensive consumer products, we emphasize the simulation of embedded behavior, such as software, and realistic appearance. VR techniques play an important but not exclusive role in the realization of a virtual reality prototype. In general, implementing a virtual reality prototype does not rely on computer models alone, but it can include, for example, partial hardware mockups. Using virtual and physical models together helps provide customers with useful information about the overall quality and various features of the prototyped product.

Thus, VRP integrates advanced modeling, multidiscipline simulation, interactive user interface, and VR techniques. To allow concentration on a particular aspect of a product, VRP combines VR and visualization techniques with various simulation models to provide different levels of reality. In fact, VRP introduces heterogeneous prototypes, but in the wider sense as it is understood in software engineering, where heterogeneous means different abstraction levels of software, not abstraction levels of complete product features such as mechanics, design, and electronics.

Simulated features

To support the special characteristics of software-intensive multitechnology electronics and telecommunications products—especially, cross-engineering and heterogeneous prototyping of these areas—we found the following simulated product features to be significant: visual appearance, audio properties, user-interface functionality, internal functionality and behavior, and external functionality.

In addition to connections to and from the outside world, *external functionality* encompasses stimuli and functions reacted to or accessed through hardware or software, but not directly by the user. For example, value-added services, software agents, and sensory input represent external functionality in a mobile phone. As Figure 1 shows, VR technology can combine attributes from virtual and physical models

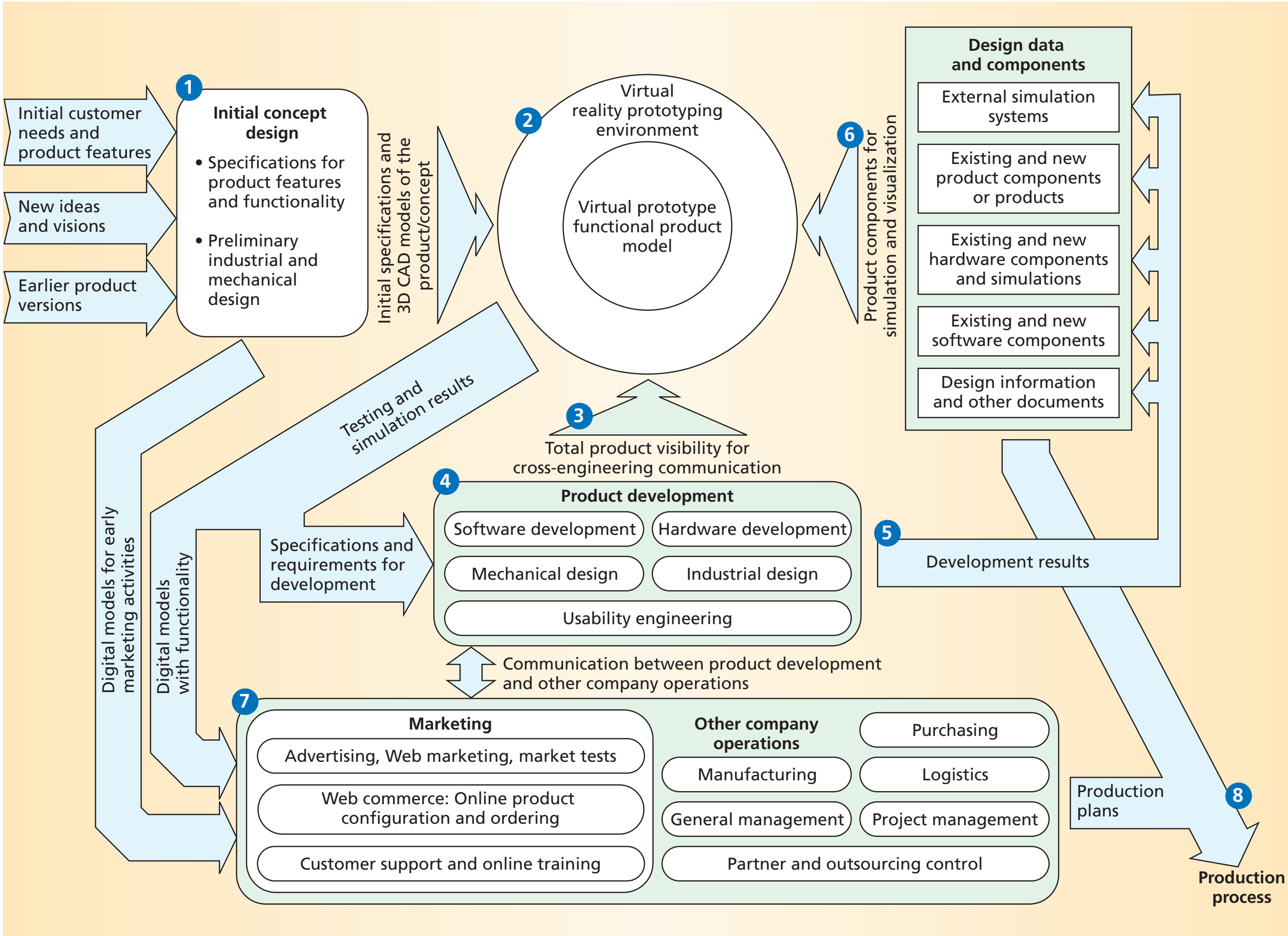

Figure 2. Virtual design process model. The process includes two phases. During initial concept design, developers create the first product concept and place it in the VRP environment. During the virtual design process, developers refine and finalize the product. The numbered circles indicate steps in the design process.

in a simulation. For example, combining tactile and force feedback sensing with stereoscopic viewing in user-to-product interaction can provide new simulation properties.

VIRTUAL PRODUCT DESIGN PROCESS

To further study the possibilities that derive from introducing VRP into an organization, we extended our approach to include virtual product design. In this context, we define *virtual design* as a product development and business process framework that applies VRP in selected process phases and tasks.

The design task extends beyond formal engineering practices to include innovative and creative design work and decision making, especially when the product must achieve high levels of customer satisfaction and appeal. In this sense, our understanding of design parallels Kaneo Akiyama's viewpoint,[11] which emphasizes creative information manipulation and decision making in transforming abstract customer needs into product specifications and components.

Because we believe that product development should have versatile links to and active cooperation with other internal business operations, we do not restrict our definition of virtual design to product development alone. Depending on a company's organizational model, the boundary between product development and other activities can vary. In addition, in the high-technology industry, product development often plays a central role in the organization, and other business activities build upon it to ensure that the business operates profitably. However, product development's role will often change as the company matures.

Basic process model

Figure 2 shows the basic virtual design process model. We describe the model with only one diagram and a very simple notation. The notation does not include any software or other engineering-specific process-flow notations because engineers, managers, and other experts from differing backgrounds should be able to easily understand our general-process framework. This is only a reference process model: In the real world, the implementation and realization of a model always need organization- and business-dependent modifications.

The virtual design process includes two phases. During the first phase, developers create the initial design concept and place it in the VRP environment. The second phase is the actual virtual design process, which finalizes the product. We built this two-phase process model with five main component types:

- *Virtual-reality prototype environment.* This environment forms the core of the process that integrates design information and components into

The information from prototyping and the total product visibility that VRP offers help to advance the design and manufacturing processes.

a common functional and interactive product presentation. It provides participating actors with views of the design target and its status.

- *Actors*. These elements include development teams, company departments, or functions that use design data and the component repository to process information and material and update the virtual reality prototype.
- *Information and material flows*. Light-blue arrows in Figure 2 show the kinds of information or material transferred between actors.

- *Design data and component repository*. The building blocks for the virtual reality prototype include parts produced during the current design process or existing parts recycled from earlier products and projects. The repository content can vary from design documentation to actual product components and can even include external tools and simulation systems.
- *Design target view and status*. This component emphasizes the total product visibility that the integrated virtual prototyping environment and virtual product models offer to design teams and other participating actors.

The numbered circles in Figure 2 denote the steps that comprise our process and the order in which they execute, as follows:

1. *Concept design*. We base the initial concept design on existing business knowledge combined with product versions tailored to match new customer needs. If no existing material is available, we produce the first concept based on new product ideas and visions alone. The amount of effort needed for this phase varies considerably, depending on whether we are designing a completely new product or updating an existing product. This phase produces the initial requirements and specifications for the target product and its first 3D industrial and mechanical design models.
2. *VRP environment*. VRP prototypes simulate and visualize the design results for all participants in a development project. We build the first prototypes from initial 3D CAD drawings, using software engineering tools that support rapid logical modeling to add mechanical functionality and define the product's logical behavior. We add more precise implementation components to these higher-level descriptions later, when, for example, we integrate existing software components and hardware simulations into the prototype.
3. *Total product visibility*. The VRP environment's functional product prototypes offer a base for cross-engineering communication and further product development. Team members can see immediately what they are expected to build and how their own contribution fits into the process.
4. *Development teams*. During iterative development cycles, development teams receive more detailed specifications for their work, mainly from the prototyping system and the marketing department, which can use virtual reality prototypes to acquire precise customer feedback via the Web. This approach also supports user-centered design by continuous usability testing, including ergonomic studies—an activity that benefits from the evolving virtual reality prototype.
5. *Design results*. Development work produces new components, documents, and other material for the design data and component repository. The process model does not restrict implementation to different design areas, but allows spreading the detailed design work across the various teams and expertise areas, matching the work to those most suited to do it at any given time.
6. *VRP environment inputs*. An advanced prototyping system can benefit from several different input sources. Depending on the state of development work for a particular product prototype, the sessions can use existing or new software and hardware components, external and third-party design tools and simulation systems, and existing products or earlier versions of the product under development.
7. *Other company operations*. The full digital product models that VRP provides are useful in early marketing activities. For example, new product versions or ideas can be advertised in advance on the Web. Later, functional models can support e-commerce by providing potential customers with a more concrete concept of the company's product. In addition, the greater detail that virtual reality prototypes provide can enhance Web-based customer support and online user training. The information from prototyping and the total product visibility that VRP offers help to advance the design and manufacturing processes. Similarly, virtual reality prototypes can provide more detailed and clearer specifications for subcontractors.
8. *Exit to production*. Once the developers have evaluated a sufficient number of research-and-development cycles and designed and tested the manufacturing process, production can begin.

VIRTUAL PRODUCT DESIGN TOOL SUPPORT

Although we can use current design tools and IT systems to apply virtual design, taking full advantage of the process requires new tools and methods. VTT

Electronics initiated the development of VRP tools in 1996 with demonstrations and evaluations of different VR technologies. During this early phase, as Figure 3 shows, VTT sought to clarify the possibilities of VR devices: stereoscopic glasses, a data glove for manipulating objects such as a physical mockup of the product, and a haptic device to give tactile and force feedback about objects.

This work revealed that, in addition to being expensive, the available devices were not yet sufficiently practical or efficient for industrial use. For example, the haptic device's point-based touch feedback could not naturally approximate the normal use of a mobile terminal. Over time, the research focus shifted increasingly to an open architecture simulation and design environment for developing electronics and telecommunications devices—the VRP environment.[12] This environment retains the earlier idea of using VR technology, but now as only one of many options for simulating a product's look and user interface.

The VRP environment uses a Java-based toolset that can build a functional system-level simulation of a target product during the very early phase of the development process. The environment supports the following functions:

- *Component-based design.* The environment can integrate existing software and hardware components, user interface components, external simulation systems, and design tools.
- *Different user interface simulation levels.* Developers can use 2D or 3D presentations, VR-based simulations, physical hardware mockups, or combinations of these approaches.
- *Heterogeneous and iterative simulations.* The environment supports using both real product components and simulations. Software simulation components can evolve into real product components.
- *Plug-in architecture.* Developers can easily add new tools, features, design tasks, and areas to the environment.
- *User interface design.* Developers can customize the product's graphical user interface and its physical user interface components. GUI component accuracy equals that of real product components. For example, a developer can define a display's resolution and incorporate the resulting user interface text, icons, and animations directly into real products.
- *Behavior modeling.* A graphical design notation that describes instances of virtual components and their connections can model the product architecture. Likewise, the developer can use a hierarchical state machine formalism to build a product's behavior.

Figure 3. Virtual reality prototyping system used at VTT Electronics in 1996. Built around stereoscopic glasses, a data glove, and a haptic device that gave tactile feedback, this early system proved impractical.

- *Code generation and simulation-model execution.* Developers use the behavior model to generate classes in Java, and can select from different options—such as the user interface simulation type—for the final model.

VIRTUAL DESIGN CASE STUDY

We have demonstrated VRP technology and tools in several publicly conducted pilot projects. These projects covered ground applicable to many industries in which a new product's design must begin before all necessary implementation techniques are available. For example, developing a real-life functional model of the ComPen, a wireless interface device for the cellular phone,[13] was impossible given the technology available when we initiated the project.

Specifying the design flow of a mobile-phone user interface provided one of the first opportunities to apply VRP technology and the virtual design process at Nokia Mobile Phones. We customized VTT's VRP environment for this design domain and tested it with selected design cases developed in concept projects. Figure 4 shows the virtual design process we implemented.

Implementing this process gave us valuable feedback from the industrial world and proved that our virtual design approach and VRP technology did indeed offer advantages by fulfilling many of the product development requirements described in the "Efficient Product Development" sidebar. That a virtual reality prototype can evolve through a design process proved to be the most promising finding. The model, starting from a relatively simple concept design, matured into a functional and detailed prototype, some parts of which actually mirrored the capabilities of real product components.

Comparing the case study results with our requirements demonstrates that virtual design supports the following features:

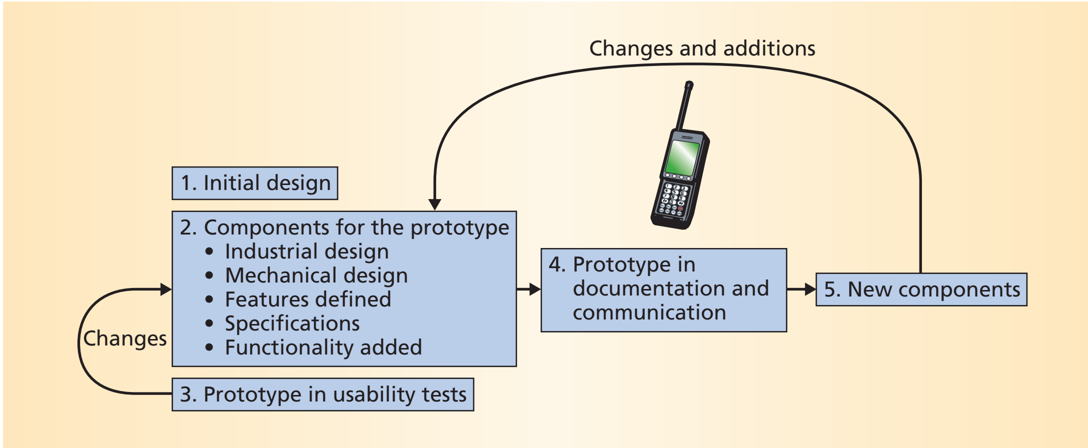

Figure 4. Virtual design phases in the development of a mobile-phone user interface. One of the first industrial applications of VRP technology, this project used a customized VRP environment. (Copyright Nokia Mobile Phones Ltd., 2001.)

- *Multidiscipline cooperation.* In the case study, team members with mechanical, software, and usability engineering; industrial design; and documentation backgrounds developed the user interface.
- *User-centered design.* The technology showed great potential in usability engineering, especially when testing complex product features. For example, when we combine value-added services with usability testing, the role of VRP tools becomes even more promising from the viewpoint of already implemented case designs.
- *Concurrent development.* To cite an example from the user-interface case study, once developers set the screen-size specifications, they could begin the layout design in parallel with the detailed mechanical design.
- *Component reuse.* Developers took advantage of component reuse when, for example, documentation writers and user-manual editors transferred the user-interface layouts from the design directly to their documents.
- *Design domain integration and cooperation between tools.* The VRP tools we used in the case study combined different design phases into a fluent workflow. Likewise, the VRP tools functioned as an integrative platform for other design and IT tools.
- *Cross-engineering simulations and heterogeneous prototyping.* Developers used integrated components from mechanical and software design to build the functional simulations of the mobile phone concept. They also simulated some user interface layouts with hardware mockups that attached real display components to the simulation.

Even though the virtual design of the mobile-phone user interfaces represents only one subprocess of an actual product development process, the experience and results achieved with it were so promising that Nokia Mobile Phones and VTT Electronics continued developing the tools until their commercialization by Cybelius Software. Currently, the use of these tools and practices is expanding from concept to actual products, and from user-interface development to other design areas.

To take advantage of rapidly changing markets and advancing technology within the consumer electronics special product group, companies must increase concept innovation, shorten the concept-to-market time, reduce development costs, and improve the accuracy and quality of development. We base our solution to these requirements on VRP and virtual product design approaches that facilitate designing, testing, and evaluating concepts and products in advance—before creating any physical design models. Further, these simulations can incorporate expected but yet-to-be-developed technology implementations. In general, in addition to the case study we describe, we have found promising results in testing our approach in several other case studies.

Virtual prototype technology also can benefit other company operations: marketing, manufacturing, subcontracting, and distributed work management. Extending the use of interactive, functional, and photorealistic 3D product models into these domains can facilitate effective business practices that cover the entire value chain, from simulated product idea to Web-based customer support of the final product. This wider adoption of the VRP technology and process offers ample scope for further research.

Acknowledgments

The authors thank all the companies and research organizations that participated in the VIRPI, VIRVE, CE-NET and Beyond the GUI projects for their contributions to this article. We also thank Technology Development Centre Finland for its financial support. Finally, we thank Marko Salmela at VTT Electronics for the close cooperation with his VRP and VIRPI projects and his personal support of our work.

References

1. J. Karjalainen et al., "Practical Process Improvement for Embedded Real-Time Software," *J. Quality Eng.*, vol. 8, no. 4, 1996, pp. 565-573.
2. R.I. Winner and J. Pennel, "The Role of Concurrent Engineering in Weapon Systems Acquisition," tech.

report IDA R-338, Inst. for Defense Analyses, Alexandria, Va., 1988.
3. *IEEE Std. 1220*, *IEEE Standard for Application and Management of the Systems Engineering Process*, IEEE Press, Piscataway, N.J., 1998, p. 80.
4. Electronic Industries Alliance, *ANSI/EIA-632*, *Processes for Engineering a System*, Arlington, Va., 1999, p. 131.
5. US Dept. of Defense, *DoD Guide to Integrated Product and Process Development*, v1.0, Office of the Under Secretary of Defense for Acquisition and Technology, Washington, D.C., 1996.
6. B.W. Boehm, "A Spiral Model of Software Development and Enhancement," *Computer*, May 1988, pp. 61-72.
7. J. Connell and L. Shafer, *Structured Rapid Prototyping—An Evolutionary Approach to Software Development*, Prentice-Hall, Englewood Cliffs, N.J., 1989.
8. *ISO/IEC-12207, Informational Technology—Software Life Cycle Processes*, ISO/IEC Copyright Office, Geneva, 1995, p. 57.
9. *ISO/IEC 15288*, "System Life Cycle Processes," to be published by the ISO/IEC Copyright Office, Oct. 2002.
10. E.J. Haug et al., "Virtual Prototyping Simulation for Design of Mechanical Systems," *J. Mechanical Design, Trans. ASME,* vol. 117B, 1995, pp. 63-70.
11. K. Akiyama, *Function Analysis: Systematic Improvement of Quality and Performance*, Productivity Press, Cambridge, Mass., 1991.
12. M. Salmela and H. Kyllönen, "Smart Virtual Prototypes: Distributed 3D Product Simulations for Web-Based Environments," *Proc. 5th Symp. Virtual Reality Modeling Language,* ACM Press, New York, 2000, pp. 87-93.
13. M. Kerttula, M. Salmela, and M. Heikkinen, "Virtual Reality Prototyping—a Framework for the Development of Electronics and Telecommunications Products," *Proc. 8th Int'l Workshop Rapid System Prototyping*, IEEE Press, Piscataway, N.J., 1997, pp. 2-11.

***Mikko Kerttula** is a business development manager at Cybelius Software. The work presented here is based on his positions at VTT Electronics from 1996 through 2001. His research interests include future user-interface technologies, ubiquitous computing, context-aware products, and product development processes. Kerttula received an MS in computer engineering from the University of Oulu. Contact him at Mikko.Kerttula@cybelius.com.*

***Timo Tokkonen** is a senior research manager at Nokia Mobile Phones. His research interests include computer simulations and visualization and the total user experience. Tokkonen received an MS in astronomy from the University of Oulu. Contact him at Timo Tokkonen@nokia.com.*