

# Do software engineers like multimedia?

Masahito Hirakawa

Faculty of Engineering, Hiroshima University  
1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527, Japan  
hirakawa@huis.hiroshima-u.ac.jp

## Abstract

*Multimedia is not a dream anymore. Nowadays we use images (e.g., GIF, JPEG, and BMP), videos (e.g., MPEG, AVI, and QuickTime), and sounds (e.g., MIDI and WAV) as well as alphanumerical information even on a personal computer.*

*People whose research topics relate to computer network, operating system, real-time computing, user interface/HCI, or consumer products have been very active in developing multimedia tools/systems/products, including high speed network, ATM, video servers, streaming video, push technology, computer games, CD-ROM titles, and so on. But how about software engineers? It seems that they are not interested in multimedia. Why? Is software engineering independent from multimedia? I should say no.*

*In this paper, we present a personal view about this point and related topics which includes multimedia software engineering.*

## 1. Introduction

What is multimedia? Ramesh Jain, Editor-in-Chief of IEEE MultiMedia magazine, said in the first issue of the magazine in 1994 that “multimedia is computers using multiple media to let people deal with information naturally. [1]” He also referred to the fable of “Six Blind Men and an Elephant, where each blind man perceived something completely different – the nose as a snake, the leg as a tree trunk, the tail as a rope – and none comprehended the whole. “

Five years have passed since then, but it seems nothing changed. As a fact, Sorel Reisman said in a column of IEEE MultiMedia in the January-March issue of 1998 that “I’d be willing to bet that a survey of 10 different “experts” in multimedia would result in 10 different solutions. [2]” In addition, the column begun with the sentence “Multimedia is dead. In fact, it never really existed.”

What is interesting for me is that, though nobody knows what multimedia is, multimedia is recognized as one of the most important keywords in the computer field in 1990s.

People in both academia and industry have rushed

toward multimedia (who knows the goal?). In fact, network/communication engineers have been very active in developing multimedia related systems. People in a variety of computer fields like user-interface/HCI, database, operating system, and real-time computing have also been interested in multimedia. Artists, medical doctors, school teachers, and many other non-computer specialists go into rhapsodies over multimedia. But how about software engineers? Are they interested in multimedia?

The following is a personal view about this point and some related topics which includes multimedia software engineering. In Chapter 2, we refer to the activities in the software engineering field in these days. Chapter 3 concerns multimedia software engineering, a new trial towards a bridging of the software engineering and multimedia computing fields. What is expected in development of multimedia software will be given in Chapter 4. Finally, in Chapter 5, conclusion is given.

## 2. Where are Software Engineers Headed?

I attended the International Conference on Software Engineering (ICSE) held in Kyoto, Japan in April, 1998. ICSE is the most prestigious conference in software engineering field and this year about 900 people came to Kyoto to see the latest results in this field. 41 technical papers were selected and categorized into the following 15 sessions:

- Experiences with Software Process Improvement
- Formal Modeling
- Reverse Engineering
- Formal Methods and Object Orientation
- Internet/Mobile Code/Security
- Software Evolution
- Testing
- Novel Approach to Design and Validation
- Managing Objects
- Large Scale and Complex System Development
- Computer Supported Cooperative Work
- Object-Oriented Technology
- Project Estimation and Simulation

- Project and Workflow Management
- Estimation Technologies

As you can see, there is no multimedia term. Then I checked the paper title and the keyword list in each paper. Still I couldn't find any paper containing the term, multimedia. This was surprising for me. We see many multimedia applications in the market in these days. I have no idea whether software engineers don't like multimedia, but it might be reasonable to say that multimedia is not a main stream for software engineers.

Why aren't they interested in multimedia? My guess is that, as illustrated in Fig.1, they assume multimedia applications are rather smaller than the applications that software engineers have traditionally treated, and consider multimedia applications to be a research target worth little (Fig.1 (a)). This may be true in some sense. But I would say the difference between multimedia applications and traditional ones is not just the size, but rather the domain of users (Fig. 1(b)).

More specifically, traditional software applications are business-oriented, like ticketing, banking, logistics, and factory automation. Key features of the software in those applications are *simplicity*, *uniformity*, and *efficiency* (Fig. 2). Systems are expected to be simple and uniform for enabling people to perform their tasks with the maximum efficiency.

Meanwhile, though there exist some serious business-oriented applications such as teleconferencing and virtual

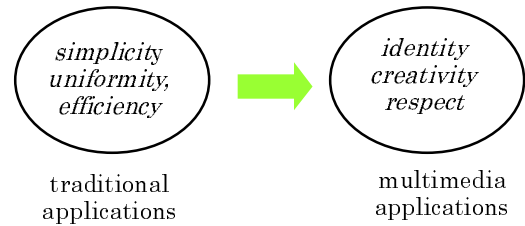


Figure 2 From business-orientation to entertainment-orientation

factory, multimedia applications are basically entertainment-oriented, where *identity*, *creativity*, and *respect* are assumed to be important.

Let me explain a little bit these three key terms. Without having identity in a system, nobody recognizes the system. In addition, the system should help users create new ideas, not just perform a given task [3]. Furthermore the system is expected to be one where software developers respect users who use the system and users respect software developers who developed the system.

This difference between traditional applications and multimedia applications essentially requests us to change the way of developing software. The change cannot be done by simply changing user-interface and/or look-and-feel of the user-interface. Rather we should reconsider the model of developing software.

Here let me give you an interesting observation appeared in the Alan Cooper's book [4]. He said:

“The user's goals are often very different from what we might guess them to be. For example, we might think that an accounting clerk's goal is to process invoices efficiently. This is probably not true. Efficient invoice processing is more likely the goal of the company or the clerk's boss. The clerk is more likely concentrating on goals like

- Not looking stupid
- Not making any big mistakes
- Getting an adequate amount of work done
- Having fun (or at least not being too bored)”

I think this is very significant. User-centered design has been considered effective and valuable. Definitely. The problem is that we are not always seeing *actual* users who use the system. This misunderstanding is somewhat possible because traditional applications are mostly for the company but not for the people. Efficiency is the primary concern in such systems. Software engineering has been investigated under the philosophy, as seen in the Japanese education system, where everything is managed in a

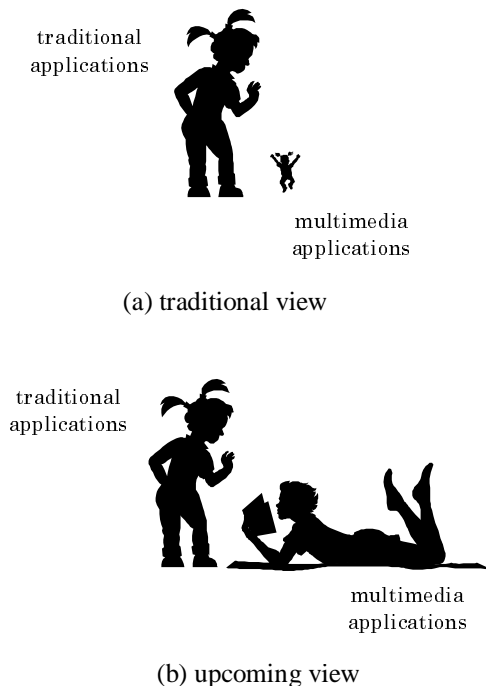


Figure 1 Relationship between traditional applications and multimedia applications

uniform manner to get the best efficiency.

In the following, let me explain a trial called multimedia software engineering to cope with the situation we are faced with.

### 3. Multimedia Software Engineering

Sourav Bhattacharya and I organized as Program Co-Chairs a workshop on Multimedia Software Engineering (MSE'98) held in conjunction with ICSE'98. You may wonder what multimedia software engineering is. Does it mean software engineering for multimedia? Or someone may associate it with engineering for multimedia software, software engineering using multimedia, or some others.

A commonly acceptable interpretation of multimedia software engineering doesn't exist at present. I'd like to classify multimedia software engineering into the following two: 1) exploring the power of multimedia to do a better software/systems engineering, and 2) exploring software/systems engineering issues in the development of systems that involve multimedia.

Examples of the first category can be seen in the research areas of visual languages including visual programming, software visualization, and algorithm animation, for example. Figure 3 shows display snapshots of those systems. Research has been done in this category for almost 20 years. In fact, the Alto computer was developed in the middle of 1970s, and the first IEEE workshop on Visual Languages was held in 1984. Multimodal interfaces would also be recognized to be included in this category.

Meanwhile, as for the trials of systems in the second category, tools for implementing multimedia application software are available in these days. Java is a good example. This is a general-purpose programming language/tool, but has powerful facilities for implementing Web-based multimedia applications.

Multimedia authoring/editing tools like Macromedia Director and Adobe Premiere, or MIDI software including sequencer/software synthesizer are other examples of multimedia application development tools. Figure 4 shows examples of those systems. Interestingly, visual programming scheme is applied as a technique of coding/programming. That is, the user can create multimedia contents visually by arranging media objects on the screen.

One more important trial I should refer to is the Extensible Markup Language, XML, which is a data format for structured document interchange on the Web. This is flexible enough to define languages for a wide variety of applications.

For example, SMIL (Synchronized Multimedia Integration Language) is an XML-based language which allows multimedia developers to create time-based, streaming-type multimedia presentations that combine media objects such as images, video, and audio [5]. The developer can define control of the position and timing of media elements in a multimedia presentation with a simple markup language similar to HTML. Figure 5 shows a display of RealPlayer G2 which is one of the SMIL players available. VoXML is another markup language, which allows developers to easily add speech interfaces to their Web applications [6].



Figure 3 Software development tools/systems using multimedia



Figure 4 Multimedia contents creation tools



Figure 5 A multimedia document specified by SMIL

It is crucial for us to extend “software engineering” issues in the sense that implementation is not the only activity in software systems development. Object-orientation could work as a general framework for multimedia software development, but it is not sufficient. A more advanced, higher-level model, for example, is needed. The musical design pattern proposed by Borchers [7] is one trial towards that goal. Other interesting trials

appear in the proceedings of MSE’98, where the papers are classified into the following four regular sessions:

- Multimedia modeling and specification
- Multimedia systems development
- Metric/evaluation
- Applications

#### 4. Where Should We Be Headed?

Here let me introduce a cross-disciplinary forum called iD-Summit which I have been involved in. “iD” means interface design, and this forum was organized in 1996 for encouraging people having different disciplines to come and act together toward development of new types of user-interface [8]. Almost half of the members are computer engineers, and other half are graphic designers. We have a few other members from publishing, cognitive science, and other related fields.

At the 1997 meeting, a question arose: “Should computer engineers learn graphic design? Should graphic designers learn computer engineering/programming?”

We had active discussion there. Interestingly, most computer engineers said that graphic designers didn’t

need to learn programming. Their idea was “Please wait a while. We’ll provide you a sophisticated tool where the programming skill is not necessary anymore.” On the other hand, graphic designers didn’t mind learning programming. They are willing to study programming to get the better result of design. In fact most of the advanced graphic designers write programs in Java as well as HTML.

Industrial/product designers as well as graphic designers are interested in taking part in the software development (Fig. 6). This fact is explained by the definition of industrial design given by Tomas Maldonado as follows:

“The ultimate aim of industrial design is to determine the quality of form of an industrial product ...The quality of form does not point to external features but to structural and functional relationships that can transform a product into a unified whole that possesses consistency in the eyes of the manufacturer or its user ... The quality of form is a result of the adjustment and unification of various factors – functional, cultural, technical and economic – that in some way or another always relate to the process of form fabrication ... Establishing the quality of form is giving form to the reality which corresponds to an internal organization, that is, the reality which has grown with the product.”

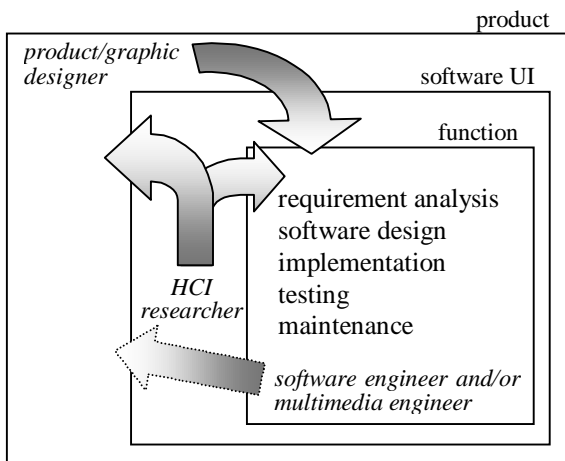


Figure 6 Software as a product

The idea of uniting producers, consumers, and designers together in industrial design came from the work of *Arts and Crafts Movement* initiated by William Morris in the late 19th Century. He tried to make products that are comfortable *for the users*.

In addition, Vertelney and Booker said in [9] that “User interface is more than software – it includes every interaction that a user has with a product. Industrial

design, traditionally concerned with product packaging, can play a valuable role in user interface design by extending the user interface to the physical product and beyond.” They stressed the importance of integrating user interface design and industrial design, by saying

$$\text{User Interface} + \text{Industrial Design} = \text{Whole-Product User Interface}$$

Human computer interaction (HCI) practitioners have also noticed the necessity of broadening the scope. For example Haw mentioned in his article [10] that “HCI practitioners need to take leadership, not just in the design of the software user interface, but the design of the product interface, too.” To do this, he said “For HCI practitioners to be able to apply successfully the methods and techniques to the product interface, they need be positioned in an organization in a way that facilitates maximal interaction and collaboration with the other product groupings – technical writers, marketers, trainers, support staff, etc., and not just with software engineers.”

Software engineers are, unfortunately, not interested in multimedia. Is it fair to sit and wait for the fruits to eat? As was mentioned in [10], “All parts of the product interface are of equal importance and should be designed with every other part in mind so that together they form a gestalt.” Engineers should bear our share of multimedia software research. Let’s take the place of software engineers, shall we?

## 5. Conclusion

The idea of industrial design was born at the period between World Wars I and II, called *the machine age*, where the industrialization came into home. People were allowed to use motor power at their home, resulting in a drastic change of their daily life.

It may be said that we are now standing at the stage of the second machine age, or let me call *multimedia software age*. Multimedia software has been coming into home. It is requested that multimedia software is really usable and fun to anyone who wants to use it.

To realize this, multimedia software must be designed, implemented, and maintained carefully. This is traditionally the topics of software engineering, but software engineers are not interested in multimedia software. Who take the position of software engineers? Industrial/graphic designers or HCI people? *Do you multimedia engineers like software engineering?*

## References

- [1] Jain, R., “Multimedia Computing,” IEEE

- MultiMedia, Vol.1, No.1, pp.3-4, Spring 1994.
- [2] Reisman, S., "Multimedia is Dead," IEEE MultiMedia, Vol.5, No.1, pp.4-5, January-March 1998.
  - [3] Ichikawa, T., "Birth and Growth of the Task Force," Newsletter of the Task Force on Multimedia Computing, Vol.1, No.1, pp.3-4, June 1993.
  - [4] Cooper, A., "About Face: The Essentials of User Interface Design," IDG Books Worldwide, Foster City, 1995.
  - [5] <http://www.w3.org/>
  - [6] <http://voxml.motorola.com/>
  - [7] Borchers, J. and Muhlhauser, M., "Design Patterns for Interactive Musical Systems," IEEE MultiMedia, Vol.5, No.3, pp.36-46, July-September 1998.
  - [8] <http://id-summit.fishbone.tm/> (in Japanese)
  - [9] Vertelney, L., and Booker, S., "Designing the Whole-Product User Interface," in The Art of Human-Computer Interface Design (ed. by Laurel, B.), pp.57-63, Addison-Wesley Publishing, Massachusetts, 1990.
  - [10] Haw, D. C., "Designing the Product Interface," ACM interaction, Vol.5, pp.9-12, September + October 1999.