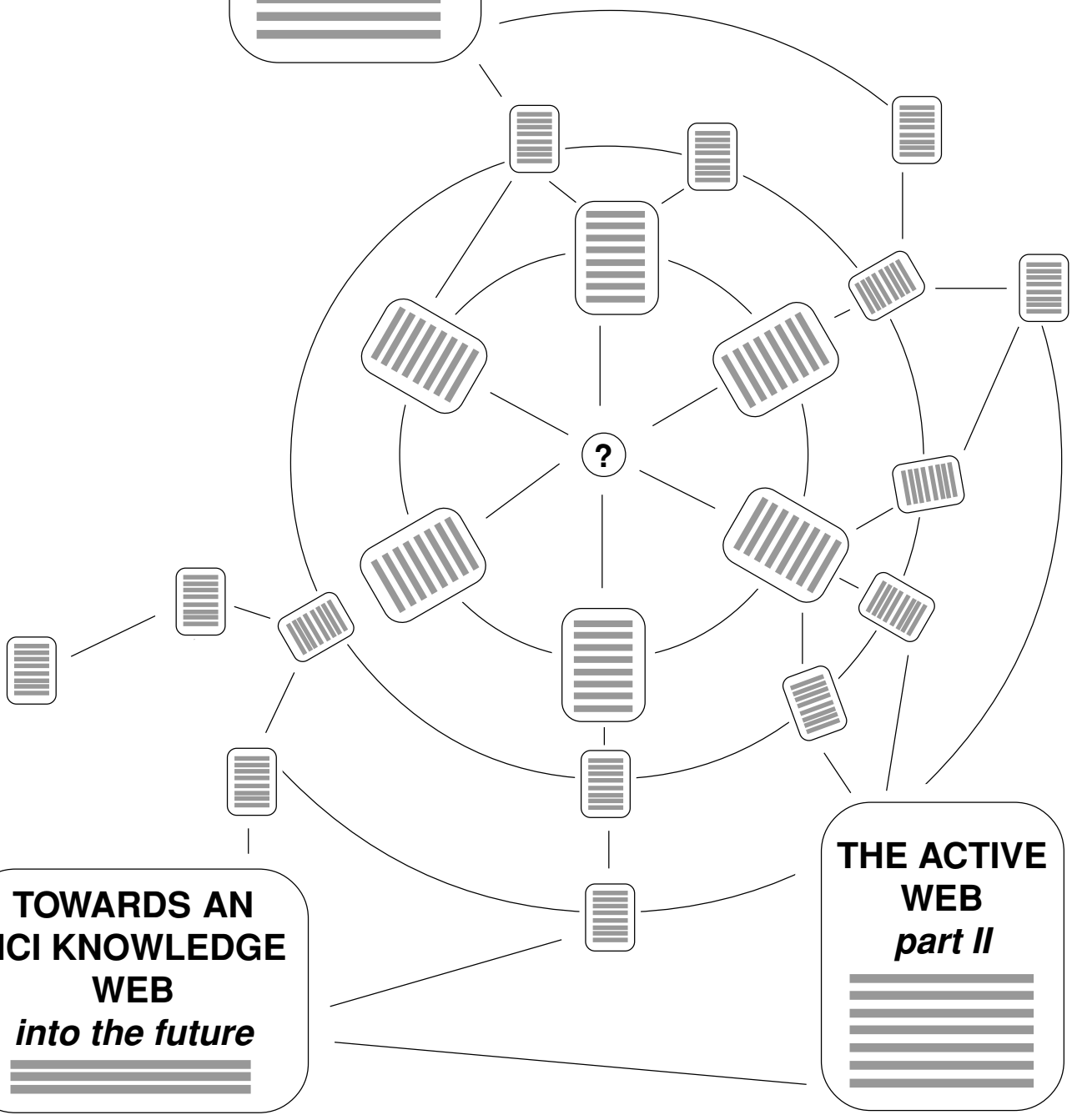
Interfaces

Interfaces 39

AMULET and GARNET *prototype-based programming*



TOWARDS AN HCI KNOWLEDGE WEB *into the future*

THE ACTIVE WEB *part II*

Published by the British HCI Group ISSN: 1351-119X

contents

From the INTERACT '99 Chair

The summer of 1990 is still remembered by some as one of the high points of HCI conference life in the UK. INTERACT '90 in Cambridge. Pints by the Cam, punting on it, a marquee full of "visual" papers, Dan Diaper holding court at the entrance. Unbeatable. But then there was HCI '91 in Edinburgh. Another conference of fond memory. Wonderful keynotes, Festival nights until nearly dawn. Well, here we go again. In 1999 you're being offered the heady mixture of INTERACT '99, HCI '99, Edinburgh and its International Festival, all wrapped up in one package.

This is our conference. Operating under the umbrella of IFIP Technical Committee 13, INTERACT '99 is hosted in the UK by the British Computer Society through the agency of the British HCI Group.

Keynote speakers will include pioneers such as Brian Gaines, who started the first HCI journal in 1969, innovators such as Joy Mountford who contributed so much to the success of Apple in changing the way we think about interaction, and entrepreneurs like Karen Holtzblatt bringing new insights into design support for a wide range of clients. The Professional Practice and Experience track offers usability professionals the opportunity to share experience and present new ideas, and the Panels will support lively debate on key issues and significant trends. To allow more doctoral students to enjoy the benefits of expert feedback and peer discussion of their work, there will be two instantiations of the Doctoral Consortium, one on Monday 30th and one on Tuesday 31st August. And for intensive, in-depth exploration of research, teaching, theory, products, or practice, the Workshop track provides a flexible and effective structure for small groups to meet together for one or two days before the start of the main conference sessions.

INTERACT '99 will take full advantage of the fact that it coincides with the third week of the Edinburgh International Festival of the Arts. The conference dinner will be in the Royal Scottish Museum, preceded by a tour of the adjoining new National Museum of Scotland, due to open on November 30th this year, marking the culmination of one of the largest and most ambitious museum projects in Europe in recent years and incorporating sophisticated and innovative multimedia presentations. After dinner there will be the Edinburgh Festival Fireworks which we hope to be able to view from the roof garden of the new Museum.

The Best HCI Conference Ever? It's Up to YOU!

The message is: start planning and, more importantly, *start writing*. The deadline for full papers is January 25th next year, and for posters May 10th. We would particularly like to encourage early submission of panel and workshop proposals. Although these can be accepted up till May 10th, there will be a first round of reviewing for proposals received by the earlier deadline of January 25th; proposals accepted following this review will be included in the advance programme. Think creatively and soon about panels and workshops, and get your ideas to us as soon as possible. The co-chairs (David Benyon and Dianne Murray for Panels, Alistair Sutcliffe and Alan Newell for Workshops) will be pleased to discuss your ideas at an early stage, and help in getting your proposal into a shape which will maximise its chances of acceptance. Early commitment to panels and workshops will help us to achieve an attractive advance programme, as well as ensuring that a wide range of delegates buy in early to conference participation.

And of course the other key delegate attractor will be tutorials. An attractive package is on offer for prospective tutorial providers. The deadline for proposals is 25th January. Being a tutorial presenter is an excellent way of subsidising the cost of attending. The tutorials chair, Janet Finlay, will be happy to discuss your ideas with you at an early stage.

The setting for INTERACT '99 is outstanding but its success depends on you. Give us your best thoughts, words, and ideas, and spread the word to all your international contacts. Help us to mark the closing months of the decade, century, and millennium with the best HCI conference ever.

For more information, see the Call for Papers on page 12.

Editorial

Welcome to Issue 39 of Inter*faces*. Predicting the future development of the web is, as Simon Buckingham Shum says in his article, 'a tricky business' and we will not attempt it here. But this issue presents a number of contributions pertinent to this debate. Alan Dix offers the second part of his review of 'The Active Web', looking at the range of technology now available for increasing the interactivity and maintainability of web pages. Simon considers the potential of the web for supporting research far beyond the current dissemination of information. Please read his article and take part in his pilot study – it presents an exciting opportunity for the HCI community.

Many of you will be familiar with the Garnet and Amulet interface development environments. Brad Myers and his team at CMU present an overview of this major 10-year project and the benefits and drawbacks of this mode of development, as part of the *Software Support for HCI* series. If you have any contributions in this area please contact Dave Clarke (see page 9).

The next issue of Inter*faces* is rather special. It represents 10 years of the British HCI Group magazine and to mark this we are putting together a special issue looking back over the past 10 years and more – and looking forward to where HCI is heading. We are particularly keen to track down some of the very early issues – if you have Issue 1 or 2 somewhere on your bookshelves we want to hear from you! More about this in the box below.

Janet Finlay
Editor

RIGHT TO REPLY

Make Inter*faces* interactive! We invite you to have your say in response to issues raised in Inter*faces* or to comment on any aspect of HCI that interests you. Submissions should be short and concise (500 words or less) and, where appropriate, should clearly indicate the article being responded to. Please send all contributions to the Editor.

With thanks to:
commissioning editors: Dave Clarke and Alistair Kilgour. Inter*faces* is looking for additional commissioning editors. Please contact the editor for details.

To receive your own copy of Inter*faces*, join the British HCI Group by filling in the form on page 14 and sending it to the address given.

NEXT ISSUE

We would like your reminiscences from the last 10 or more years of HCI – of the personalities, the conferences, the British HCI Group, the magazine, or whatever is memorable for you. And if you have any predictions for the future of HCI then why not send those too. You don't have to write a long article – a brief paragraph or two is fine – we will see if any of us agree on this one! The deadline is **December 30th** – we look forward to hearing from you.

Deadline for issue 40 is **December 30th**. Deadline for issue 41 is **March 31st 1999**. Electronic versions are preferred: RTF, plain text or MS Word (5/6), via electronic mail or FTP (mail fiona@hiraeth.com for FTP address) or on Mac, PC disks; but copy will be accepted on paper or fax.

Send to: Inter*faces*, c/o Janet Finlay, School of Computing and Mathematics, University of Huddersfield, Queensgate, Huddersfield HD1 3DH
Tel: +44 (0)1484 472913; Fax: +44 (0)1484 421106; Email: j.e.finlay@hud.ac.uk

and **copy email submissions** to Fiona Dix, Inter*faces* production editor; email: fiona@hiraeth.com

Prototype-Instance Object Oriented Programming in Amulet and Garnet

Abstract

Over the last 10 years, the CMU User Interface Software Project has been investigating prototype-based programming in two large-scale systems: Garnet in Lisp and Amulet in C++. The goal of these systems is to provide an effective way to prototype and implement user interface software. In addition to using a prototype-instance object model, these systems also use constraints to tie objects' values together, and new models for input and output. The result is a significantly different style of programming than conventional class-based object systems.

An earlier and much longer version of this article appeared as "The Prototype-Instance Object Systems in Amulet and Garnet" in James Noble, Antero Taivalsaari and Ivan Moore, eds., *Prototype Based Programming*, Springer-Verlag, 1999.

This research was sponsored by Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597, and NCCOSC under Contract No. N66001-94-C-6037, Arpa Order No. B326. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NCCOSC or the U.S. Government.

Introduction

We have built two user interface development environments that use prototype-instance object models. Garnet [7], is in Lisp and was started in 1988. Amulet [10], is in C++ and was started in 1994. The goal of these systems is to investigate ways to make the creation of user interface software significantly easier, especially for highly interactive, graphical, direct manipulation user interfaces. These systems also use constraints, which are relationships that are declared once and maintained by the system, to tie objects' values together. Amulet and Garnet also include many other innovations, which are described in various papers. This article concentrates on the prototype-instance object systems in Garnet and Amulet.

In Garnet's and Amulet's prototype-instance object systems, there is no concept of a "class" since every object can serve as a prototype for other objects. Values of objects are stored in "slots" (sometimes called "fields" or "instance variables"), each of which has a name and can hold a value of any type. Any slots that are not declared locally in an object are inherited from the prototype. Another important feature of these object systems is that there is no distinction between methods and data: individual objects can override an inherited method in the same manner as inherited data (unlike class-instance systems where a different method requires a sub-class, whereas each instance can have different data). The object systems are dynamic in that slots in objects can be created, set, and destroyed at run-time, and the types of values in slots can also change (for example, the value in a slot can change from a string to a float). The object systems also support a part-owner hierarchy, by which objects can be grouped together. For instance, the graphics in a window are added as *parts* of the window.

Amulet and Garnet also provide *constraints* which can be arbitrary code. Any slot of any object can contain a constraint, rather than a regular value. We have found that the style of writing code in these systems is quite different than in other systems. By writing constraints, the programmer can specify relationships that should hold in the interface in a *declarative style*, and leave it up to the system to maintain them.

For Amulet, we took the opportunity to fix a number of problems we experienced with Garnet, and Amulet also contains a number of important innovations, including incorporating the part-owner hierarchy into the object system, control over the inheritance of slots, support for multiple constraint solvers, and a flexible demon mechanism. In addition, Amulet's default constraint solver is more flexible than other one-way systems. Finally, it is interesting to note that we were able to provide dynamic slot typing, a dynamic prototype-instance system, and constraints in C++ without using a pre-processor or a scripting language.

Summary of the Garnet and Amulet systems

The Garnet and Amulet user interface development environments aim to make the design, prototyping, implementation, and evaluation of user interfaces significantly easier, while supporting flexible experimentation with new styles of interaction. The term "user interface development environment" is used to signify that these systems are more than just a collection of widgets (also called "controls") like menus, buttons and scroll bars, like the Macintosh or Motif widget sets. Instead, Garnet and Amulet are higher-level tools that provide support for creating the complete application, and therefore might be classified as "application frameworks" comparable to MacApp [16] or the Microsoft Foundation Classes. For a complete discussion of user interface tools, see [6].

Amulet and Garnet include a number of design and implementation innovations including new models for constraints, objects, input, output, commands, undo, and animation. An important goal is to support new kinds of *interactive tools* where the designer can create much of the user interface for a system by direct manipulation or by demonstration [5] rather than by writing code. For example, we created a variety of "interface builders" which allow widgets to be laid out interactively using the mouse, and their parameters can be set, in the style of Visual Basic. Other tools allow constraints and behaviours to be demonstrated without writing code.

Garnet, which stands for Generating an Amalgam of Real-time, Novel Editors and Toolkits, is in Common Lisp and runs on Unix X/11 and the Macintosh. Amulet, which stands for Automatic Manufacture of Usable and Learnable Editors and Toolkits, is in C++ and runs on Unix X/11, Microsoft Windows 95 and Window NT, and the Macintosh.

Brad A. Myers, Rich McDaniel, Rob Miller, Brad Vander Zanden, Dario Giuse, David Kosbie and Andrew Mickish

We distribute the systems for use by others because we feel this will help to demonstrate that the innovations we have incorporated are sound and effective, and will hopefully facilitate technology transfer. Both systems are in the public domain and can be used for free. To get Garnet, see `http://www.cs.cmu.edu/~garnet`. To get Amulet, see `http://www.cs.cmu.edu/~amulet` or send mail to `amulet@cs.cmu.edu`. Amulet is being downloaded about 1000 times a month.

Why a new object system?

We did not set out to investigate new object models, but we decided that existing object systems were not sufficiently flexible to support our requirements for user interface tools. When we started Garnet, the standard Common Lisp Object System (CLOS) was not yet available, and after it appeared we decided that it did not fully meet our needs anyway. For Amulet, we found the C++ object system was much too restrictive.

Our particular requirements for an object system are:

- Class creation and reflection at run-time.
- Querying the names of slots to find out what slots are in an object dynamically.
- Dynamic typing a slot can contain different types of values.
- Each instance can have different methods, as well as different data.
- Support for constraints.

Object system design

The main concept in Garnet and Amulet is that everything is represented as an object which has a set of slots. The objects support completely dynamic redefinition of prototypes with automatic change propagation to the prototype's instances. An instance can add any number of new slots, and slots that are not overridden in an instance inherit the values from the prototype. In fact, the inheritance can change dynamically, as an object can add or remove slots at any time. There is no distinction between data and method slots. Any slot can hold any type of value, and a method is just a type of value. This allows the methods that implement messages to change dynamically, which is not possible in conventional object systems like C++. The ability to dynamically add, delete, and modify methods has proven important in graphical interface builders since they often will temporarily insert their own methods during "build" mode, and then remove them during "test" mode.

A new object is created by making an instance of another object, which is called the *prototype*. An instance starts off inheriting everything from the prototype, and the slots can then be set with new values. Alternatively, an object can be *copied* (or "cloned"), in which case it immediately gets a copy of all values in the original.

When an instance of an object is created, the slots that are not set locally inherit their values from the prototype object. If a slot of the prototype is changed, then the value also changes in all of the instances that do not override that property. The inheritance mechanism is an important distinction from other prototype-instance object systems, such as Self [2, 13], in which all the slots are always *copied* into instances so changes to prototypes never affect instances. Although Amulet's model requires more overhead, we think it is useful for prototyping to be able to change properties of prototypes and see the effect on all instances immediately. Many Garnet and Amulet applications have taken advantage of this capability.

An important feature of Garnet's and Amulet's object systems is that there is no distinction between the inheritance for *methods* and *data*: any instance can override an inherited method in the same manner as inherited data. In a conventional class-instance model such as Smalltalk, C++, or CLOS, instances can have different data, but only subclasses can have different methods. Thus, in cases where each instance needs a unique method, conventional systems must use a mechanism other than the regular method invocation, or create a new subclass and a single instance of that subclass each time. For example, a button widget might use a regular C++ method for drawing but would have to use a different mechanism for the call-back procedure used when the user clicks on the button, since each instance of the button needs a different call-back. In Amulet and Garnet, the draw method and the call-back use the same mechanism. Methods can be set into a slot in the same way that data is set into a slot.

An innovation in Garnet and Amulet is the support for "structural inheritance," where the parts of an object can also be inherited, not just the values. In Garnet, this is supported by special `aggregate` objects, and all of the parts of an `aggregate` have to be graphical objects. In Amulet, the part-owner hierarchy is available for *any* type of object, and we have found many uses for the part-owner hierarchy that are independent of graphical relationships. When an instance is made of an object which contains parts, the new instance will have instances of all the prototype's parts.

There are many advantages of the prototype-instance model. Having no distinction between classes and instances, or between methods and data, means that there are fewer concepts for the programmer to learn and a consistent mechanism can be used everywhere. Another advantage of the prototype-instance object system is that it is very dynamic and flexible. All of the properties of objects can be set and queried at run-time, and interactive tools can easily read and set these properties. In fact, most of today's toolkits implement some form of "attribute-value pairs" to hold the properties of the widgets, but our object system provides significantly more flexibility and capabilities.

Another advantage of our prototype-instance object systems over C++ is the ability to treat "classes" as first-class objects. In C++, one cannot store a class object in a variable so that different kinds of objects can be created at run-time. For example, C++ does not allow code like:

Prototype-Instance Object Oriented Programming in Amulet and Garnet

```
obj_to_create = Rectangle; //NOT ALLOWED IN C++
new_object = new obj_to_create; //NOT ALLOWED
                                    IN C++
```

Instead, the operand of the `new` operator must be a fixed class, leading to large `case` statements and other inflexible and error-prone constructs. In contrast, since one can create an instance of any object in a prototype-instance object system, one can store a reference to an object in a variable and later use the variable to create a new object:

```
Am_Object obj_to_create = Am_Rectangle; //Typical
                                        Amulet code
Am_Object new_object = obj_to_create.Create();
```

Although it was designed to support the creation of graphical objects, many users have discovered that the prototype-instance object system is useful for representing their internal application data. The flexibility and dynamic nature of the objects make them ideal when varied and changing data types are necessary.

Constraints

A very significant difference between the Garnet and Amulet object systems and others is the support for *constraints*. Constraints result in a quite different programming style, and they influence other aspects of the object system design as well.

Garnet and Amulet both support *formula* constraints, which act like spreadsheet formulas. This means that instead of containing a constant value like a number or a string, any slot of any object can contain an expression which computes the value. If the expression references slots of other objects, then when those objects are changed, the expression is automatically re-evaluated. If the other objects' values have not changed, then the formula is *not* re-evaluated and a cached value is returned instead. Thus, the constraints are primarily "one-way". Formula expressions can contain arbitrary code. Amulet also supports multiple *kinds* of constraints, including multi-way constraints and animation constraints, which are discussed below.

Although many other research systems have provided constraints, Garnet was the first to truly integrate them with the object system and to make them general purpose. They are also well integrated in Amulet. An important result of this is that constraints are used throughout the systems in many different ways. For example, the built-in `text` object has constraints in its width and height slots that compute its dimensions based on the current string and font.

An interesting observation about this use of constraints is that it allows *arbitrary* delegation of values, not just from prototypes. Any slot can get its value from any slot of any other object through constraints. Therefore, the constraints can be used as a form of inheritance. Of course, constraints are more powerful than conventional inheritance since they can perform arbitrary transformations on the values.

Indirect constraints

The Garnet constraint system was the first to allow the dynamic computation of the objects to which a constraint refers, so a constraint can not only compute the value to return, but also *which objects* and slots to reference. This allows such constraints as "the width is the maximum of all the components", which will be updated whenever components are added or removed as well as when one of the components' position changes. This capability is also provided by Amulet.

These "indirect constraints" [15] are a form of "procedural abstraction" since the constraints can be thought of as relationships that can be reused in multiple places, with different values for their parameters. In fact, there is a library of predefined constraints, which can be used for many of the basic relationships frequently found in user interfaces.

Multiple solvers

An ongoing research area in user interface software is creating new kinds of constraint solvers (e.g. [3, 4, 14]). One Ph.D. research project was to try to add a multi-way constraint solver into Garnet that would cooperate with the existing formula solver [12]. The difficulty of this task in Garnet inspired us to create an architecture in Amulet that allows multiple constraint *solvers* to coexist. Currently, in addition to the one-way solver described above, Amulet supports a multi-output, multi-way solver called a "web"[1] and an animation constraint solver.

A web constraint can have an arbitrary number of input and output slots, and it can dynamically compute the dependencies like formula constraints. Webs also keep track of the order that dependencies change. We use this solver to keep the various slots of lines and polygons consistent.

We have also created a novel *animation constraint solver* for animating objects [11]. Adding animation to interfaces is a very difficult task with today's toolkits, even though there are many situations in which it would be useful and effective. Amulet's animation constraints detect changes to the value of the slot, immediately restore the original value, and cause the slot to take on a series of values interpolated between the original and new values. Animations can be attached to any object, even existing widgets from the toolkit, and any type of value can be animated: scalars, coordinates, fonts, colours, line-widths, vertex lists (for polygons), booleans (for visibility)[2], and so on.

Debugging

Debugging interactive applications requires more mechanisms than supplied with conventional development environments. Garnet and Amulet provide an interactive *Inspector* that displays the object's properties, traces the

1. Web constraints are not related to the "world-wide-web."
2. Special animation constraints are used for the Visible slot which cause the object to fade out or fly off the screen. The value of the Visible slot itself therefore only changes when the animation is complete.

Brad A. Myers, Rich McDaniel, Rob Miller, Brad Vander Zanden, Dario Giuse, David Kosbie and Andrew Mickish

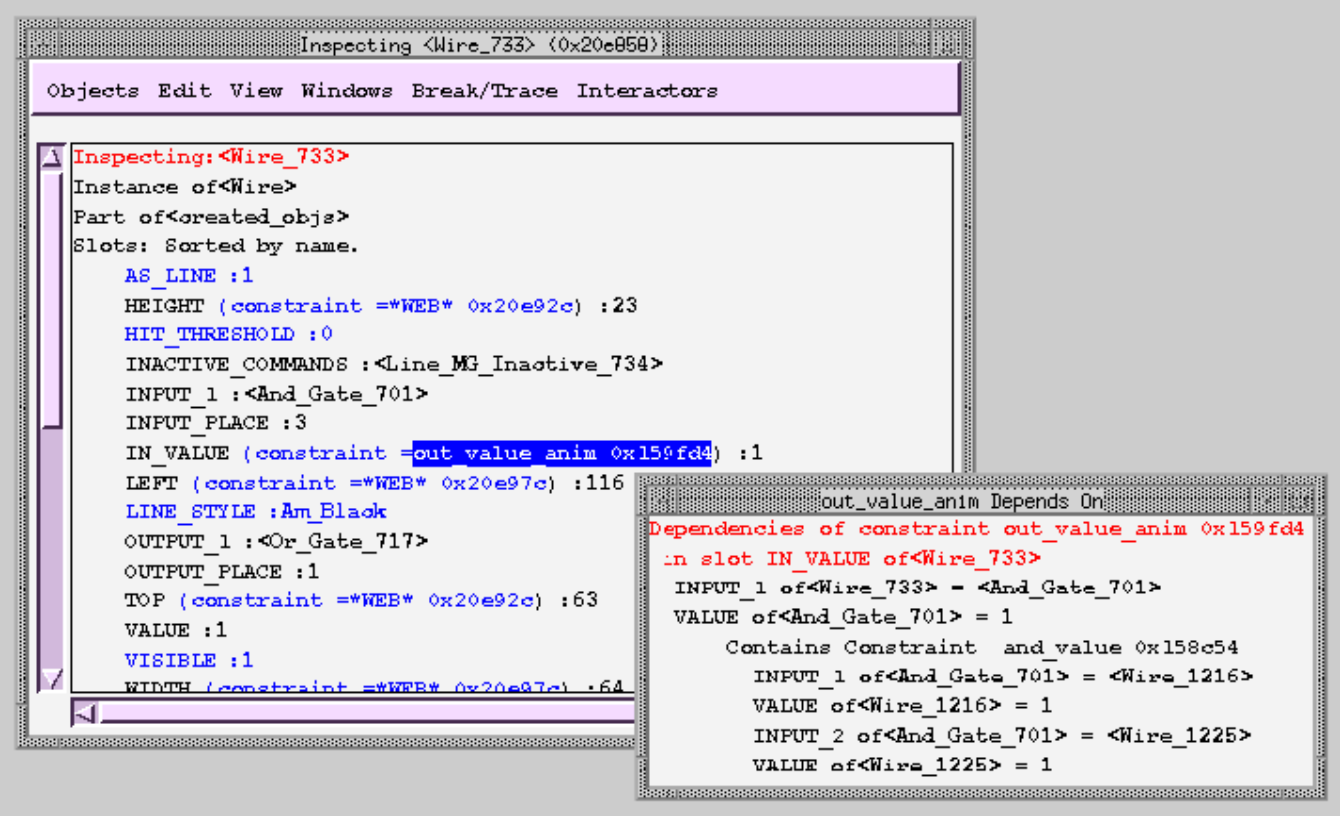


Figure 1 **The Inspector in Amulet, viewing a wire object, and the dependencies of the constraint in its IN_VALUE slot. The constraint is named out_value_anim and it depends on two slots: INPUT_1 of a wire and VALUE of an And_Gate. The value of the And_Gate in turn contains a constraint. In the main Inspector window, the slots which are inherited are shown in blue on the screen. Notice that the high-level names are shown for methods, constraints and objects.**

execution of interactive behaviours, and displays the dependencies of constraints (see Figure 1). From the Inspector, programmers can also set breakpoints or have messages printed whenever the value of a slot changes. The Inspector is also useful for development, since the developer can experiment with different values for colours, positions, and other parameters.

Evaluation

We have now been programming with prototype-instance object systems and constraints for almost 10 years. This section discusses some of our opinions about the advantages and disadvantages.

- The ability to dynamically create new slots in objects at run-time is very convenient, and allows the system to add annotations and other new information to built-in objects. One disadvantage of this is dealing with the problem of name clashes: we have occasionally had two different parts of the system try to use the same slot name in an object for different purposes.
- The ability to query all properties of all objects at run-time is crucial to making the systems, and applications created using them, easier to debug. Keeping around sufficient information to find out the list of all slots of an object, all instances created of a prototype, all parts of a group, and the names of everything, makes most of the properties and data visible to the programmer. However, this takes up a great deal of space, and also makes programs run slower. Therefore, we have provided mechanisms to compile the libraries without the debugging information, for delivering applications.
- Having the primary interface to objects be a set of values which are set and accessed, rather than a set of methods that are called, makes many operations much easier. Constraints can be used to compute those values or to use them in other

calculations. The objects can be automatically animated by simply setting the slots with a sequence of values, without any need for the object to even know it is being animated. Furthermore, Amulet's general undo facility takes advantage of this feature by remembering the old values of each changed object, and then just restoring the values if the user asks for an Undo [9].

- An interesting disadvantage of the prototype-instance object system is that it is harder to see the specification of an object from the code. Slots can be added anywhere, and there is no clear declaration of the internal versus external slots in Garnet or Amulet. In the exported header files, the main object names are just listed, with no indication of what slots control them. Therefore, discovering which slots are in each object, and what each slot does, usually requires carefully reading the documentation.
- Although we feel that the performance of Garnet and Amulet is adequate, the prototype-instance object system is still quite a bit slower than the "native" object systems (CLOS and C++). This is due to the dynamic look up of slots, and other overhead for supporting querying and constraints. The space overhead of objects is still quite large, with a typical object, containing 20 slots with a dozen formulas, being about 1K bytes. Thus, it would not be appropriate to use Garnet or Amulet objects if 100,000 were needed. Typical large Garnet and Amulet applications today would have 2000 to 5000 objects. We are investigating ways to provide "glyphs" [1], which are very small objects for these situations [8].
- We have worked hard to provide a reasonable syntax for objects in C++, but we are still not entirely happy. Beginners often have trouble with the syntax, and small slips and typos can result in unintelligible compiler error messages, or even worse, code which compiles but does unexpected things. Designing a language from scratch, as was done for Self, would be much more appealing, but then the resulting system might not be able to be used by as many people.

Conclusions

The style of programming in the Garnet and Amulet object systems is quite different from other object systems: the programmer collects together graphical objects, and then writes constraints to define the relationships among them. Much of the design of user interfaces can be done with graphical, interactive tools, rather than by writing code. Even when not using interactive tools, programmers rarely write methods when creating Garnet and Amulet code. Our experience suggests that using a prototype-instance object system is very effective for graphical user interfaces, and we continue to investigate ways to enhance its advantages and overcome its weaknesses.

Acknowledgments

We want to thank the many developers and users of Garnet and Amulet over the years. For help with this article, we would like to thank Randy Smith.

References

1. Calder, P. R. and M. A. Linton (1990). Glyphs: Flyweight Objects for User Interfaces. *Proceedings UIST'90: ACM SIGGRAPH Symposium on User Interface Software and Technology*, Snowbird, Utah, 92–101.
2. Chambers, C., D. Ungar and E. Lee (1989). An Efficient Implementation of Self, a Dynamically-Typed Object-Oriented Language Based on Prototypes. *Sigplan Notices*, 24(10), 49–70. ACM Conference on Object-Oriented Programming; Systems Languages and Applications; OOPSLA'89.
3. Gleicher, M (1993). A Graphics Toolkit Based on Differential Constraints. *Proceedings UIST'93: ACM SIGGRAPH Symposium on User Interface Software and Technology*, Atlanta, GA, 109–120.
4. Hudson, S. E. and I. Smith (1996). Ultra-Lightweight Constraints. *Proceedings UIST'96: ACM SIGGRAPH Symposium on User Interface Software and Technology*, Seattle, WA, 147–155. <http://www.cc.gatech.edu/gvu/ui/sub_arctic/>
5. Myers, B. A. (1992). Demonstrational Interfaces: A Step Beyond Direct Manipulation. *IEEE Computer*, **25**(8), 61–73.
6. Myers, B. A. (1995).User Interface Software Tools. *ACM Transactions on Computer Human Interaction*, **2**(1), 64–103.
7. Myers B. A., D. A. Giuse, R. B. Dannenberg, B. Vander Zanden, D. S. Kosbie, E. Pervin, A. Mickish and P. Marchal (1990). Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces. *IEEE Computer*, **23**(11), 71–85.
8. Myers, B. A. et al (1994). *Making Structured Graphics and Constraints Practical for Large-Scale Applications.* Carnegie Mellon University Computer Science Department, CMU-CS-94-150. Also appears as CMU-HCII-94-100.
9. Myers, B. A. and D. Kosbie (1996). Reusable Hierarchical Command Objects. *Proceedings CHI'96: Human Factors in Computing Systems*, Vancouver, BC, Canada, 260–267.
10. Myers, B. A. et al (1997). The Amulet Environment: New Models for Effective User Interface Software Development. *IEEE Transactions on Software Engineering*, **23**(6), 347–365. June.
11. Myers, B. A. et al (1996). Easily Adding Animations to Interfaces Using Constraints. *Proceedings UIST'96: ACM SIGGRAPH Symposium on User Interface Software and Technology*, Seattle, WA, 119–128. <http://www.cs.cmu.edu/~amulet>
12. Sannella, M. (1994). *Constraint Satisfaction and Debugging for Interactive User Interfaces.* Ph.D. Thesis, Dept. of Computer Science and Engineering, University of Washington, Seattle, WA. TR 94-09-10.
13. Ungar, D. and R. B. Smith (1987). Self: The Power of Simplicity. *SIGPLAN Notices*, **22**, 241–247. ACM Conference on Object-Oriented Programming, Systems, Languages and Applications; OOPSLA'87.
14. Vander Zanden, B. (1995)*An Incremental Algorithm for Satisfying Hierarchies of Multi-way, Dataflow Constraints.* Computer Science Department, University of Tennessee.
15. Vander Zanden, B. et al (1994). Integrating Pointer Variables into One-Way

Constraint Models. *ACM Transactions on Computer Human Interaction*, **1**(2), 161–213.

16. Wilson, D. (1990). *Programming with MacApp*. Reading, MA: Addison-Wesley Publishing Company.

About the Author

Brad A. Myers is a Senior Research Scientist in the Human–Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University, where he is the principal investigator for various projects including: User Interface Software, Demonstrational Interfaces, Natural Programming, and the Pebbles PalmPilot Project. He is the author or editor of over 175 publications, including the books "Creating User Interfaces by Demonstration" and "Languages for Developing User Interfaces", and he is on the editorial board of five journals. He has been a consultant on user interface design and implementation to 30 companies. Myers received a PhD in computer science at the University of Toronto where he developed the Peridot UIMS. He received the MS and BSc degrees from the Massachusetts Institute of Technology during which time he was a research intern at Xerox PARC. From 1980 until 1983, he worked at PERQ Systems Corporation. His research interests include User Interface Development Systems, user interfaces, Programming by Example, programming languages for kids, Visual Programming, interaction techniques, window management, and programming environments. He belongs to SIGCHI, ACM, IEEE Computer Society, IEEE, and Computer Professionals for Social Responsibility.

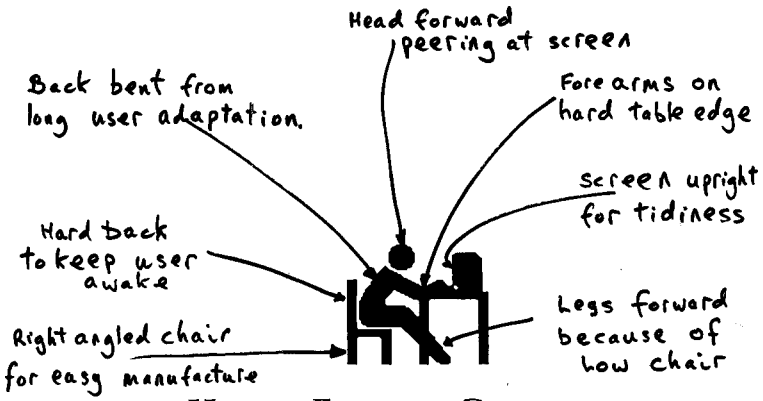
Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891
bam@cs.cmu.edu
http://www.cs.cmu.edu/~amulet

Wanted – articles on Software Support for HCI. The Software Support series gives leading practitioners and researchers the opportunity to discuss how user interface software tools, along with supporting methods and techniques, can aid in the production of good human-computer interfaces. Possible topics include:

- User interface specification, design and construction tools
- Web design and authoring tools
- Specification and design methods to support their use
- Tools which aid in interface evaluation and testing
- Case studies on such tools and their success (or not, as the case may be!)
- Intelligent and adaptive front-ends
- Visual Programming
- Programming by example and demonstration systems

This list is not exhaustive: any article that fits under the heading 'Software Support for HCI' will be considered for publication. Please send submissions to: Dave Clarke; email: dclarke@visualize.uk.com (or on disk c/o Inter*faces*, address on page 3). Articles should be sent in RTF, MS Word or straight ASCII format. Length should not exceed 3000 words. Figures and references may be included where appropriate.

Notes for Authors are at <http://www.visualize.uk.com/bcshci/interfaces>



HUMAN FACTORS GUIDELINES
for the design of
COMPUTER - BASED SYSTEMS

sent in by John Harrison at BAe SEMA, New Malden

My Thesis

Joan Aarvold

Name and address of researcher
Joan Aarvold
Faculty of Health, Social Work and Education
University of Northumbria at Newcastle
Tel: +44 (0)191 2273433
Email: joan.aarvold@unn.ac.uk

Title of thesis

Close encounters of the fourth kind

Supervisor, department and institution

Director of Studies: Bob Heyman, Professor of Health Social Research, Faculty of Health, Social Work and Education, University of Northumbria at Newcastle

What my thesis is about

To novice adult learners the computer represents an alien universe. Qualitative methods have been used to understand the experience of learning; the pathways travelled and the strategies employed.

How I got into this

As a lecturer responsible for developing and teaching an IT course for health professionals I observed multiple learning pathways (and much headbanging!) in my students. Mack, Lewis and Carroll's work had just been published in Preece & Keller (1990) and this fuelled my research interest. However, I was surprised as I waded into the HCI literature that there was very little reference to established learning theories. Instead, new models (GOMS, CCT, SOAR) were offered which did not answer my questions about how learners made sense of, and felt about, the alien universe. Experiential approaches were noticeably absent.

My contribution to HCI research

The normative theoretical approach to develop user-models, against which subsequent user behaviour is assessed, contains an inbuilt tendency to denigrate users. It also tends to regard 'users' as isolated individuals, learning in a vacuum, rather than social beings who bring a range of skills and experiences to their learning (Gagne 1985). Most HCI research follows an etic approach, and although more user-perspective strategies are being employed, the findings are quantified and tend to reflect task competencies. I was made aware at the recent HCI conference that unless research is about influencing interface design then it is not HCI research. How learners recognise, make sense of, manage and react to problems has informed my design and methods. I am most interested in the H bit of HCI, but feel that I have a contribution to make.

An initial 10 week study, based on normal IT classroom observations and interviews, confirmed my belief that multiple learning pathways operated. Teachers invariably attempted to correct difficulties, usually by the shortest route. Learners, on the other hand, sought understanding of the events confronting them. Their sense of the new 'universe', however, could neither be explained nor understood through observation of whole classes.

I then followed the progress of six novice university students, with different levels of self-expressed anxiety, during their normal introductory computer classes. A final series of video recordings were then made (over several weeks) of three additional novice adult learners as they worked alone (with me as teacher/researcher). The data were analysed using a grounded theory approach (Strauss and Corbin 1990). Other HCI researchers who have used qualitative approaches rarely complete their analyses, acknowledging the complexity and time consuming nature (Lazonder & Van der Meij 1995, Ramsay 1997, Allwood & Kalen 1997).

Contrasting attitudes and expectations between learners were apparent. One explored the new 'universe' with a pioneering spirit, he was not put off by mistakes. He carried on regardless, rarely understanding the alien signposts. His intrepid style helped him and pleased his teacher; however, history reminds us that such explorers often met a sticky end. Another student, on the same course, could not have had a more different approach. He knew his was going to be a difficult journey. He consulted his 'map' at every stage, unfortunately it was seldom helpful. Trainers need to invest time ascertaining learners' feelings and hopes prior to embarking on courses. HCI researchers, according to Downes (1987) have been able to replicate the intellect but not the passions of learners as they interact with computers. The 'feeling' part of learning has been neglected in the mental models research (where is the feeling bit in GOMS?). As with all good relationships, trust plays an important part. Learners' own insecurity during interactions meant that they constantly sought meaningful signposts from the computer.

The use of 'success indicators' for novice users could play a significant part during the steep initial learning phase. All novice participants in my study wanted reassurance that their work 'really had been saved'. Much angst could be saved if a simple message stating 'Your work is now saved' appeared on screen at strategic moments.

Qualitative approaches also demand an equal, trusting relationship between learner and researcher. Without critical questioning at key stages, understanding of events from the learner's view would not have been possible. I could not have known how many different interpretations there are, for example, of the words 'restore' or 'cancel'. The video recordings captured my own responses, thus enabling them to be included in the analysis (rare in HCI). The use of metaphor is not confined to students. My reference to the scroll bar as a kind of 'lift' turned out to provide a 'false signpost' which subsequently misled one student, who later presumed the 'Esc' on a key was short for escalator and pressing it would take her to the top of the page!

What I want to do next

I have come to realise through my work and research interests that there are many occasions (particularly in health and social care fields) where adults have difficulty

continued on page 21 …

My Thesis

Suziah Sulaiman

Name and address of researcher
Suziah Sulaiman
Universiti Teknologi Petronas
31750 Tronoh
Perak, Malaysia
Tel: 605-367 8018
Email: suziah@petronas.com.my

Title of thesis

Heuristics for evaluating the usability of Computer Mediated Communication (CMC) systems

Supervisor, department and institution

Dr Fintan Culwin, School Of Computing, Information Systems and Mathematics, South Bank University, 103 Borough Road, London SE1 0A

What my thesis is about

My thesis attempts to develop and evaluate a set of heuristics that can be used for Computer Mediated Communication (CMC) evaluation.

How I got into this

Initially, my interest was in finding ways to encourage usability evaluation to be taken up early in the production life cycle. As my work progressed, I realised that heuristics provided a cost-effective technique and that little work had been reported on CMC evaluation.

My contribution to HCI research

I identified three goals: to determine how much work has been carried out in CMC evaluation, to investigate how others carried out the activities and what techniques they used, and to propose a cost-effective way to evaluate CMC systems.

Several investigations and a practical study were carried out in order to achieve these goals. A literature survey of conference proceedings, journals, books and articles from the Internet was carried out. The findings indicated that little had been reported on CMC evaluation. Those which involved CMC evaluation applied techniques used in general HCI. Most techniques involved ethnographic approaches and only recently had formal methods been applied.

The HCI evaluation techniques identified were then examined to see how well they could support CMC. This indicated that the techniques could support CMC evaluation provided communicative issues are considered. Even though HCI techniques could be useful, constraints in system development need to be taken into account. It is necessary to find cost-effective techniques which use less evaluation time and so consume minimum project cost, whilst maintaining quality. A literature survey on comparison studies showed that predictive techniques were the most cost-effective and Nielsen's heuristics were chosen for further research.

A practical study was carried out to examine the usefulness of Nielsen's list in supporting CMC evaluation. As Nielsen's list is intended for single-user stand alone purposes, it was predicted that the heuristics could only support the human–computer interaction issues. The evaluators who participated in the investigation were neutral on the overall usefulness of Nielsen's heuristics but agreed that some heuristics are more useful than others. This study resulted in the formulation of a list of requirements for CMC evaluation, in part influenced by Ambe and Monk's (1997) criteria for effective CSCW. This new list formed the basis of a new set of heuristics rephrased in a way similar to Nielsen's. An investigation was carried out to justify the effectiveness of these new heuristics, the findings indicated that the new set of heuristics are more effective than Nielsen's list for CMC evaluation. Even though there are many flaws in the production of the heuristics, it has been demonstrated that this new list could be useful for CMC evaluation.

Acknowledgement

This research was funded by South Bank University, England.

The Seventh IFIP Conference on Human–Computer Interaction

http://www.bcs.org.uk/hci/i99

You are invited to participate in INTERACT '99, a major international conference presenting leading-edge research and development in all aspects of interactive computer systems and technology. This is the last major HCI conference of the millennium. It will review more than thirty years of pioneering work which has shaped the discipline, and highlight the trends, discoveries and innovations which will shape a new era of HCI in the twenty-first century. The venue is Edinburgh, Scotland's historic capital city, currently the centre of a scientific, artistic and political renaissance of wide significance.
The conference takes place during the Edinburgh International Festival, when the city plays host to leading musicians, actors, artists and entertainers from all over the world, as well as to a large number of visitors, who give the city a uniquely festive and cosmopolitan atmosphere.

CONFERENCE TOPICS

Contributions which advance the theory or practice of any aspect of HCI are welcomed. The review policy is to support innovation, insight, and integration in the theories, methodologies, tools and technologies which contribute to HCI, and to support the wider dissemination of effective theory and practice both nationally and internationally. Contributions which bridge several domains, or explore the expansion of HCI into new areas of theory or application, are particularly encouraged.

PAPERS

Deadline for submission: 25 January, 1999.
Additional enquiries to: M. Angela Sasse (University College London) Tel: +44 (0) 171 380 7212 Fax: +44 (0) 171 387 1397
Email: a.sasse@cs.ucl.ac.uk

PANELS

Deadline for submission: 10 May, 1999.
Proposals received by the 25 January 1999 will be reviewed early and, if accepted, will be publicised in the advance programme.
Additional enquiries to: David Benyon (Napier University) Tel: +44 (0) 131 455 5317 Fax: +44 (0) 131 455 4552
Email: d.benyon@dcs.napier.ac.uk

TUTORIALS

Deadline for submission: 25 January, 1999.
Additional enquiries to: Janet Finlay (University of Huddersfield) Tel: +44 (0) 1484 472913 Fax: +44 (0) 1484 421106
Email: j.e.finlay@hud.ac.uk

WORKSHOPS

Deadline for submission: 10 May, 1999.
Proposals received by the 25 January 1999 will be reviewed early and, if accepted, will be publicised in the advance programme.
Additional enquiries to: Alistair Sutcliffe (City University, London) Tel: +44 (0)171 477 8411 Fax: +44 (0) 171 477 8859
Email: a.g.sutcliffe@city.ac.uk

POSTERS

Deadline for submission: 10 May, 1999.
Additional enquiries to: Gilbert Cockton (University of Sunderland) Tel: +44 (0) 191 515 3394/2752 Fax: +44 (0) 191 515 2781
Email: Gilbert.Cockton@sund.ac.uk

DOCTORAL CONSORTIUM

Deadline for submission: 25 January, 1999.
Additional enquiries to: John Karat (IBM TJ Watson Research Center) Tel: +1 914 784 7832 Fax: +1 914 784 6211
Email: jkarat@us.ibm.com
James Alty (Loughborough University) Tel: +44 (0) 1509 222648 Fax: +44 (0) 1509 211586
Email: J.L.Alty@lboro.ac.uk

30 August – 3 September 1999 • Edinburgh, Scotland

VIDEOS
Deadline for submission: 10 May, 1999.
Additional enquiries to: Richard Coyne (University of Edinburgh) Tel: +44 (0) 131 650 2332 Fax: +44 (0) 131 650 8019
Email: Richard.Coyne@ed.ac.uk

INTERACTIVE EXPERIENCE
Deadline for submission: 10 May, 1999.
Additional enquiries to: Steve Brewster (University of Glasgow) Tel: +44 (0) 141 330 4966 Fax: +44 (0) 141 330 4913
Email: stephen@dcs.gla.ac.uk

LABORATORY AND ORGANISATIONAL OVERVIEWS
Deadline for submission: 10 May, 1999.
Additional enquiries to: Rob Procter (University of Edinburgh) Tel: +44 (0) 131 650 5177 Fax: +44 (0) 131 667 7209
Email: rnp@dcs.ed.ac.uk

PROFESSIONAL PRACTICE AND EXPERIENCE
Deadline for submission: 10 May, 1999.
Additional enquiries to: Graham Johnson (NCR) Tel: +44 (0) 1382 598321 Fax: +44 (0) 1382 598313
Email: graham.johnson@dundee.ncr.com

INSTRUCTIONS FOR AUTHORS and FURTHER INFORMATION
A more detailed paper version of the Call for Participation and detailed Instructions for Authors are available from the INTERACT ’99 Secretariat:
INTERACT ’99 Secretariat, Meeting Makers, Jordanhill Campus, 76 Southbrae Drive, Glasgow G13 1PP Tel: +44 (0) 141 553 1930 Fax: +44 (0) 141 552 0511
Email: interact@meetingmakers.co.uk

SPONSORSHIP
Organisations are invited to participate in INTERACT ’99 as corporate sponsors. This is an ideal opportunity to demonstrate the commitment of your organisation to HCI and achieve a marketing profile for your company. Several levels of sponsorship are available.

EXHIBITION
Exhibits are invited from interested commercial organisations, concerned with the design, development, deployment or evaluation of all types of interactive systems, services or devices, or from organisations involved in HCI or human factors research areas and involving such aspects as usability, support tools and techniques, and education. INTERACT ’99 is THE international conference in the area of Human–Computer Interaction. This large and diverse conference provides companies with an excellent opportunity to showcase their HCI-related products and services.
Brochures giving details of the exhibition and/or sponsorship are available from the INTERACT ’99 Secretariat, Meeting Makers, Jordanhill Campus, 76 Southbrae Drive, Glasgow G13 1PP. Tel: +44 (0) 141 553 1930 Fax: +44 (0) 141 552 0511
Email: interact@meetingmakers.co.uk

British HCI Group – Application Form 1998

Please print or type

Contact Details (Give a personal contact when asking for Corporate Membership)

Title ..... First Name ..... Last Name .....

Work Address .....

.....

Tel. ..... Fax. .....

E-mail. .....

Nature of the work you do: .....

Home Address .....

.....

Please send mailings to: my work address ☐; my home address ☐.

Membership Status

Current British HCI Group Membership No. (if applicable) .....

Current British BCS Membership No. (if applicable) .....

Student status (if applicable) .....

Professional Interests (please indicate up to six areas of professional interest)

.....

.....

Data Protection Act

The data on this form will be treated as confidential to the BCS. Names and address may be used, under our strict control, for mailings judged by the British HCI Group Executive to be of value to the membership.

Membership Directory

Do you wish your contact details and professional interests to be listed in the Membership Directory sent to all members of the group? (We will NOT use your home address, unless that is all you have given us.) Yes ☐ No ☐

Getting Involved…

We are always looking for people interested in contributing to HCI group activities by, writing for Inter*faces* magazine, helping run the annual conference or joining the executive. If you are able to contribute in this way or if you have ideas for 1-day meetings or new activities please contact the membership secretary, Ismail Ismail (I.Ismail@cs.ucl.ac.uk; Fax. 0171-387-1397).

Membership Fee

Membership classes and fees for 1998 are:

BCS Member £25 ☐ Non BCS Member £30 ☐ Student £10 ☐ £ .....

Corporate £195 ☐ Corporate membership entitles the organisation to 8 copies of Inter*faces* and other mailings; membership rate for any 4 individuals at British HCI Group events, as well as, a free one-page entry in the membership handbook.

Journal Subscription to 'Interacting with Computers'

The HCI Group manages a journal, *Interacting with Computers*, published quarterly by Elsevier Science. Members may subscribe to this journal at a reduced rate. Vol. 10-No.1 will appear in the Spring of 1998.

Please send me Vol. 10 (1998) of *Interacting with Computers (£50)* £ .....

Please send me Vols. 9 & 10 of *Interacting with Computers (£100)* £ .....

Please send me a free sample issue ☐

Payment

Please enter the total amount for membership and subscriptions £ .....

I enclose a cheque/postal order (in Pounds Sterling only please), made payable to *British HCI Group*

or

Please debit my Access/Visa/Mastercard

Card number Expiry

☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐/☐☐/☐☐

The information provided on this form is to my knowledge correct and I agree to the conditions stated.

Signature: ..... Date: .....

Card holder's name and address if different from above:

.....

.....

.....

Send completed forms and cheques to:

HCI Membership, British Computer Society,
1 Sanford Street, Swindon, SN1 1HJ, UK
(Tel.+44(0)1793 417417)
Queries about membership can also be e-mailed to: hci@bcs.org.uk

diarydiarydiarydiarydiarydiarydiary

1999 International Conference on INTELLIGENT USER INTERFACES
5–8 January 1999, Los Angeles, California. USA
Further Info: URL: http://sigart.acm.org/iui99
Summary: IUI 99 is the annual meeting of the intelligent interfaces community and serves as the principal international forum for reporting outstanding research and development on intelligent user interfaces.

HICSS-32: Collaboration Technology minitrack
5–8 January 1999, Hawaii
Further Info: Murray Turoff; Tel: 201-596-3399; Fax: 201-596-5777; Email: turoff@eies.njit.edu; URL: http://www.cba.hawaii.edu/hicss
Summary: Part of the Collaboration Systems and Technology Track at the Hawaii International Conference on System Sciences (HICSS). This minitrack deals with the theoretical and methodological foundations of research with all forms of collaboration technologies. The focus is on the development, critical evaluation, and validation of theories that guide the design, implementation, and use of collaboration technologies; and various approaches/methodologies adopted to develop, evaluate, and validate these theories.

The Active Web
20 January 1999, Stafford, UK
Further info: Email: activeweb@hiraeth.com; URL: http://www.visualize.uk.com/conf/activeweb
Summary: The web is changing. Its pages are no longer static, but moving, changing, interacting. Even as we watch, it is evolving from an information repository into a distributed interface to a global networked computational engine. But this change has its price. Can users understand what they are seeing? What is the appropriate technology? How do we ensure that the goals and objectives of our highly dynamic and interactive sites are met? This day conference will focus on issues surrounding dynamic interfaces for the Web.

WSCG'99: The Seventh International Conference in Central Europe on Computer Graphics and Visualization 99
8–12 February 1999, Plzen, Czech Republic
Further Info: Vaclav Skala, c/o Computer Science Dept., Univ.of West Bohemia, Univerzitni 8, Box 314, 306 14 Plzen, Czech Republic; Tel: +420-19-7491-188; Fax: +420-19-7491-188; Email: skala@kiv.zcu.cz; URL: http://wscg.zcu.cz
Summary: Event in cooperation with EUROGRAPHICS and IFIP Working Group 5.10 on Computer Graphics and Virtual Worlds

WACC'99: Work Activities Coordination and Collaboration
22–25 February 1999, San Francisco, California, USA
Further Info: Email: wacc99@cs.colorado.edu; URL: http://www.cs.colorado.edu/wacc99
Summary: WACC'99 brings together researchers and practitioners from a variety of disciplines who are addressing or facing issues in work activities coordination and collaboration. Various aspects of this topic have been addressed previously under the separate banners of workflow, software process, groupware, and computer-supported cooperative work.

GW '99 – THE 3rd GESTURE WORKSHOP: 'Towards a Gesture-based Communication in Human–Computer Interaction'
17–19 March 1999, Gif-sur-Yvette, France
Submissions by 12 September (long), 11 December (short)
Further Info: Gesture Workshop '99, LIMSI - CNRS, BP133, F-91403 ORSAY cedex, FRANCE; Email: gw99@limsi.fr; URL: http://www.limsi.fr/GW99/
Summary: GW is an interdisciplinary event for those researching gesture-based communication who want to meet and exchange ideas across disciplines. Under the focus of human–computer communication, the workshop will encompass all aspects of gestural communication.

PAAM99: Fourth International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology
19–21 April 1999, London, UK
Submissions by 11 January 1999
Further Info: PAAM'99 Secretariat, The Practical Application Company, PO Box 137, Blackpool, Lancs FY2 9UN, UK; Tel: +44 (0)1253 358081; Fax: +44 (0)1253 353811; Email: info@pap.com; URL: http://www.demon.co.uk/ar/PAAM99/
Summary: The emphasis of PAAM is unique. It combines the peer-to-peer paper review process of academic conferences with that of the applied industrial mainstream commercial event. The contrast of theory and practice, research and deployment is rarely found elsewhere. PAAM99 will provide a rich blend of tutorials, invited talks, refereed papers, panel discussions, poster session, social agenda and a full industrial exhibition. The result is an ideal forum for the exchange of ideas and knowledge between experts from a broad spectrum of international industries and key technologies.

Joint EUROGRAPHICS – IEEE TCCG Symposium on Visualization
26–28 May 1999, Vienna, Austria
Further Info: Helwig Loeffelmann, Inst. of Computer Graphics, Vienna Univ. of Technology, Austria; Email: helwig@cg.tuwien.ac.at, vissym99@cg.tuwien.ac.at; URL: http://www.cg.tuwien.ac.at/conferences/VisSym99/
Summary: The tenth EUROGRAPHICS workshops on Visualization in Scientific Computing will cover all aspects of computer-based visualization.

UM'99: International Conference on User Modeling
20–24 June 1999, Banff, Canada
Further Info: Judy Kay Basser, Dept of Computer Science, Madsen F09, University of Sydney, AUSTRALIA 2006; Tel: +61-2-9351-4502; Fax: +61-2-9351-3838; URL: http://www.cs.usask.ca/UM99/
Summary: User modeling has been found to enhance the effectiveness and usability of software systems in a wide variety of situations. A user model is an explicit representation of properties of a particular user. A system that constructs and consults user models can adapt diverse aspects of its performance to individual users. Techniques for user modeling have been developed and evaluated by researchers in a number of fields, including artificial intelligence, education, psychology, linguistics, human–computer interaction, and information science. The International Conferences on User Modeling provide a forum in which academic and industrial researchers from all of these fields can exchange their complementary insights on user modeling issues. The size and format of the meetings support intensive discussion, which often continues long after the conference has ended.

PEOPLE IN CONTROL: An International Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres
21–23 June 1999, Bath, UK
Further Info: PIC 99 Secretariat, IEE Conference Services, Savoy Place, London WC2R OBL, UK; Tel: +44(0)171 344 5473/5467; Fax: + 44(0)171 240 8830; Email: PIC99@iee.org.uk; URL: http://www.iee.org.uk/Conf/PIC99
Summary: This new international conference aims to attract industrialists and academics with an interest in how people control complex systems – and in the tools that support them. These people may be working in control rooms for chemical plants, power stations or ships, in aircraft cockpits or motor cars, or in police or military command centres. A key aim of the conference is to bring together practitioners from different application areas so that they can learn from each other's experiences. The conference will have a broad technical scope and include: case studies from the power, chemical, aerospace or transport industries; developments of new display and control equipment; presentations of research results in ergonomics and psychology

HCI International '99: 8th Int. Conference on Human-Computer Interaction
22–27 August 1999, Munich, Germany
Further Info: HCI International '99 – Conference Secretary, Fraunhofer IAO, Nobelstr. 12, 70569 Stuttgart, Germany; Tel: +49 711 970 2331; Fax: +49 711 970 2300; Email: HCI99@iao.fhg.de; URL: http://hci99.iao.fhg.de
Summary: Under the general theme of 'Creating New Relationships', new links and synergies will be explored between information technologies and their users, between people working together, and in the context of the rapidly evolving global information society. The conference will provide an international forum for exchanging and discussing ideas, research results and experiences related to analyzing, designing, developing, applying, and evaluating information and communication technologies for work, leisure, and personal growth.

To receive more information via email on all these events and others, together with full details of many industrial, academic, and research studentship posts, subscribe to our electronic mailing list by sending the following two-line message, filled in appropriately, to the mailbase server:
mailbase@mailbase.ac.uk

join bcs-hci [optional title] <your first name> <your last name>
stop

Evolving the Web for Scientific Knowledge

First Steps Towards an "HCI Knowledge Web"

In this article, I consider the challenge of building a Web-based infrastructure for scholarly research which moves beyond the basic dissemination and linking of documents, to support more powerful searching and analysis of the cumulative knowledge in the literature's documents. Taking the HCI research community as an example, the goal would be to enable HCI researchers to search for interesting documents and phenomena, and discover previously unknown but conceptually related research, for instance, other groups addressing persistent problems in the field, the structure of debates, or when and how new theoretical perspectives began to make an impact. I propose that focusing on the scientific relationships between documents is important, and has advantages as the basis for a Web metadata scheme to enrich the HCI community's Web.

Your desktop, in the not too distant future...

You are starting a new HCI research project, and want to find out what's been done so far. (You can hardly believe it, but 2 years ago, you would have had to search the Web, or one of the few HCI digital libraries, using basic keywords. The servers and search engines knew nothing about how HCI research is conducted and so could provide no assistance. Documents were not described in any machine-readable form other than keywords, and were not linked in any way beyond citations.)

You connect to your local server in the HCI Knowledge Web, and issue queries for the following:

- **documents/websites using or extending the StarViz software system**
- **documents analysing the applied problem of visualizing large datasets in astronomy**
- **documents building on a particular theoretical framework of interest to you, and extending the RouteFinder class of graphing algorithms**
- **documents challenging evidence that the RouteFinder class does not scale up**
- **documents problematising a methodology closely associated with StarViz**

It's been said many times in recent years, and it's still true:

> *The Net, particularly the Web, provides an unprecedented opportunity in scientific history to locate, interconnect and analyse ideas and documents.*

But…

> *The Web is becoming a more chaotic place by the day. As the signal to noise ratio gets worse, research communities need better support for tracking developments and finding relevant documents.*

It is currently impossible for search engines (Web or otherwise) to answer complex questions commonly posed by researchers, such as the following:

- are there distinct schools of thought in this field?
- what impact did this evidence have?
- who is currently tackling this applied problem?
- has anyone built a system based on this theory?
- has anyone applied this theory to other fields?

The reason these questions are impossible to answer at present is that there must be a way to abstract meaningful patterns of documents. Thus, in relation to the first question above, we need to ask: what is a 'school of thought', how might it manifest itself in the literature, and are there corresponding patterns that could be detectable by a software agent to present to researchers as potentially significant? It is possible that useful information may be extracted through intelligent analyses of texts, but often this information is not explicit in documents, but implicit in the minds of domain experts. *Metadata* (introduced shortly) is an alternative way to provide such information (if experts encode it), but there is rarely metadata about scientific documents expressed using the conceptual language of that field. The Web now makes this technically possible – thousands of users can now contribute structured information to a shared, searchable repository.

But is a lingua franca (using metadata or otherwise) possible for a scientific research community, HCI in particular, what would constitute a good scheme, and what social and technical systems would need to co-evolve to make it sustainable? Let's begin by briefly considering what the HCI community has available to it today.

Today's HCI digital library

What is the state of the 'HCI digital library' accessible over the net today? Within the HCI community, the pioneering work of Gary Perlman's *HCI Bibliography Project (HCI-Bib)* has made thousands of abstracts (some linked to other digital libraries) downloadable and searchable over the Web <www.hcibib.org>. Professional and learned societies such as the ACM <www.acm.org/dl> and IEEE <computer.org/epub> are creating digital libraries providing subscriber access to many HCI-relevant journals and conferences (perhaps we can expect a BCS digital library soon?…). Most

Simon Buckingham Shum

scientific publishers are now providing subscriber access to digital copies of journals. There is work on automatically linking citations to abstracts, although these depend on inter-publisher agreements (see sidebar). And of course, many workshops and individuals provide access to full papers.

Preprint servers provide repositories of technical reports, and, if widely used within a community, are perhaps the best way to track new work (although unreviewed). The Los Alamos National Laboratory (LANL) preprint server <xxx.lanl.gov> set up by Paul Ginsparg, initially to serve the high energy physics community, has become the first place to publish new technical reports in that field (which are then replaced by the final versions when published – journals have been sidelined in this respect). Recently, a Computing Research Repository (CoRR) has been added to the LANL preprint server <xxx.lanl.gov/archive/cs/intro.html>, including an HCI subject area moderated by Terry Winograd. Preprint servers allow you to define interests using keywords, after which the server sends email alerts whenever new material is added (all of the above servers provide these). It will be interesting to see if the CoRR server achieves the same uptake as within the physics community (it may be that LANL's success derives from the premium on being the first to publish results in physics, arguably much higher than in computer science, or HCI). There used to be an HCI server at the The London & South-East Centre for High Performance Computing (SEL-HPC) but this appears to have ended (<www.lpac.ac.uk/SEL-HPC/> no longer works).

These resources are a welcome alternative to having to order and wait for paper documents. But they are just a start. Overwhelmingly, the Web as a resource for scientific knowledge is still serving as a searchable paper-publication resource, plus simple linkage. The Web's success reflects the power and attractiveness of this simple model, but it is an 'entry level hypertext system' in comparison to the power of a rich hypertext which exploits machine-processable node and link semantics. Such semantic hypertexts were implemented in early research hypertext prototypes dating back to the mid-1980s, based on cognitive science's concept of the semantic networks.

The key challenge for any effort to create a better system is deciding on the representational scheme to use (what should be the schema determining the node and link types?) and the usability of the system (how much effort is required to encode information using this scheme, and to subsequently interpret the system?). Is there a way to negotiate these inevitable overheads, in order to begin reaping the benefits of a more powerful system? I propose that metadata could be used, but in a novel way that differs from current metadata schemes.

Scholarly publishers in the brave new world

The ideal of freely accessible information to all, whilst now practical at a technical and usability level with the arrival of the Web, is of course dogged by copyright restrictions. This is not the place to go into detail on this fraught topic, but suffice to note that electronic publishing has the potential to change the rules that bound researchers to publishers when they were wholly dependent on paper for dissemination. The Net provides the basis for scholars to forge new relationships with publishers, who may have to find new roles (some radical proposals and debate on this topic can be found in <cogsci.ecs.soton.ac.uk/~harnad/subvert.html>). Nor is the simplistic equation that *the Net = unreviewed, low quality material* sustainable. It is peer review and other forms of quality control that add value and reliability, not the paper medium per se. Electronic journals are showing how the Web is well suited to scholarly publishing and peer review (e.g. <www-jime.open.ac.uk>).

Metadata schemes

Use of *metadata schemes* is one way to make the Web a semantically enriched hypertext. On the right is some imaginary metadata for a document, using <angle brackets> to delimit each metadata field.

Note that some of the metadata tags simply describe the *content* of the document, whilst the last two actually describe particular kinds of *relationships* to others (eg. this document is not a PREREQUISITE-FOR Q777, it BUILDS-ON it). A search engine providing a query form with fields for these tags enables users to search specifically for documents which have 'community of practice' as a CORE-CONCEPT, and are PREREQUISITE-FOR 'B888'.

We are seeing the emergence of W3C's XML scheme (a stripped down version of SGML) for adding one's own tags to text, initial work on the Resource Description Framework (RDF, for managing multiple metadata schemes), and internationally coordinated initiatives such as the Dublin

<TITLE=Unit 11: Knowledge Management Technologies>

<COURSE=B823>

<PRESENTATION=Nov1999>

<INSTITUTION=Open University UK>

<AUTHOR=Simon Buckingham Shum>

<CORE-CONCEPTS=knowledge, information, representations, interpretation, technology, community of practice>

<BUILDS-ON=Q777, B823-Unit 2>

<PREREQUISITE-FOR=B888>

Evolving the Web for Scientific Knowledge

Core metadata scheme, which provides a basis for communities to use or, if desired, specialise their general scheme <purl.oclc.org/metadata/dublin_core>. Coupled with toolkits such as ROADS for structuring information gateways <www.ukoln.ac.uk/roads>, and protocols such as Z39.50 <lcweb.loc.gov/z3950/agency> for distributed searching of servers, we have the emerging basis for more powerful infrastructures for content discovery.

Not surprisingly, given their interest in classification and their already large document repositories, the library and information sciences are leaders and early adopters of metadata. But other research fields are initiating consortiums for resource description, e.g. the Instructional Management System for online educational resources <http://www.imsproject.com>.

Metadata focused on scientific relationships

It is striking to note that in most metadata schemes, *relational* information (how does this document relate to others?) tends to be the poor cousin of *content* information (what's in this document?). This may be because much of the work to date has been driven by library/information scientists. However, *relationships* are critical for researchers, who invest a lot of energy in articulating and debating different claims about the significance of conceptual structures. Moreover, it is often precisely the issue of how to describe the status of a document or idea that is under debate in research – an approach that allows only one way to encode material will fail to meet the needs of a community which is constantly contesting claims.

A principle from hypertext research is to avoid loading *nodes* (e.g. web documents) with semantic information (e.g. metadata encoding), and focus instead on the *links*. That way, a given node remains 'neutral' on its own, but can be referred to in many different ways by different authors; it is its place in the network which determines its role and interpretation. It may be, therefore, that the generic *relation* field in a Web metadata scheme such as Dublin Core could provide the anchor that researchers need to specialise into a set reflecting the important relationships in their field.

A metadata scheme grounded in concepts and relationships for scholarly discourse would, for instance, provide a way to 'semantically tag' keywords and references. Consider, for example, how you choose keywords for your papers. They typically reflect many different conceptual relationships. Instead of an undifferentiated list, very different keywords such as 'Java' and 'Situated Cognition' could be tagged to indicate (to a software agent) that they refer to *software* and a *theory/framework*, respectively. Another example: both human and software agents would be interested to know that a citation to 'Smith, 1998' is not evidence of its reliability (the implicit interpretation of science citation indices), but that it is *problematising* the *method* used in that paper.

I therefore propose that a more tractable goal is a scheme which reflects *the WAY in which HCI research discourse proceeds* as a discipline, focusing not on encoding the content of documents (other techniques exist for doing this automatically), but on the *scientific relationships between documents* that are hard, if not impossible, to infer automatically. Consider familiar relationships between papers in the literature such as *modifies, describes, supports, problematises.* These verbs are commonly used in conjunction with concepts such as *applied problem, theory/framework, software, evaluation, trends.* I suggest that these are relatively uncontentious and stable – they are how we think about documents and their inter-relationships.

A scheme based on accepted scientific relationships is less brittle than classification schemes which seek to reflect key subject matter in the field, but which require regular updating (cf. the ACM 1998 Computing Classification Scheme for keywords <www.acm.org/class/1998>).

So, what might a scholarly metadata scheme – reflecting the *modes of discourse* common in HCI, not a master classification scheme – look like?

A possible HCI metadata scheme

Consider the form on the right, which provides a way to construct common relationships between research documents – could you describe one of your publications using this scheme? It usually takes a little effort (perhaps productive) to distill the key contributions of a document, but informal testing has shown that most can be described using the constructs offered. Some examples follow:

Examples

Given the building blocks of the above metadata scheme, here are some fragments of metadata description to show its application.

ANALYSES APPLIED-PROBLEM Air traffic controller cognitive overhead
REF: Smith, J. (1997) ATC Overload. Journal of ATC, 3 (4), 100–150

USES/APPLIES THEORY/FRAMEWORK Situated Cognition, Activity Theory

DESCRIBES-NEW EVIDENCE use of video, undergraduate university physics, student ability

PROBLEMATISES SOFTWARE GOMS cognitive modelling tools

MODIFIES/EXTENDS LANGUAGE Knowledge Interchange Format (KIF)

CHARACTERISES/RECASTS TREND Electronic trading over the internet
REF:
REF:
REF:

CHALLENGES SCHOOL-OF-THOUGHT Postmodernism
REF:
REF:
REF:

SUPPORTS EVIDENCE multimedia, school chemistry teaching

Simon Buckingham Shum

TITLE:
AUTHOR:
CITATION:
URL:
ABSTRACT:
KEYWORDS:

Key Contributions, and Relations to other work

From the following table, copy and paste a **RELATION**, add a **CONCEPT**, a **Description** of the Concept, and any **References/URLs** for the Concept**.**

Repeat until you are satisfied that you have summarised the document's key content and relationships to the existing literature.

RELATION	**CONCEPT**	**Description**	**Reference/URL**
Most Relations pair meaningfully with most Concepts. The next table summarises legitimate pairings.		*Name/keywords for the* **CONCEPT**	… for the **CONCEPT**
ANALYSES	**APPLIED-PROBLEM**	…	…
SOLVES	**THEORETICAL-PROBLEM**	…	…
DESCRIBES-NEW	**METHOD**	…	…
USES/APPLIES	**LANGUAGE**	…	…
MODIFIES/EXTENDS	**SOFTWARE**	…	…
CHARACTERISES/RECASTS	**EVIDENCE**	…	…
EVALUATES	**THEORY/FRAMEWORK**		
or more specifically	**TREND**		
SUPPORTS	**SCHOOL-OF-THOUGHT**		
PROBLEMATISES			
CHALLENGES			

Most of the Relations can be sensibly combined with any of the Concepts, but the table below shows nonsensical combinations (dark cells).

CONCEPT / **RELATION:**	PROBLEM	THEORY/ FRAMEWORK	LANGUAGE	SOFTWARE	METHOD	EVIDENCE	TREND	SCHOOL-OF-THOUGHT
ANALYSES								
SOLVES								
DESCRIBES-NEW								
USES/APPLIES								
MODIFIES/EXTENDS								
CHARACTERISES/ RECASTS								
EVALUATES								

Such fragments can be built into more complex structures. Returning to our earlier question about schools of thought, we might define a 'structural signature' in the document web which we would find interesting. We could therefore define and search for patterns in the literature (of encoded documents) which suggested the emergence of distinctive perspectives through a structural signature (perhaps graphically constructed) expressing the following: a 'school of thought' is a perspective, in contrast to at least one other, on a common phenomenon. A perspective can be recognised by the common **THEORY/FRAMEWORKS** on which a group of researchers draws (size=N?), the associated **METHODS** and **LANGUAGES** which they deploy, and the body of **EVIDENCE** that they mutually support. Conversely, the set of **THEORY/FRAMEWORKS**, **METHODS**, **LANGUAGES** and **EVIDENCE** that they collectively **CHALLENGE** or **PROBLEMATISE** may represent a different perspective.

Within HCI, we might recognise several examples of perspectives that seem to fall into distinctive 'camps', building as they do on very different conceptual foundations. Consider ethnographic/sociological approaches as opposed to information processing approaches to studying the workplace. Or situated cognition and learning 'versus' symbolic AI perspectives on interaction. Or discount usability as opposed to cognitive modelling techniques. In summary, a school of thought may be declared by someone as shorthand in their description of their document, but such phenomena might also be detected as an emergent pattern within the literature.

Making it work: technical and social processes

As the Olde Englishe proverb goes, 'a metadata scheme alone doth not a knowledge web make' – even if it does provide a successful lingua franca. There's no denying that many issues remain, which we are currently seeking resources to investigate. Interesting challenges that would quickly emerge if this initiative took off include:

User interfaces: for assisting in the construction of hypertext and metadata-based queries, interest profiles and agents, and the display of search results involving potentially large document sets and complex inter-relationships.

Managing terminological variations: the subtleties of language are important to researchers in expressing their ideas so the *relational types* in the metadata scheme need to be acceptable and document *content* indexing needs to cope with terminological variations. 'Bottom-up' information analysis techniques for analysing text corpuses (such as latent semantic indexing), and thesauri for synonym matching need to be used in synergy with 'top-down' metadata (which provides the valuable relational knowledge of researchers).

Supporting emergent structures: a particular claim, or a network of ideas, may be asserted by one author, but what do others say? Mechanisms need to be worked out to enable detection of structural patterns that are widely subscribed to, and which therefore may be more credible than a 'lone voice' (e.g. a theory/framework which provides the motivation for subsequent work; a software system or design method that is widely used and extended). Authority naturally rests with more established researchers, so one would expect a facility to define filters and interest profiles which prioritise perspectives from particular people, research groups or institutions.

A lesson and guiding principle from the HCI Bibliography project is that any large scale, community-centred initiative must be realisable through a *little effort* from a *lot of people.* In scientific research, it is the *authors* who have the interest in maintaining the 'electronic visibility' of their work, to ensure accessibility and, hopefully, impact on the field. If individual HCI researchers take responsibility for enriching the descriptions of their own publications – the key content, and its linkage to other work – then a spectrum of new possibilities opens up, and we have an answer to the challenge of encoding documents. It is realistic to envisage connecting to a web server, completing a form based on a metadata scheme, and submitting it to the repository, which is mirrored around the world. Again, this is already standard practice in certain communities.

Invitation to participate in a pilot study

We can dream about the possibilities, but the first step is to pilot a metadata scheme to describe HCI publications, seeking the right balance of simplicity and expressiveness.

I therefore invite you to 'beta test' the above metadata scheme by using it to describe just one or two of your own HCI publications (initial design iterations have already been done). Please send me your form(s) and your feedback. The forms will undergo a preliminary analysis to assess the scheme's usability and the potential power of the information it generates. The more participants we have, the larger our testbed dataset for testing serverside tools and demonstration services such as alerting and visualisations (cf. interesting work by Chaomei Chen who has automatically generated VRML maps of ACM CHI and Hypertext proceedings <www.brunel.ac.uk/~cssrccc2>). We hope to combine such techniques with the approach described here, starting with the metadata from the pilot study.

This article, the metadata form and example metadata descriptions for a couple of HCI publications are on the *HCI Knowledge Web* pilot site at:

kmi.open.ac.uk/sbs/hciweb/pilot.html

From there, download the form as an RTF or HTML document and import into your wordprocessor:

kmi.open.ac.uk/sbs/hciweb/HCI-Pilot-Metadata.rtf

kmi.open.ac.uk/sbs/hciweb/HCI-Pilot-Metadata.html

Predicting the Web's evolution is a tricky business. But there's one thing we can be sure about: no-one will – or *can* – do it for the HCI community, but the HCI community.

Simon Buckingham Shum
Knowledge Media Institute, Open University, Milton Keynes, MK7 6AA, UK
Email: S.Buckingham.Shum@open.ac.uk
WWW: kmi.open.ac.uk/sbs

My Thesis *... continued from page 10*

learning new skills and adopting different lifestyles. Interacting with computers requires new skills and a suspension of previously held beliefs. Perhaps I am being too enthusiastic (or naïve), but I see increased potential for HCI research from what Hammersley (1997) refers to as 'methodological eclecticism' and approaches which become (more) social (Carroll 1996).

References

Carroll, J. M. (1996). Becoming social: expanding scenario-based approaches in HCI. *Behaviour & Information Technology*, **15**:4, 266–275.

Gagne, R. M.(1985). *The conditions of learning*, 4th edition. New York: Holt, Rinehart & Winston.

Hammersley, M. (1997). The relationship between qualitative and quantitative research: paradigm loyalty versus methodological eclecticism. In *Handbook of qualitative research methods*, J. T. E. Richardson (ed.). Leicester: The British Psychological Society.

Preece, J. and P. Keller (1990). *Human-Computer Interaction*. Open University Press.

Strauss, A. and J. Corbin (1990). *Basics of qualitative research. Grounded theory procedures and techniques*. London: Sage Publications.

Lazonder, A. W. and H. van der Meij (1995). Error-information in tutorial documentation: supporting users' errors to facilitate initial skill learning. *International Jnl of Human-Computer Studies*, **42**, 185–206.

Ramsay, J., H-J. Popp, B. Thull and G. Rau (1997). The evaluation of an information system for intensive care. *Behaviour & Information Technology*, **16**:1, 17–24.

Downes, S. (1987). Human-computer interaction: a critical synthesis. *Social Epistemology*, **1**:1, 27–36.

Allwood, C. M. and T. Kalen (1997). Evaluating and Improving the usability of a user manual. *Behaviour & Information Technology*, **16**:1, 43–57

Announcement and Call for Papers

A one-day workshop on workspace models for collaboration

12th April 1999

Hosted by the
Department of Computer Science
Queen Mary & Westfield College
University of London

Workshop Aims

The aim of this workshop is to gather experience of designing, using and evaluating models of workspaces for collaboration – in particular, the ability of these models to support users' differing and changing needs. The workshop will consist of talks by and discussion with invited speakers, and debate and discussion about the issues raised.

Workshop Topics

The following list gives an outline of topics on the theme of adaptable workspace models, and is not intended to be exhaustive:

- Surveys, characterisations and critical appraisal of existing workspace models, including their modes of adaptation and configuration for different groups and tasks
- Requirements studies: task requirements, group requirements
- Identifying design features and dimensions
- Relationships to theories of work-group dynamics and social theories
- Case studies and evaluations
- User interface design

Format and Attendance

The workshop will consist of talks by invited speakers (to be confirmed), and debates and discussion based on topics drawn from the submitted position papers. Full details will appear on the workshop WWW page: http://www.dcs.qmw.ac.uk/research/distrib/Mushroom/workshop/
The workshop will be limited to 50 attendees, and participation is by invitation only. Those interested in attending should submit a position paper of up to four pages, giving an outline of ideas that will stimulate debate and discussion on the workshop topic. Publication will be sought if there are enough high quality position papers, and if the selected authors are willing to expand them into full papers.

Important Dates

Position papers due	7th January 1999
Notification of acceptance	11th February 1999
Registration deadline	12th March 1999
Workshop	12 April 1999

Registration and further information

Registration details will appear on the workshop web page.
Contact: Tim Kindberg <timk@dcs.qmw.ac.uk>

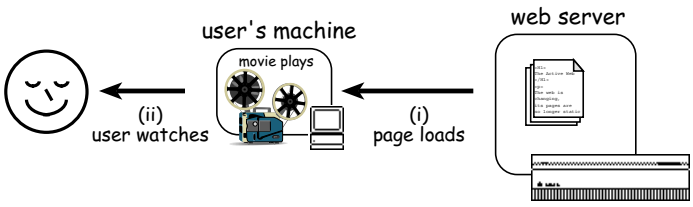
The Active Web

Part II

This is the second part of a two part article. In Interfaces *38 we looked at the issues affecting the choice of an appropriate web technology and at the use of basic animation and media. In this part we'll examine several scenarios for adding interactive elements to web pages and for generating and updating web pages from databases.*
keywords: *world-wide web, interaction, design, HCI, CSCW, Java, JavaScript, JDBC, CGI, servlets*

Where we got to last time

In Part I of this article we briefly considered the plethora of technologies available now on the web and some of the critical issues for active web design. From the external, user's viewpoint we need to ask *what* is changing: media, presentation or actual data; by *whom*: by the computer automatically , by the author, by the end-user or by another user; and *how often*, the *pace* of change: seconds, days or months? From a technical standpoint we need to know w*here* 'computation' is happening: in the user's web-browsing client, in the server, in some other machine or in the human system surrounding it. The 'what happens where' question is the heart of architectural design. It has a major impact on the pace of interaction, both f*eedback*, how fast users see the effects of their own actions, and *feedthrough*, how fast they see the effects of others' actions. Also, where computation happens influences where data has to be moved to with corresponding effects on download times and on the security of the data.



Animated gif or movie

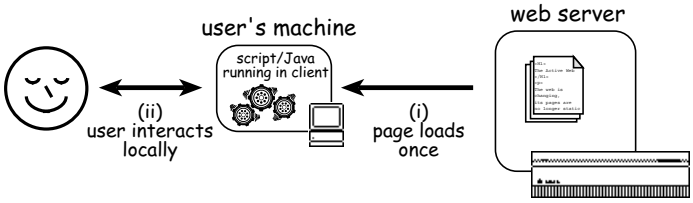
In the first part we looked at the simplest form of active web page, those with movies, animated gifs or streaming video. These are simplest, not in the sense that no effort is required – a short video clip may require many days of production effort – but in the sense that they have least user interaction. In this part we'll look at more complex forms of interaction. First, where the actual content is fixed, but the user can change the form of presentation, secondly at the generation of pages from database content, and finally at the update of database information through the web.

Although the last of these could be considered 'collaborative' at a fairly minimal level, putting real CSCW applications on the web adds an extra level of complexity. I don't have space to cover this here, but if you are interested see the proceedings of the 1996 ERCIM workshop on "*CSCW and the Web*", last year's CSCW Journal special issue on the subject (reprinted as a Kluwer book) and my own pages on web architecture [2, 1, 4]

Fixed content – local interaction and changing views

Probably the most hyped aspect of the web in recent years has been Java. In fact, Java can be used to write server-end software and platform-independent standalone programs (not to mention the embedded systems for which it was originally designed!), but the aspect that most people think of is Java applets.

In fact, applets are just one of the techniques that can be added to give client-end interaction (and about the least well integrated into the rest of the page). The most common alternatives are JavaScript and VBScript (which are also the power behind Dynamic HTML), but vendor-specific solutions such as Shockwave are also heavily used, and in research systems client-end Tcl/Tk plug-ins are available [7]. These techniques share the characteristic that they are downloaded to the user's own machine and thereafter all interaction happens on the PC, not across the network (with caveats – see below).



Java applet or JavaScript running locally

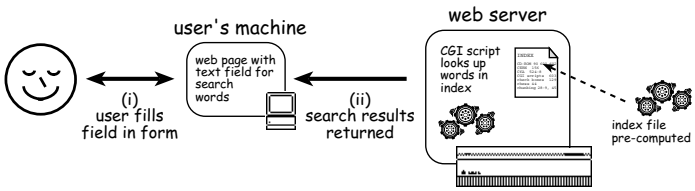
The simplest use of this is to add interaction widgets such as roll-over buttons (usually using JavaScript). More complex pages may add the equivalent of an interactive application on the page. For examples, see my own pages on coin-tossing experiments [12], which use JavaScript to emulate real and biased coins, and also my pages on interactive stacking histograms [13], which use a Java applet. See Sun's own sites for many more examples. The addition of DHTML gives even more opportunities for dynamic pages where parts of the page can move, change size, or change content all without any interaction with the web-server.

Notice how this local interaction confuses the static model of the web. What should happen when you go back to a previously visited page, or reload it? Do you get the original state or the last state of your interaction? What happens if you launch a second window on the same page? The actual behaviour tends to be browser specific and not always what you would expect! In particular, some browsers do not re-initialise applets on a reload and so if you edit the applet's parameters and then reload you may not see the effects of your changes. More of a problem for web developers than end-users, but very confusing.

Some user-driven interaction can be accommodated at the client-end, but not all. Consider search engines. It would be foolish to download several megabytes of information so that a Java applet can search it locally! Instead all common

Alan Dix

web search pages work by submitting forms to the server where CGI programs perform the searches and return results. An additional reason for this approach is that most browsers support forms, but many still do not support Java or scripting in a consistent manner. The search engine for our *Human–Computer Interaction* textbook web pages works in this way. The user's keywords are submitted to the server using an HTML form, compared against pre-prepared indexes at the server, and all matching paragraphs in the book are returned [10]. This also reminds us of another reason for not downloading all the text to the user's machine: security – we don't want to distribute the full electronic text for free!



HCI book search [10]

Notice that, in all the above, the underlying content does not change, the variable factor is the user's input. The computation (working out what to show) needs both the data supplied by the web author (pages, databases, indexes, etc.) and the user's input. The result must end up on the user's screen. Either the data must come to the user's machine (as in my dancing histograms where the histogram data is in applet parameters), or the user's input must go to the server (as with the search). We can see from the examples that the choice between these depends on the required pace of interaction, the size of the dataset required, security and available technology.

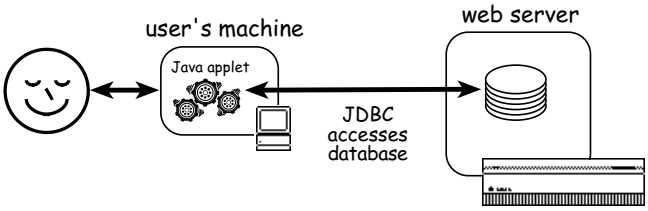
Automatic generation

It was evident in the earliest days of the web that a key problem for the future would be maintenance. In the first rush of enthusiasm individuals and organisations produced extensive and fascinating sites built largely of hand-crafted HTML. Not surprisingly, vast areas of the web are not just static but in perpetual stasis. Web surfing sometimes seems not so much a water-sport, but an exercise in archaeology.

From the beginning it was clear that web sites would eventually need to be created from databases of content combined with some form of template or layout description. However, at that stage there were no tools available and those who saw the database future used a variety of ad hoc methods. Some of my own earliest web material was automatically created from HyperCard stacks or tagged text files and I have supervised projects generating pages using Access, Visual Basic and C. Indeed, these ad hoc approaches are still more flexible and often easier than the vendor-specific solutions.

Happily there are now a (sometimes bewildering) array of products for automating web production from existing and bespoke databases. These include vendor-specific products such as Oracle Web Server, Domino (for publishing Lotus Notes), Cold Fusion and Microsoft's Visual Interdev/ASP; and also more general techniques such as using SQL, ODBC or JDBC to access databases from CGI scripts or even from running Java applets.

There are many advantages of database-generated web sites. They make use of existing data sources. They guarantee consistency of different views of the data within the site and between the site and the corporate data. They allow easy generation of tables of contents, indices, and inter-page links. They separate content and layout. Of course, this separation of content and presentation has been an issue in user-interface architecture for many years, being the driving force behind the Seeheim model, MVC, PAC and the Arch–Slinky model [6, 5, 3, 9]. It is also an issue the web community is embracing with the development of CSS and XML.



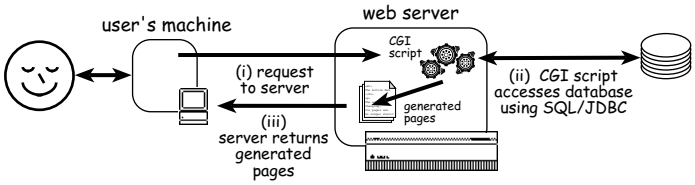
Java applet accesses database using JDBC

Probably the most high-tech way to get database content on the web is to access a database directly from a running applet. The interface can then have any look-and-feel that can be programmed in Java and can allow very rapid interaction with the user. The Java applet can establish an Internet connection back to the web-server to access data files using HTTP (as if they were web pages), can connect to a bespoke server (e.g. for chat-type applications) or can use the standard database access methods in JDBC. Using JDBC the applet can issue complex SQL queries back to the database meaning that some of the most complicated work happens there.

In all cases, the Java security model built into most web browsers means that the applet can only connect back to the machine from which it came. This means that the database server must run on the same machine as the web-server. Think about this. The most insecure part of any system is usually the web-server, both because it is easy to leave loopholes in the many file access permissions and also because it often sits outside the most secure part of a corporate firewall.

The more common solution is where the user uses a web forms interface (or special URL) and then a CGI script runs at the server end accessing the database. The CGI script generates a web page which is then returned to the user. Some of the vendor-specific solutions use essentially this approach but bypass the web-server/CGI step by having their own special web-server which accesses the database

The Active Web

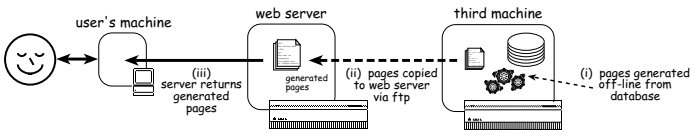


CGI script accesses database

directly using their own scripting language or templates. Similar effects can be obtained using other server technology, such as Microsoft's ASP, Java servlets, and some forms of server-side includes.

The user interface of such systems is limited to standard HTML features. This is a limitation, but is at least consistent and means that they will work with virtually any browser. Java applets can offer more rapid surface interaction, but both have to wait for the actual data to move between server and client. Of course, the pages generated by a CGI script can themselves contain JavaScript or Java applets for local interaction, so the difference between the two solutions is not so radical as first appears.

From a security angle the database accessed from the CGI script can run on a separate machine (using standard database remote access methods or even a Java/JDBC CGI program), thus making the system more secure. However, the database cannot be entirely secure – if the web-server machine is compromised the CGI scripts can be altered to corrupt or modify the database! The special vendor-specific web-servers are probably more secure as they don't require a standard web-server to be running.



Batch pre-generation of web pages

Finally, we have what at first sight appears to be the low-tech solution, the generation of pages off-line from a database. My own early HyperCard solutions were of this form as is HyperAT (8).

This is certainly the simplest solution as it separates out the task of page generation from that of page delivery. Pages can be generated using many standard database packages including PC-based databases such as Access, or HyperCard, or using programs accessing a database such as Visual Basic, Java or C. These can run on a central computer, or on your own PC. The generating program simply produces a set of HTML pages on your own disk which can be checked locally and then copied onto the web-server using ftp or shared network disks. Many people think that this will be difficult, but in fact it is remarkably easy as you can use the tools you are used to – if you can create a text file you can create HTML. In fact, the snippet of Visual Basic below is a fully functioning HTML generator!

```
Set db = openDatabase("C:\test.mdb");
sql = "select Name, Address from Personnel;"
Set query = db.OpenRecordset(sql)
Open "out.html" For Output As #1

Print #1, "<h1>Address List</h1>"
query.MoveFirst
While Not query.EOF
  Print #1, "<p>" & query("Name") & "  "&
                    query("Address")
  query.MoveNext
Wend

Close #1
query.Close
```

Visual Basic code to generate a web page

As well as the ease of programming, the off-line generation of web pages means that there is no need for an on-line connection between the web-server and the database, so a breach in the security of the web-server doesn't compromise your database. In addition, it may mean that the web-server can be configured without CGI scripting enabled at all, which considerably increases its security. Another benefit is that even with *very* high hit rates the database engines are not overloaded, which results in better performance.

The downside is that you can only show the indices and pages that you can pre-compute. So, you could use a product database to produce a pro-forma page for each stock item, as well as alphabetic and categorised lists, but could not produce a special list based on a user's own search criteria.

As you can probably tell, this low-tech solution is my favoured one in many circumstances – whenever the pace of change is low (e.g. overnight or periodic updates acceptable), the volume of data not too large and no on-line searching is required. Even when some of these conditions don't hold it is possible to use the general approach. For example, searching can often be arranged by having a much cut-down database or index on the web-server with most pages pre-computed. Similarly, on our Intranet in the School of Computing at Staffordshire, we have timetables organised by module, year group, and member of staff, all drawn from a central Oracle database. Whenever the timetable database is updated (via a non-web-based Oracle form) the affected pages are automatically regenerated and file-transferred to the web-server. This means that the timetables are constantly up-to-date, but that the Oracle database is totally secure.

Dynamic content

The above mechanisms manage the feedthrough when the database is updated by some non-web means. Perhaps the

most 'active' web pages are those where the content of the pages reacts to and is updateable by web users.

If pages are generated from database content using either the Java-applet/JDBC method or the CGI method, the same mechanisms can as easily be used to update as to access the database. The feedback of changes to the user is thus effectively instantaneous – you check for seat availability on the theatre web page, select a seat, enter your credit card details and not only is the seat booked, but you can see it change from free to booked on the web page.

If instead we consider an estate agent's web page, with houses for sale and potential buyers, the situation is rather different. The pace of change is slow, house purchases take place over days, weeks and months. A solution which automatically marked a house as sold would neither be necessary nor desirable! In this case a socio-technical solution using low-tech database generation would likely be sufficient. The web page can have contact telephone number, email address or message form. These (as well as non-web-based queries) come to the estate agent who then makes the decision as to when to mark the house 'sold'. There is a continuous loop between web user and database, but it involves human as well as automatic processes.

The new British HCI Group consultancy and HCI course web pages [11] are slightly more automated, but embody a similar socio-technical style of solution. Web users can fill out a form with details of their courses or consultancy services. This form is processed by a CGI script which generates an HTML page describing the course/service, but does *not* automatically link this in to the listing pages. Instead, a copy of the details is sent to moderators for the pages who when they are satisfied inform the web administrator who adds the link.

Are you interested?

Are you working on active web technology either as a researcher or practitioner? If so you may be interested in a workshop on "The Active Web" to be held on 20th January 1999. You can register on the web at:

`http://www.visualize.uk.com/conf/activeweb/registration/`

Acknowledgements

Thanks to Dave Clarke for his helpful comments.

Bibliography

Web research and comment:

1. Bentley, R., U. Busbach, D. Kerr and K. Sikkel, eds (1997). *Groupware and the World Wide Web*, Kluwer, Dordrecht. (reprinted from *Computer–Supported Cooperative Work: The Journal of Collaborative Computing* (**6**)) http://www.wkap.nl/book.htm/0-7923-4534-7
2. Busbach, U., D. Kerr and K. Sikkel, eds (1996). *CSCW and the Web - Proceedings of the 5th ERCIM/W4G Workshop. Arbeitspapiere der GMD 984,*. GMD/FIT, Sankt Augustin, Germany. http://orgwis.gmd.de/projects/W4G/proceedings/
3. Coutaz, J. (1987). PAC, An Object Oriented Model For Dialog Design. In Human-Computer Interaction - INTERACT '87, H.J. Bullinger and B. Shackel (eds), 431–436.
4. Dix, A. (1998) Web Architecture (research topics page). http://www.hiraeth.com/alan/topics/webarch/
5. Lewis (1995). *The Art and Science of Smalltalk*. Prentice Hall.
6. Pfaff, G. and P.J.W. Hagen, eds (1985). *Seeheim Workshop on User Interface Management Systems*. Berlin: Springer-Verlag.
7. Reiff, F. (1998). *Migrating from Character-based Interfaces to Web-based Interfaces*. Computing Department, Lancaster University. http://www.comp.lancs.ac.uk/computing/research/cseg/projects/Intranet/FormstoWeb.pdf
8. Theng, Y. L. and H. Thimbleby (1998). Practical Authoring Tools for Web Authoring. *Interfaces*, **38** (Summer 1998): 8–10
9. UIMS (1992). The UIMS tool developers workshop: A metamodel for the runtime architecture of an interactive system. In SIGCHI Bulletin, 24(1), 32–37
10. van Welie, M. and A. Eliëns (1996). Chatting on the Web. in [2], 73–80.

Web examples:

11. British HCI Group course and consultancy pages (CGI update): http://www.hiraeth.com/hci-group/courses/ http://www.hiraeth.com/hci-group/consult.html
12. coin tossing (JavaScript): http://www.hiraeth.com/alan/tutorials/stats/coinrace.html
13. dancing histograms (Java): http://www.hiraeth.com/alan/topics/vis/hist.html
14. free-text search pages for *Human–Computer Interaction 2e*, Dix, Finlay Abowd and Beale, Prentice Hall, 1998 (CGI search): http://www.hiraeth.com/books/hci/search.html
15. School of Computing and Mathematics, University of Huddersfield, Short Course Programme (batch HyperCard generation) http://www.hud.ac.uk/schools/comp+maths/short-courses/
16. School of Computing Science, Middlesex University, StyleGeezer, http://www.rsa.org.uk/

Web books I use myself. Lots of O'Reilly titles as they are written assuming a technically competent reader, and normally give a clear and concise treatment.

- Flanagan, D. (1997). *Java in a Nutshell 2nd Edition (Java 1.1)*. Sebastopol: O'Reilly & Associates.
- Flanagan, D. (1997). *JavaScript: The Definitive Guide (2nd edition)*. Sebastopol: O'Reilly & Associates.
- Gundavaram, S. (1996). *CGI Programming*. Sebastopol: O'Reilly & Associates.
- Lemay, L., K. Murdock and J. Couch (1996). *3D Graphics and VRML 2*. Indianapolis: Sams.net.
- Powers, S. (1998). *Dynamic HTML*. Foster City, CA: IDG Books.
- Schwartz, R. L. (1993). *Learning Perl*. Sebastopol: O'Reilly & Associates.

See also the Part I bibliography and the web version of this article which has active links to technology, examples and a mini-glossary:
<http://www.hiraeth.com/alan/papers/ActiveWeb/>

Alan Dix
aQtive limited, Birmingham Research Park
Vincent Drive, Birmingham, B15 2SQ;
School of Computing, Staffordshire University
PO Box 334, Beaconside, Stafford, ST18 0DG
email: alan@hiraeth.com

The Politics of Usability
A Practical Guide to Designing Usable Systems in Industry
Lesley Trenner and Joanna Bawa

The Politics of Usability
A Practical Guide to Designing Usable Systems in Industry
Lesley Trenner and Joanna Bawa
Springer, BCS Practitioner Series, 1998
194 + xxii pp. ISBN: 3-540-76181-0. £29.50.

Reviewed by
Dave Clarke, Visualize Software Ltd *and* Stella Mills, CGCHE

For many years, researchers have been studying HCI (Human-Computer Interaction) as a discipline in its own right, and as a result many new useful theories, methods and techniques have been proposed. These have been well documented in the past, and there is certainly no shortfall of texts on the subject, discussing in detail how these approaches work and the benefits that can be gained from using them. Although there is little doubt that using such techniques results in more usable systems, the problems often arise when one tries to adopt such approaches in commercial environments – one is now also faced with tight project deadlines, limited budgets, awkward teams and corporate politics in abundance.

This book is an attempt to show how to overcome many of these problems. Aimed at anyone intending to adopt usability engineering into a commercial organisation, it consists of a collection of well-defined chapters, written by those who have been faced with such problems before. The book tackles the majority of problematic issues a usability engineer is likely to face and the hurdles an organisation is to overcome. The various chapters discuss issues such as obtaining initial funding and how to make the most of it within an organisation, whilst also covering specific user interface design techniques and how they may be used. Politics and different cultures are addressed throughout, with later chapters addressing usability engineering on an international scale, tackling both cultural and methodological barriers. Usability consultants will also find several sections of interest to them in particular – establishing effective client relationships is covered along with how to promote their services as usability professionals in their own right.

The whole book is well written and easy to read, organised into clean, concise sections, which aids delving into individual chapters as required. The authors have clearly drawn from their experiences as usability professionals and this shows in what is an interesting and useful guide.

Value for money: 4 out of 5.

Dave Clarke
Visualize Software Ltd

This book is one of the latest in the BCS *Practitioner Series* which already has produced well-received titles on such computing areas as Key Java and Project Management. Each book in the series endeavours to give its readers up-to-date information not only on the subject itself but on its use in industry. These books are written for the practitioner as against the academic and so invariably their authors are practitioners themselves. In this sense, Trenner and Bawa's work is no exception since of the 18 authors contributing to the book, only three have academic allegiance.

Each of the 14 chapters is devoted essentially to a case study of experience within a specified organisation. Thus, Chris Nodder writes about setting up Natwest's Usability Service (Chapter 1), while Janet Saunders and Alan Arnfeld discuss the problems of developing a usability culture within Thames Water (Chapter 6). More generally, the chapters are clustered into four groups: the first two chapters form Part *1 - The Politics of Funding: Justifying your Existence*, while Part 2 (four papers) has the title *The Politics of Set Up: What to Do with the Money once You Have Got It.* Three papers form Part 3: *The Politics of Survival: Keeping Usability on the Political Map w*ith the remaining five papers covering Part 4: *The Politics of Expansion: How to Work effectively on an International Multi-Cultural Level.* Besides industry from the UK, that of other countries is also featured, with examples from the USA (Glaxo Wellcome Inc., Chapters 2 and 10) and Switzerland and Japan (Hewlett Packard, Chapter 14).

It should be noted that this book is not about improving interface design but highlights the ways that the politics of industry have been used to further the cause of usability-based methods in industry. As such, it is a very welcome addition to the literature but the academic should be warned that, by its very nature, few of the procedures are validated and referenced material is very thin. However, the book is not intended for academic reading although I suspect that many academics with an eye on industrial funding will find it exceedingly useful as will the teacher who wants *real* examples from industry to support other teaching material. There is no doubt that this book fills a gap in the literature and it should be read by all who have a practical interest in usability procedures.

Stella Mills
Cheltenham and Gloucester College of Higher Education

Alternative Realities

Our current conception of what HCI is and should be cannot possibly last. We are at a transition point in the relationship between people, information technology, and society, and need to break up a few of our cosy preconceptions about the field. **Alternative Realities** is a new regular section which is intended to serve as a forum for expressing much-needed alternative, and preferably controversial, views of what is, should, or will be going on. Contributions are sought which might be brief and jokey, or more serious in tone and deeper in argumentation. Articles should not be merely amusing though – we are looking for meaty issues behind the views expressed, however lightly. So, get it off your chest and write to **Alternative Realities**!

John Waterworth *Department of Informatics, Umeå University, S-901 87 Sweden. Email: jwworth@informatik.umu.se*

a child's eye view …

… taken from the UK-HOME-ED mailing list, November 1998, an exchange between two home-educating mothers

>To Kita it was just another toy … and
>unlike 'us' grownups [she] wasn't a bit
>frightened of tapping keys and never
>worried about losing everything on the
>screen …

Yup … we used to have an old monochrome Amiga computer and had a few 'kiddie' programs. We got them for our older son, who was about four at the time, but his brother Tim, at 2, was very keen to join in.

I will never forget waking up one morning and hearing Tim yelling in fury, 'NO, you silly 'puter! You're wrong!' We tiptoed through to find out what was happening …

Turned out he'd gone and switched on the Amiga by himself and loaded in one of the games (it had a windows-type interface so he just needed to click an icon). It was a colour-recognition game with voice software that asked 'What colour is this?', then showed a patch of colour, then, after a pause, said 'This colour is green' (or whatever).

Unfortunately, as I mentioned, it was a monochrome monitor, so everything appeared as shades of orange/brown on black.

So when it asked 'What colour is this?' Tim said 'orange', and when the computer announced 'This colour is green', he tried to argue! After about four of these he got angrier and angrier, yelling at the computer – and then, of course, randomly it would have orange as the colour, and he'd be pleased and think the computer understood at last … until the next one.

We realised then that Tim didn't think it was a game for *him* to learn about colours, he thought the computer wanted to know and was asking him, and then deliberately getting it wrong!

Used by permission

UK Home Education Support Internet mailing list

If you would like to join this free online discussion and support forum, send an email with just the following line in the message area:

SUBSCRIBE UK-HOME-ED Your Name

substituting your name for `Your Name'
and send this to:

LISTSERV@LISTSERV.AOL.COM

If you experience any difficulties subscribing,
please email:
UK-HOME-ED-REQUEST@AOL.COM

HCI Executive *Contact List*

Chair
Andrew Monk
University of York
Tel: +44(0) 1904 433148
Fax: +44(0) 1904 433181
Email: am1@york.ac.uk

Secretary & membership
Ismail Ismail
University College London
Tel: +44(0) 171 419 3664
Fax: +44(0) 171 387 1397
Email: I.Ismail@cs.ucl.ac.uk

Treasurer
Chris Johnson
University of Glasgow
Tel: +44(0) 141 330 6053
Fax: +44(0) 141 330 4913
Email: johnson@dcs.glasgow.ac.uk

Pauline Smith *(from January 1999)*
Nottingham Trent University
Tel: +44(0) 115 941 8418
Fax: +44(0) 115 941 6518
Email: mps@doc.ntu.ac.uk

Interfaces
Janet Finlay
University of Huddersfield
Tel: +44(0) 1484 472913
Fax: +44(0) 1484 421106
Email: j.e.finlay@hud.ac.uk

Meetings officer
Sandra P. Foubister
Napier University
Tel: +44(0) 131 455 5340
Fax: +44(0) 131 455 5394
Email: sandra@dcs.napier.ac.uk

Mailing list moderator
Adrian G. Williamson
University of Paisley
Tel: +44(0) 141 848 3752
Fax: +44(0) 141 848 3542
Email: A.Williamson@paisley.ac.uk

HCI Web resources
Simon Buckingham Shum
The Open University
Tel: +44(0) 1908 655723
Fax: +44(0) 1908 653169
Email: S.Buckingham.Shum@open.ac.uk

IwC editor
Dan Diaper
Bournemouth University
Tel: +44(0) 1202 523172
Email: ddiaper@bournemouth.ac.uk

BCS liaison
Stella Mills
Cheltenham & Gloucester College of Higher Education
Tel: +44(0) 1242 543231
Fax: +44(0) 1242 543205
Email: smills@chelt.ac.uk

SIGCHI liaison
Gilbert Cockton
University of Sunderland
Tel: +44(0) 191 515 3394
Fax: +44(0) 191 515 2781
Email: Gilbert.Cockton@sunderland.ac.uk

Conference planning
Chris Roast
Sheffield Hallam University
Tel: +44(0) 114 225 3763
Fax: +44(0) 114 225 3398
Email: C.R.Roast@shu.ac.uk

Practitioner representatives
Rory Channer
AIT Ltd
Tel: +44(0) 1491 416778
Fax: +44(0) 1491 416601
Email: Rory.Channer@ait.co.uk

Dave Clarke
Visualize Software Ltd
Tel: +44(0) 410 481863
Email: dclarke@visualize.uk.com

Ian Curson
Serco Consultancy
Tel: +44(0) 181 614 3784
Fax: +44(0) 114 614 3765
Email: icurson@usability.serco.com

Michael Gardner
BT Laboratories
Tel: +44(0) 1473 606500
Fax: +44(0) 1473 606759
Email: michael.gardner@bt-sys.bt.co.uk

David Jennings
David Jennings Associates
Tel: +44(0) 114 249 3435
Fax: +44(0) 114 249 3450
Email: dj@syspace.co.uk

Student representatives
Daphne Economou
Manchester Metropolitan University
Tel: +44(0) 161 247 1492
Fax: +44(0) 161 247 1483
Email: D.Economou@doc.mmu.ac.uk

Rakhi Rajani
Brunel University
Tel: +44(0) 895 274 000 *ext.* 2396
Fax: +44(0) 895 251686
Email: rakhi@dircon.co.uk

Peter Wild
Brunel University
Tel: +44(0) 895 274 000 *ext.* 2396
Fax: +44(0) 895 251686
Email: Peter.Wild@brunel.ac.uk

HCI Curriculum Group liaison
Linda Hole
Bournemouth University
Tel: +44(0) 1202 595251
Fax: +44(0) 1202 595314
Email: lhole@bournemouth.ac.uk

INTERACT '99 Conference
Alistair Kilgour
Heriot-Watt University
Tel: +44(0) 131 451 3438
Fax: +44(0) 131 451 3327
Email: A.C.Kilgour@hw.ac.uk

Ordinary member
Alan Dix
aQtive limited & Staffordshire University
Tel: +44(0) 121 414 7014
+44(0) 1785 353428
Fax: +44(0) 1785 353431
Email: A.J.Dix@soc.staffs.ac.uk

BCS CONTACTS
Sue Tueton (Membership) hci@bcs.org.uk
+44(0) 1793 417416
Andrew Wilkes (Committees)
awilkes@bcs.org.uk, +44(0) 1793 417471
Stephen Blanchard (Specialist groups)
Bob Hill (Printing) +44(0) 1793 417486

The British Computer Society
1 Sanford Street
Swindon SN1 1HJ
Tel: +44(0) 1793 417417
Fax: +44(0) 1793 480270
Email: hci@bcs.org.uk

Inter*faces* is published quarterly by the British HCI Group.