

---

# Hypermedia Support for Software Development: A Retrospective Assessment

TINA ROTH  
*TRW Systems Integration Group/FP1/7165U*  
*One Federal Systems Park Drive*  
*Fairfax, VA 22033 USA*

PETER AIKEN *and* SCARLETTE HOBBS  
*Department of Information Systems*  
*Virginia Commonwealth University*  
*Richmond, VA 23284-4000 USA*

Hypermedia technologies have been available for about a decade. Beginning with Douglas Engelbart, researchers have become interested in applying hypermedia concepts and technologies to software development. This paper assesses this symbiosis between hypermedia and software development as described by research contributions; by our count, more than one hundred articles. To date, no collection of, or guide to, this literature has been published. A number of significant contributions are classified, described, and appraised - providing a guide into what has been a rich but perhaps under-reported research area. Hypermedia has made significant contributions to software development in three primary areas: coping with massive amounts of information associated with software development projects; establishing and maintaining linkages between various types of software engineering documents; and enabling development engineers to record, document, and preserve knowledge about development thoughts, processes and rationale. These contributions may help direct the focus of future research towards building on, combining and assessing the resulting contributions. A research agenda is proposed along these lines.

## 1. INTRODUCTION

In *Augmenting Human Intellect: A Conceptual Framework* Engelbart suggested “the particular task of computer programming,” and programmers as “the most natural people in the world to be the very first users of the augmentation means they develop.”<sup>1</sup> This appears to be the earliest published suggestion that hypermedia could contribute to the process of software development. Since then more than one hundred articles have been published concerning hypermedia applied to aspects of software development. Much of the material has appeared in conference proceedings or has not received wide spread distribution in the software engineering community perhaps explaining why several topics have been 'discovered' several times over. This research describes, classifies, and appraises the potential symbiosis many researchers have discovered between hypermedia and software development. The paper is presented to: 1) function as a collection of domain knowledge references; and 2) focus future efforts towards building on existing research efforts eliminating unwitting duplication. Following a brief definition section; we set the context - describing hypermedia and major sources of research bases. Section three catalogs twenty applications of hypermedia applied to

software development. The paper concludes with assessments of the effectiveness of hypermedia on software development and recommendations for future research.

## 2. CONTEXT

### 2.1 Hypermedia defined

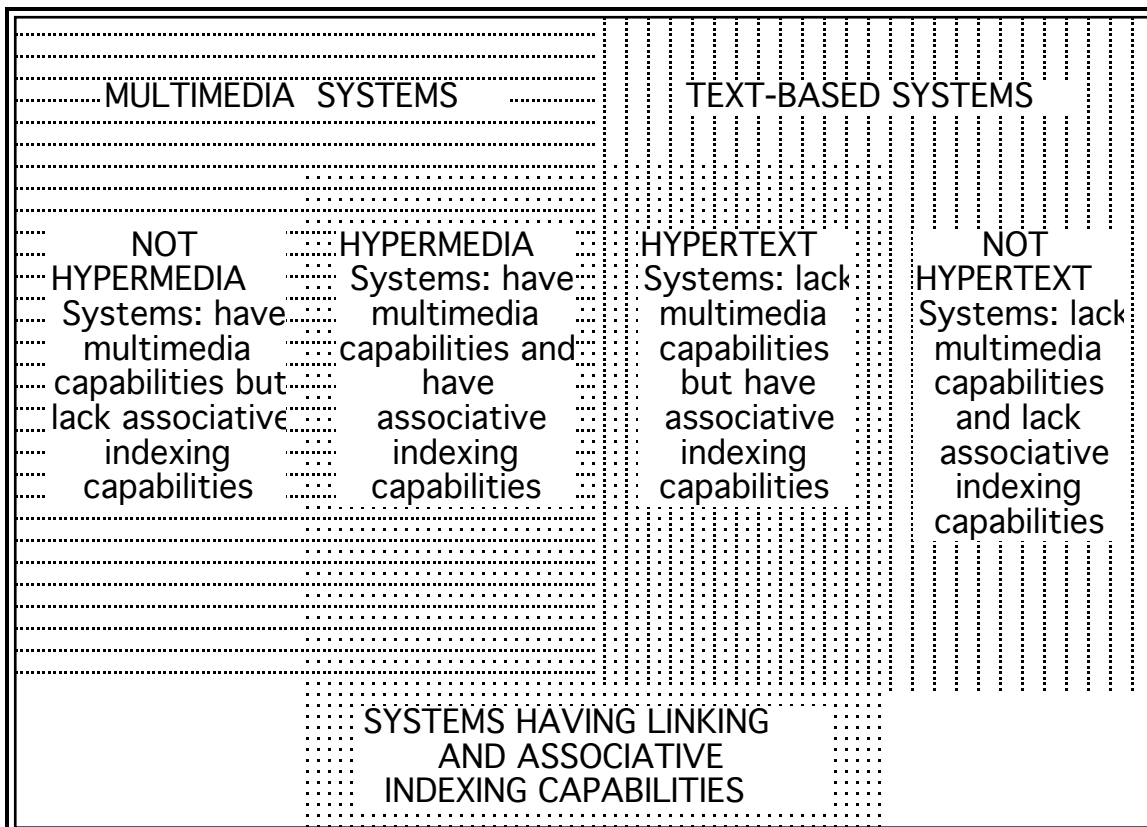


FIG. 1. *Classes of systems important to hypermedia*

Conklin (p. 7) has written "Hypermedia is not so much a new idea as a natural and general use to which to put computers."<sup>2</sup> Figure 1 graphically depicts basic hypermedia terminology by illustrating the relationships between three classes of computer systems. Briefly:

- Multimedia - defined as including or involving the use of several media."<sup>3</sup> Multimedia-based systems generally seek to create a more effective presentation by synergistically combining various media (i.e., text, graphics, video images, etc.) to enhance human interaction with screen events.<sup>4</sup>
- *Hypertext* - coined by Nelson: as " ... a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of a nonlinear text ... which can not be printed conveniently on a conventional page."<sup>5</sup>

- *Hypermedia* - defined by Frank Halasz as "a style of building systems for the creation, manipulation, presentation and representation of information in which: the information is stored in a collection of multi-media nodes; the nodes are explicitly or implicitly organized into one or more structures commonly, a network of nodes connected by links; users can access information by navigating over or through the available information structure."<sup>6</sup>

(For this paper, we use 'hypermedia' and include by definition systems with hypertext features.) The conceptual basis for hypermedia has been described by Nyce and Kahn<sup>7</sup> as well as by Conklin<sup>8</sup>, Smith<sup>9</sup>, and Nielsen.<sup>10</sup> Today, many commercial off the shelf (COTS) software systems incorporate forms of hypermedia. For example, the underlined phrases available to users when accessing help sub-systems in popular software packages are often hypermedia based. To obtain more information, users know to click on the underlined phrases (see Figure 2). Clicking permits users to conceptually traverse an associative link and view the related information. A subsequent link may permit the link to be traversed in reverse, returning the user to the point of origin.

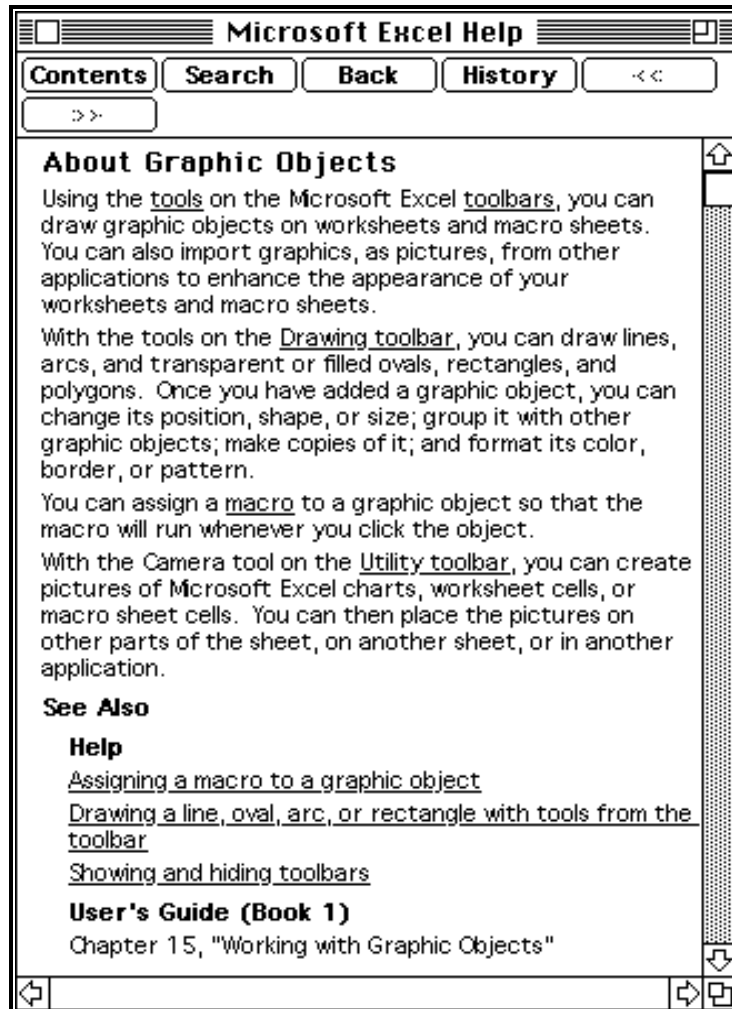


FIG. 2. The help panel from a popular spread sheet application. Underlines indicate the availability of eight additional sources of information on related topics

When contrasted with other classes of systems, hypermedia offers increased potential for information access and exchange; more versatile reporting; user-defined, non-sequential linking of presentation media; non-traditional linking of facts permitting wide-spectrum analysis and visualization techniques; function integration; a "tool kit" approach to information manipulation without regard for the information's medium; and simplified user-system interaction resulting from an increase in the available interaction modes.<sup>2 11 12</sup> Hypermedia systems are generally aimed at the needs of authors, designers, on-line readers, and other users performing knowledge intensive tasks. The systems provide tools for viewing and manipulating structure as well as content. The ability to work simultaneously with unstructured information and formalized, systematically organized, information provides alternate representational forms and structures.<sup>13</sup> These forms have attracted researchers attempting to see how they can be applied to basic software development difficulties.

## 2.2 Background literature

Four works have been particularly influential in the application of hypermedia to software development. These include publications by Engelbart, Conklin, Garg, and Akscyn. As previously mentioned Engelbart saw programming tasks as uniquely suited to benefit from what he termed 'augmentation research.' The programmers' involvement in a wide variety of tasks would lead to practical applications in a real-world context. (In 1962 programmers performed all phases of the software development life cycle.) As early adopters of technology, any success could be further leveraged to increase the research effectiveness.<sup>14 15 16</sup> Conklin's 1987 *IEEE Computer* article "Hypertext: An Introduction and Survey" is significant as the first widely disseminated description of the concepts and technology underlying hypertext. Bigelow's "Hypertext and CASE"<sup>17</sup> is apparently the earliest archival journal publication focusing directly on the potential synergy between hypermedia and computer aided software engineering (CASE) tools. It describes Neptune, a CASE environment capable of linking broad categories of software artifacts including:

- management reports ;
- specification and requirements;
- design, program documentation;
- implementation notes;
- source code;
- test specifications and results;
- object code; and
- products.

"Hypertext and Software Engineering" by Garg and Scacchi<sup>18</sup> is significant for its overall approach to supporting software development with hypermedia. Subsequent research influenced by these concepts has further extended and implemented these concepts as cataloged in the next section.

### 3. CATEGORIES OF HYPERMEDIA SUPPORT FOR SOFTWARE DEVELOPMENT

This section catalogs twenty types of hypermedia support for software development, divided into six categories. The categories are:

1. Hypermedia systems supporting argumentation/collaboration-based development activities
2. Hypermedia applied to software interface development
3. Hypermedia documentation
4. Hypermedia creation and maintenance of code
5. Hypermedia support for software artifact reuse
6. Hypermedia prototyping

The categories are not static, and specific contributions often reference multiple categories. Each category indicates specific entries describing hypermedia support and one or more references where the research can be found. In cases such as the application of hypermedia to software requirements engineering or to interface development, the same conclusions have been reached independently by multiple researchers. (Note: We have not attempted to catalog each and every contribution, thus inclusion or exclusion of specific research articles is not a judgment of the contribution references were included for their relevance and singularity.)

#### *3.1 Category 1: Hypermedia systems supporting argumentation/collaboration-based development activities*

Constructive criticism is one of the primary means development teams use to evolve and refine conceptual portions of analysis and design. Perhaps the most common instance occurs when lessons from previous development experiences are recalled by veteran personnel during reviews of evolving products. The research referenced below indicates hypermedia to be effective in assisting argumentation and collaboration development activities. Researchers have created systems supporting the rapid pace of creative analysis, design group discussions and individual thought sequences. Their research offers much to the process of recording, documenting, and preserving the often elusive thoughts and rationale occurring during the conceptual activities associated with software development.

*Support for cognitive momentum.* Lippman<sup>19</sup> describes findings regarding a critical requirement for hypermedia systems supporting argumentation/collaboration activities: *interactivity supporting cognitive momentum.* He proposes five specific requirements including: 1) interruptability, where individual threads can and often do interrupt each other; 2) granularity, identified as an appropriate size building block capable of ensuring successful interaction; 3) employing limited look ahead with a sufficient amount of

precomputed links; 4) permitting graceful degradation and recovery; and 5) giving the appearance of infinitude.

*Issue based information systems (IBIS)*. IBIS is a methodology developed by Rittel and Kunz<sup>20</sup> focusing on design discussions surrounding the deliberation of issues or design questions. Deliberation consists of assessing the pros and cons of the various issues, positions, and resolutions. The goal of the process is to improve design reasoning, rather than automate the design practices, with computer support functioning "like eyeglasses: helping you to see, rather than seeing for you."<sup>21</sup> A number of research efforts have implemented IBIS variations.

- Conklin and Begeman<sup>22 23</sup> describe gIBIS, a hypermedia system developed to support the design activities of construction and argumentation. gIBIS uses color and a quick access relational database system in a collaborative, distributed environment. The hypermedia system is used to capture information and artifacts about software development by creating a 'Design Journal' of the project development, integrating traditional documentation as well as designers' early notes, sketches and design decision/rationale.
- McCall *et al.*<sup>21</sup> describe PHIDIAS, an implementation of IBIS created for environmental designers supporting argumentation about construction. PHIDIAS features include: searching by structure and content; use of virtual structures; composite graphic nodes; query based graphic clustering; and 'triggered' queries connecting construction acts to sections of the argumentative network.
- Schuler and Smith<sup>24</sup> present Author's Argumentation Assistant (AAA), an environment for creating argumentation-based hyperdocuments. AAA is designed to support the creation of argumentation patterns using a hierarchical variation of IBIS. Argumentation structures are represented as a layered network of typed nodes and links in which different layers correspond to different levels of abstraction.
- Streitz *et al.*<sup>25</sup> extrapolate IBIS concepts to include idea processing and authoring using hypermedia to produce hyperdocuments. The authors view hyperdocuments as external manifestations of internal knowledge structures - again, potentially useful for representing design processes.
- Finally, Belew and Rentzepis<sup>26</sup> describe an IBIS-based extension of these concepts to permit users of the Internet user groups to treat these discussion groups as "a form of literature, worthy of the same preservation and augmentation that is typical of traditional printed media."

*JANUS*. Fischer *et al.*<sup>27</sup> describe JANUS, a hypermedia environment supporting architectural designers. The authors believe they can offer additional assistance to the argumentation process by providing designers with information concerning principles of design and the reasoning underlying them using intelligent agents. JANUS illustrates a practical integration of knowledge-based and hypermedia.

*Knowledge Management System (KMS)*. Akscyn. *et al.*<sup>28</sup> describe KMS, a large-scale commercially available distributed hypermedia system for networks of workstations that

has been used in many applications, including: electronic publishing, financial modeling and accounting, on-line manuals, user interface to videodisk-based materials, electronic mail and bulletin boards, and interfaces to other programs. One of the most popular features of KMS is the simplicity of the conceptual data model making the information stored in the hypermedia very accessible to users. Yoder<sup>29</sup> describes HyperLex™, a KMS-based hypermedia system supporting knowledge workers. Many of the features described are applicable to development oriented argumentation/collaboration activities.

*EUCLID*. Smolensky *et al.*<sup>30</sup> describe project EUCLID, a tool for helping people create and assess reasoned arguments and communicate the arguments with others. While the focus of the project is on user-document interaction and control of the screen, EUCLID also contributes ideas toward argumentation/collaboration activities, notably constraint-based hypermedia (CBH). CBH introduces the concept of perspectives used to influence the composition of the screen. Items added or deleted from the screen cause spatial relationships on the screen to be recomputed, a feature the authors find particularly useful in fine grained hypermedia applications.

*CONCORDE*. Hofmann *et al.*<sup>31</sup> describe how the process of knowledge acquisition is an essential step in the creation of expert systems. CONCORDE brings together and integrates the analysis, interpretations, and conclusions that a human uses when a problem is solved. This provides a quality representation of the process to be interpreted into a machine representation. CONCORDE supports knowledge acquisition for expert systems. Integration techniques used by CONCORDE provide a link to integrating databases, expert systems and hypermedia.

*Community Scholarship*. Catlin *et al.*<sup>32</sup> describe InterNote, an extension to the Intermedia hypermedia system, supporting group document review and revision. Of particular note is the system's ability to transfer data across links. Schnase and Leggett<sup>33</sup> describe an integrated computational hypermedia environment meeting 'personal information management needs of researchers' and 'dissemination of research results within a community of scholars.'

### 3.2 Category 2: Hypermedia applied to software interface development

One of the key areas where the application of hypermedia can be used to facilitate software development activities is in the area of interfaces. The relationship between hypermedia systems and interfaces is important because:

- hypermedia has been used as an interface paradigm providing 'access glue' to large collections of software development artifacts;
- interfaces to hypermedia systems have provided useful and practical examples;
- hypermedia has been used to provide access to collections of development reference materials; and
- hypermedia systems have been used to prototype software interfaces.

*"Gluing" Software Artifacts Together.* Fairchild *et al.*<sup>34</sup> describe hypermedia as a glue for representing and maintaining software development artifacts. They present requirements for development teams needing access to project artifacts in complex software production situations. The implementation of a prototype Lisp-based system called DELI Grapher focused on reducing the perceived complexity of the information and to producing information based on user expectations.

Another example of hypermedia as glue for a multimedia information system is described by Lucarella.<sup>35</sup> This system provides some ideas as to how multimedia information could be integrated into a knowledge base of system development artifacts and accessed by interface agents following inference rules. His conclusion: the interaction of hypermedias and information retrieval systems can bring obvious advantages to the process of knowledge transfer.

*Innovative Hypermedia Interfaces.* Card and Henderson<sup>36</sup> note the overhead often associated with task switching in windowing environments and identify nine properties of an interface designed to minimize these costs. The resulting proposed interface facilitates low overhead task switching in situations where users are performing concurrent tasks such as analysis and design. Their properties and evaluation should be useful for designing hypermedia-based tools supporting analysis/design and other concurrent development activities.

Examples of other hypermedia interfaces abound in the literature. In addition to Glushko's subsequently mentioned *Engineering Data Compendium*, other examples include Walker's<sup>37</sup> description of the interface considerations for Document Examiner - a commercial hypermedia system developed by Symbolics. Bieber<sup>38</sup> presents an example of a dynamic interface capable of integrating non-hypermedia documents. Laurel *et al.*<sup>39</sup> also present an example of hypermedia endowed interface agents.

*Hypermedia Access To Collections Of Reference Materials.* Collections of information about software development projects fit neatly into Schriederman's 'Golden Rules of Hypertext'<sup>40</sup> - information types appropriate to represent using hypermedia.

1. There is a large body of information organized into numerous fragments;
2. the fragments relate to each other; and
3. the user needs only a small fraction at any time.

As a result there are a number of hypermedia-based collections of reference material useful for software development. Glushko<sup>41 42</sup> describes projects incorporating the manual for MIL-STD-1472-D, *Human Engineering Design Criteria for Military Systems, Equipment, and Facilities*, and the *Engineering Data Compendium* - an 1138 entry collection of data on human perception and performance. Smith and Mosier's *Design guidelines for user-system interface software* has been transformed into a hypermedia collection called *Dynamic Rules for User Interface Development* (DRUID), by Fox.<sup>43</sup> The HyperCard version of DRUID functions as a computer-based design tool permitting users to select and weight the importance and relevance of functions and guidelines for specific applications. Niemeier<sup>44</sup> discusses SESAM (Systematics for Elaboration,

Selection and Assessment of Methods and Tools), a hypermedia database of development of methods and tools. Hochon<sup>45</sup> reports on the application of hypermedia to 'on-screen support.' Finally, Instone<sup>46</sup> reports on a set of references to hypertext research.

*Hypermedia Systems for Prototyping Software Interfaces.* There are many examples of hypermedia interface prototyping systems including Andriole,<sup>47</sup> Miller,<sup>48</sup> and Madsen and Aiken.<sup>49</sup> Put simply, hypermedia systems can provide flexible environments for user interface prototyping, permitting interactive user participation for interface development and evaluation and are growing in popularity. Hypermedia can almost be considered as an interface attribute.<sup>50</sup>

### 3.3 Category 3: Hypermedia documentation

The technology of the 90s has provided software designers and developers with an abundance of tools and platforms in which to perform their tasks. The interconnections and integration of these platforms and software packages can be assisted by hypermedia technology. Such systems can go a long way towards addressing problems associated with development information interconnectivity and synchronization. Perhaps more importantly hypermedia argumentation/collaboration systems can provide repositories for precisely the type of development information often not documented at all: crucial design decisions and the process by which the decisions were made.

Hypermedia assists in the integration of existing programming platforms and tools as well as traceability and documentation to new, changing environments. It provides integration facilities supporting divergent and different references and mediums. In addition to the previously described efforts to glue together software artifacts and the glue-like capabilities of the IBIS related efforts, hypermedia has been applying to linking traverse documents and versions of products to provide the most current and accurate information.<sup>51</sup> Integration of software documentation from different phases of the development life cycle supporting integrated access to code, documents, and system requirements can be found in research by Kaindl<sup>52</sup> <sup>53</sup> Garg<sup>54</sup> <sup>55</sup> Aiken<sup>56</sup> Conklin<sup>8</sup> Levene<sup>57</sup> Smith<sup>58</sup> Catlin<sup>59</sup> and Thebaut.<sup>60</sup>

In addition to increasing the accuracy and speed with which specific information about the development project can be retrieved, Bechtel<sup>61</sup> describes making CD-ROM (and other static storage media) a more comfortable environment for human use by augmenting text search algorithms, recovering lost benefits of the print medium, designing meaningful connections between documents to assist human communications, and allowing variable interactivity with the user.

Industry guru Martin provides a popular description of a hypermedia-based development environment: "The user of a hyperdocument reads it on a computer screen and can navigate through it at high speed, opening or closing parts of the document at the touch of a mouse key, linking to explanations, guidelines, diagrams, CASE screens, project management tools, and possibly expert-system modules, computer-based training, etc."<sup>62</sup> (p 11). Lease<sup>63</sup>, Arcieri, et al.<sup>64</sup> present a

implementations of similar concepts. Marovac<sup>65</sup> & Aiken<sup>66</sup> includes hypermedia concepts as guidelines for embedded software documentation.

### *3.4 Category 4: Hypermedia creation and maintenance of code*

Hypermedia has been applied to software code creation and code maintenance. The primary motivation has been to address the volume and complexity problems associated with software development.

*DynamicDesign.* Bigelow and Riley<sup>67</sup> describe DynamicDesign as a software engineering environment for the C language used by Tektronix. The facility permits developers to edit hypermedia objects, build hypermedia graphs from existing C source files, and browse source code, documentation and system requirements. The system is based on the Tektronix Hypertext Abstract Machine, a hypermedia-based transaction server capable of tracking change and version histories.

*Hypermedia Toolkits.* Puttress<sup>68</sup> [1990] describes a hypermedia-based toolkit permitting developers to introduce hypermedia features into their applications without requiring the systems to reside in existing hypermedia systems. The design of a hypermedia architecture and operating system allows applications to be integrated with other applications and share information. By developing this hypermedia connection the source code can be linked to other diagrams, code, COTS products, etc. Likewise, the links between elements in the code provide the development of source code graphics and reverse engineering facilities for pictorial documentation and direct access to the linked information. The actual source code can be used by hypermedia to create a tree-like structure of dependencies important for maintenance of the code over the life of the product

*COTS Hypermedia Construction Kits.* Schnase,<sup>69</sup> Schreiwies,<sup>70</sup> and Harris<sup>71</sup> describe hypermedia construction kits that eliminate computer-specific programming languages and provide a layer of abstraction over source code. This divorces the programmer from the use of specific languages and focuses on the human-problem domain communication and definition of a problem which may be coded or in the process of being coded. An environment for the programmer and the domain expert help to perform the computation and source code links.

### *3.5 Category 5: Hypermedia support for software artifact reuse*

Research has shown how hypermedia can be applied to the process of reusing software artifacts. Reusable hypermedia structures and repurposable hypermedia systems offer promise similar to reuse of code. Software development efforts have been shown to benefit from accessing connections to be made between applications design structures, architectures, and other pertinent information. These connections permit retrieval of information capable of lending insight to the design process. Reuse of software artifacts also permits reuse artifacts beyond code include items such as linkages, design decision rationale, software architectural structures. Reuse of any of these other types of artifacts permits savings in terms of not having to recreate the complex information webs and linkages surrounding the artifacts.

*Reusable Code.* Hypermedia-based environments have the capability of building graphs, linking to requirements, and browsing source code. This assists developers in understanding the code and the code structures. Investigations are reported by Biggerstaff<sup>72</sup> and Creech *et al.*<sup>73</sup> In cases of reverse engineering, links to requirements and other supporting documentation allow the development, maintenance and reuse of the code. Bigelow<sup>17</sup> Glushko<sup>51</sup> Fairchild *et al.*<sup>34</sup> and Pintado and Tsichritzis<sup>74</sup> describe similar reuse applications.

*Reusable Hypermedia Structures.* In many problem solving situations solution development is as of much interest as the actual solution. Bush recognized the value of preserving this information, stating, "The inheritance from the master becomes not only his additions to the world's record, but for his disciples the entire scaffolding by which they were erected"<sup>75</sup> (p. 108). It was Bush who first proposed reusing hypermedias as a means of preserving domain knowledge. Hypermedia provides representations of programs as knowledge networks where the code, documentation, and visual representations are external representations of the same complex internal knowledge structure. Capturing the decisions and knowledge used to construct computer programs through the use of hypermedia allows for increased potential to understand the software architecture and components. This in turn provides the essential knowledge to determine if reuse of the architecture or specific components is warranted.

Jordan *et al.*<sup>76</sup> describe Instructional Design Environment (IDE), a hypermedia-based multimedia environment facilitating rapid and accurate creation of regular network patterns in hypermedia. IDE facilitates the task of building representations in the instructional design process from conception of instructional ideas and objectives to delivery of actual instruction. IDE emphasizes the "front end" analysis portion of instructional design, where domain knowledge is analyzed and structured into semantic networks with specific patterns. IDE has a menu-based interface for defining network structures, creating new nodes containing structured content, and tailoring the interface to focus attention during different design phases.

While the focus is on eliminating the tediousness of creating, naming, organizing, and manipulating formal structures in an instructional design context, the concepts introduced can be generalized to software development. Catlin<sup>58</sup> proposes Hypermedia Templates capable of automated link creation. In addition, Schnase and Leggett's<sup>33</sup> computational hypermedia environment provides for reuse of the hypermedia structures. Finally, Aiken<sup>55</sup> offers extensions of Bush's original scaffolding concepts, including post mortem analysis of categories of requirements information and specific features which can lead to new methods and/or new criteria for collecting requirements information. For example, determining that the environmental characteristics were the most important type of information would lead to the suggestion that they should be collected prior to collecting information about the users and the tasks.

*Reusable Hypermedia Systems.* Glushko<sup>51</sup> describes the "look and feel" of various implementations of the previously mentioned *Engineering Data Compendium*. He describes the text structure granularity, treatment of non-textual components, links, access methods, and navigation and session support for implementations in three

popular commercial hypermedia systems Guide, HyperCard, and HyperTies. His assessment of the characteristics of the various products provides some guidance as to possible future uses in various contexts.

### 3.6 Category 6: *Hypermedia prototyping*

Prototyping continues to play an important role in software development. Strategies for effective prototyping range from written reports to well developed interactive prototypes (see Andriole<sup>77</sup>). Written system descriptions are easiest to produce but provide the least valuable information in terms of feedback. Users become disinterested in wading through pages of textual descriptions of the proposed system. In addition, it is difficult to gain a good mental image of the proposed system from a textual description. Better conceptual understanding can be gained from the use of storyboarding techniques and still better understanding comes from interactive prototypes.

As a rule of thumb, the more a prototype is based on terminology, functions, and images familiar to the user, the smaller the chance of a disconnect occurring. Generally, prototypes are produced to help define requirements accurately and thus provide the major source of feedback to the requirements engineering process. A number of researchers have suggested using hypermedia capabilities for prototyping including Miller<sup>31</sup> Nielsen<sup>78</sup> and Greitzer.<sup>79</sup>

While we did not find any formal studies evaluating the potential usefulness of hypermedia to the prototyping aspects of software development, there are a number of possibilities aimed at increasing the communication bandwidth between the requirements engineer and the user(s) evaluating the prototype and increasing the robustness of prototyping efforts. These include:

- hypermedia-based interactive prototypes with specific design features linked backward to statements of software and system requirements and forward to code supporting prototype features in the final product;
- high bandwidth hypermedia-based prototypes incorporating multimedia information captured during requirements accurately reproducing images, sounds, and procedures familiar to the user help them to evaluate the prototype in context;
- easy-access user interfaces used to make accessible interactive prototypes, increasing the user's ability to become involved in the prototype evaluation processes.

#### 4. ASSESSMENT OF HYPERMEDIA CONTRIBUTIONS TO SOFTWARE DEVELOPMENT

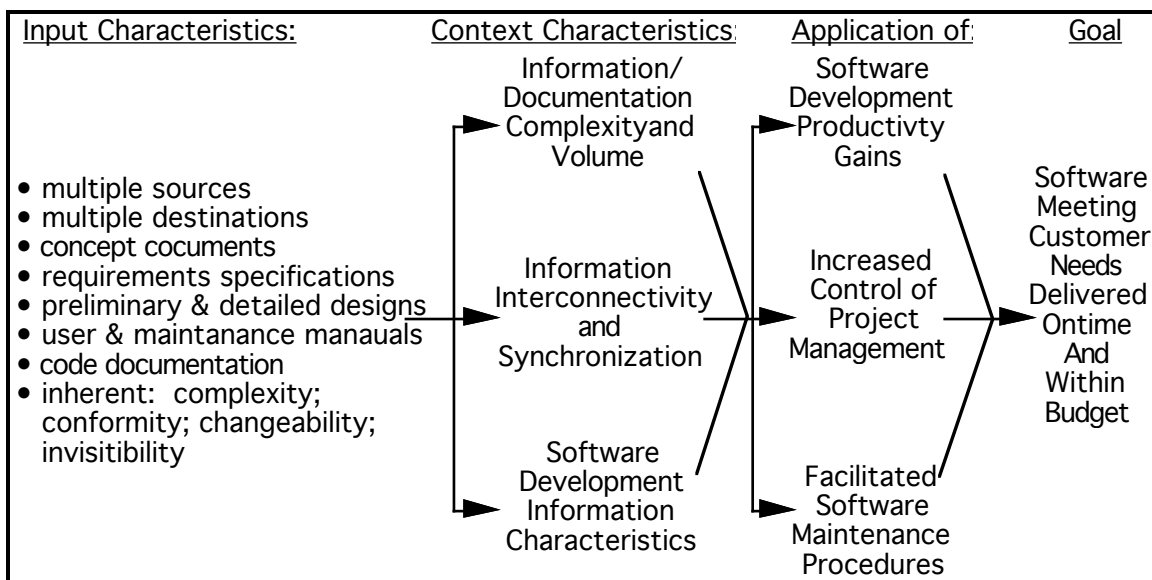


FIG 3. *Generic software development context.*

##### 4.1 Software development difficulties and obstacles

Current software development processes continue to require multiple tasks. Supporting software life-cycle activities requires significant documentation and communication. Documentation includes design specifications and a determination of the software's ultimate use. Equally important is the development of architectures, the rationale behind decisions, and solutions to development problems. Communicating this information among the development team is critical to the process. Brooks<sup>80</sup> has distilled these to four essential difficulties associated with software engineering. They are: 1. **complexity** - similar parts are combined into subroutines, but components are not alike - unlike projects such as construction, circuit boards, or airplanes; 2. **conformity** - software typically conforms to human constraints and the underlying complexity cannot be altered by changes to the software alone; 3. **changeability** - software requirements constantly change because they are subject to human users, laws and applications; and 4. **invisibility** - the lack of a concrete representation renders software invisible and unvisualizable, prohibiting developers from using powerful conceptualizing tools the way a design engineer might subject metal objects to stress. These difficulties combine to present the software developer with the situation shown in Figure 3. Context characteristics of this environment include:

1. **Information/documentation complexity and volume:** software engineers are faced with capturing, organizing, synthesizing, and presenting massive amounts of complex, interrelated information in order to produce software. The interim and final products have evolved beyond the software itself to include concept documents, requirements specifications, preliminary and detailed designs and design decisions,

user and maintenance manuals, code documentation, change requests, previous versions of the software, and corporate formalisms for coordinating the development, etc. Producing and accessing all of this information is a massive undertaking, growing geometrically with the complexity of the software being developed.

2. **Information interconnectivity and synchronization:** the evolutionary, iterative nature of software development requires development teams to plan for several stages, with each stage producing successively more detailed and accurate information about the software. This approach requires propagation of both information and of information changes for each iteration. Additionally, the contractual nature of software development may force developers to explicitly associate information between software development stages, for example, linking requirements to design, test plans to integration test results, and code to documentation.
3. **Software development information characteristics:** everyone involved in software development is charged, explicitly or implicitly, with documenting why things were done and in what manner. The nature and origin of this information, individual resistance to documentation, and standard work partitioning among team members contribute to the difficulties associated with documenting the processes required to turn user needs into formal requirements, designs, and code. Systems robust enough to capture this information and simple enough to use in the course of developmental activities have proven difficult to devise.

Advances in these three areas can positively affect software development activities resulting in gains in the overall quality of the software developed. From this perspective our assessment indicates progress in some areas but more remains to be done. Hypermedia systems have proven capable of providing support for overcoming software engineering information volume, information complexity and information characteristic obstacles (see O'Hara-Schettino and Gomaa<sup>81</sup>).

#### *4.2 Information volume obstacles*

Hypermedia strongly supports the management of information collected during development. It provides mechanisms for coping with massive amounts of information associated with software development projects, maintenance, and candidate reuse areas. In addition, it provides capabilities to capture and store decisions during development. Hypermedia systems have the two-fold advantage of providing interfaces which work in much the same manner as human thought processes and they provide integration capabilities with other system types (for example see Schlumberger<sup>82</sup>).

#### *4.3 Information complexity obstacles*

Hypermedia can provide an environment where individual information items can be organized into meaningful patterns. This makes it particularly appropriate for software development where it is used to establish and maintain links between various types of software engineering artifacts. The connection density of individual information items

can be increased up to the mental capacity of the engineer. Integrated hypermedia access to development information helps developers avoid wasting time and maintain current trains of thought. It also permits developers to concentrate on the information content of objects and not on the mechanics of getting it. Hypermedia documentation of architectures and algorithms help to mitigate complexity.

#### 4.4 Information characteristic obstacles

Hypermedia has two means of addressing information characteristic obstacles.

- First, research with IBIS and related systems have shown hypermedia to be an effective means of capturing invisible information about software development issues by enabling development engineers to record, document, and preserve development-relevant thoughts and rationale. By addressing these invisibility problems, hypermedia can provide value beyond the current development project as graphical hypermedia names, and linkages, etc. assist in visualization, maintenance, and reverse engineering projects.
- Second, hypermedia's close affiliation with multimedia technologies permits use of higher bandwidth information recording and processing technologies such as video and virtual reality. Use of multimedia can substantially increase the content and the robustness of information represented in development documentation.

As demonstrated in the discussions of the functional categories, capabilities provided by hypermedia-based technology can augment all phases of the development life cycle. Insight into these issues will eventually lead to generally accepted means of documenting thought processes and thus to new knowledge of conceptual development activities. More direct payoffs should come from productivity gains in the software development processes of software maintenance and software project management. Hypermedias have the inherent capability of meeting the requirements of user-oriented design of the human-computer interface as well as the demands for task-driven provision of functionality needed to support software development functions (see Figure 4). Within this context, one claim is that hypermedia systems will provide qualitatively new means for producing, communicating, and comprehending knowledge and thus will radically change the conditions of the information society.<sup>35</sup>

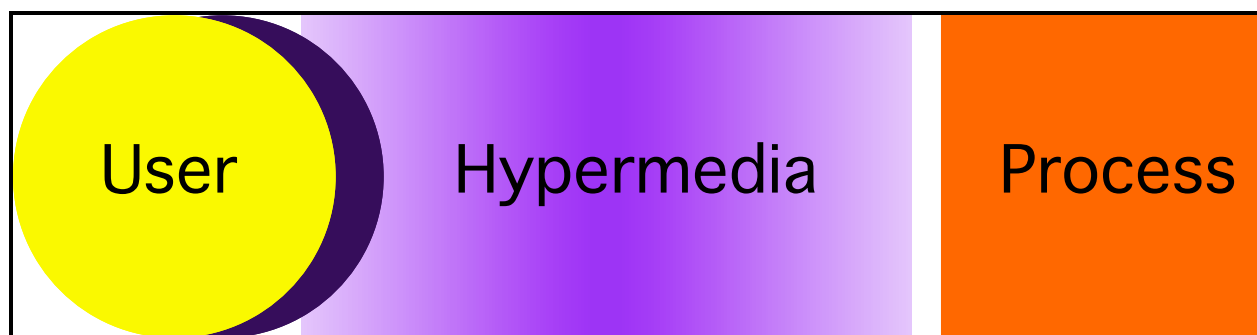


FIG 4. *The role of hypermedia.*

### 5. SUGGESTIONS FOR FUTURE RESEARCH

When preparing to research this article we anticipated locating and reporting findings along a continuum corresponding to the traditional system development life cycle (SDLC) classifications of analysis, design, code, test, and implement. Instead, the six categories of support we discovered were functional based indicating perhaps a different focus on the part of researchers. Perhaps now that researchers more fully appreciate the breadth of development work accomplished on the various pieces of hypermedia support for software development, they may shift their focus to the task of integrating these capabilities into existing development environments. If we examine the requirements for hypertext support for software development phases we will probably achieve more immediate success at getting the methods adopted than if we try to invent a whole new way of making software outside of existing environments. It seems valuable to consider the following perhaps building on them:

- Many CASE tools are supporting hypermedia facilities (see Figure 5).

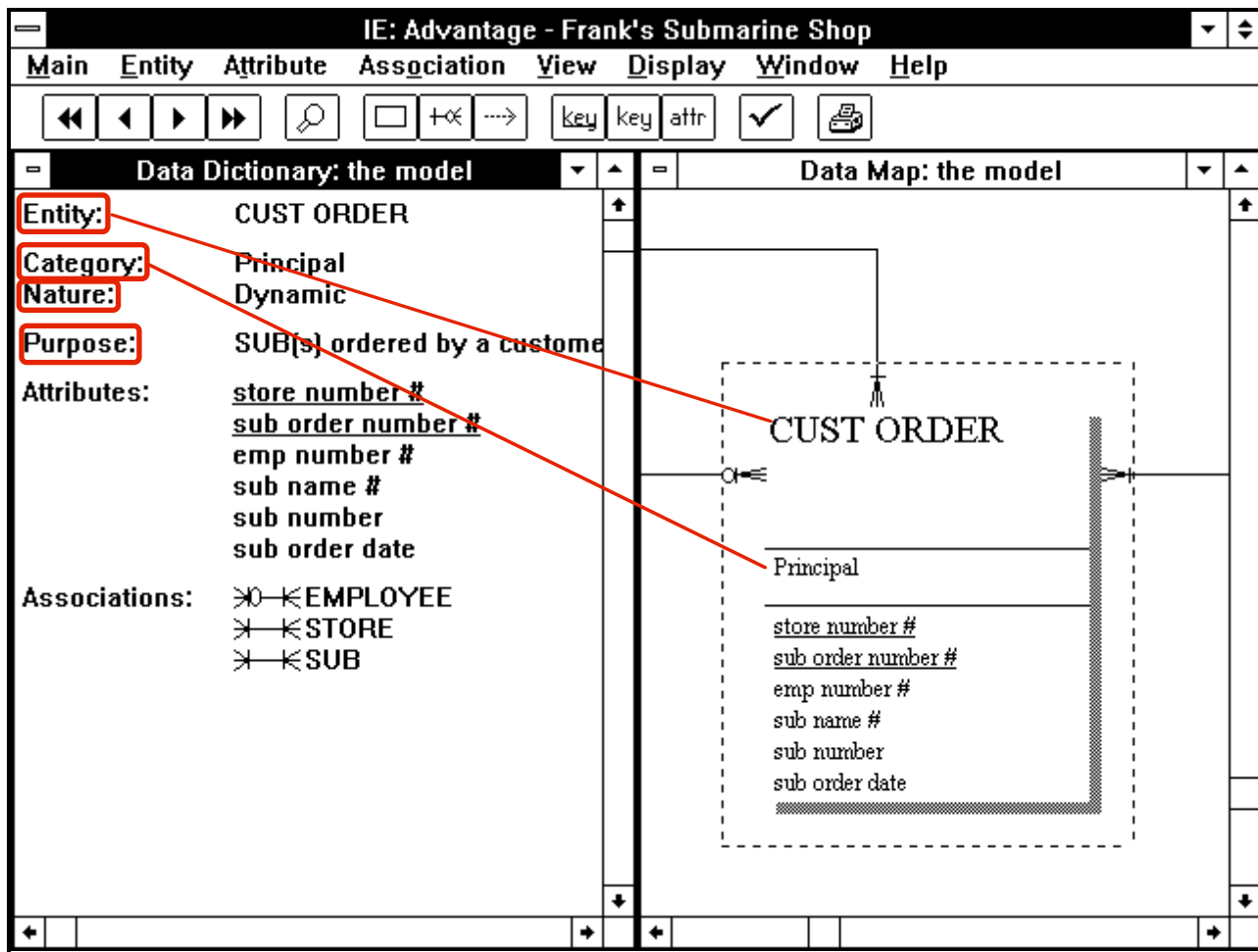


Fig 5. Screen shot from the CASE Tool IE: Advantage™ showing various information linking and synchronization capabilities

- Cybulski<sup>83</sup> presents an interesting approach to an architectural framework for integrating CASE tools under an extended hypermedia system (see also Palmer<sup>84</sup> and Gardner & Paul<sup>86</sup>).
- Mullerburg and Meyerhoff's<sup>87</sup> efforts to embed conceptual knowledge about software measurement in development environments.
- The trend towards hypermedia and interface development will continue as hypermedia-based interface applications continues to mature (see Kacmar<sup>88</sup>).
- Rugaber has been developing specifications for hypermedia support for software maintenance.<sup>89 90</sup>
- Hypermedia and various forms of decision support of interest to software developers (see Minch<sup>91</sup> Bieber<sup>37</sup> and Wolfe<sup>92</sup>).
- Real world attempts at implementation of these concepts (see Malcolm *et al.*<sup>93</sup>).
- Much to be learned from similar work in other domains such as architecture (see Gartshore<sup>94</sup> Spewack<sup>95</sup> and Connall & Burns<sup>96</sup>).

Thus much has been accomplished and much remains when coping with massive amounts of information associated with software development projects; establishing and maintaining links between various types of software engineering documents; and enabling development engineers to record, document, and preserve development-relevant thoughts and rationale.

### ACKNOWLEDGMENTS

The authors would like to thank Drs. Wynn Royce (TRW), Stephen J. Andriole (Drexel University), and Jennifer Papp (McGraw-Hill) for their valuable comments on earlier drafts of this paper.

The ACM Hypertext Compendium<sup>97</sup> has been an invaluable resource in performing this research. Editor Rob Akscyn describes the compendium as a hypertext database containing literature from the field. One of our goals is to unify, in a single hypertext database, a significant proportion of the literature of the field. Another goal is to learn and record lessons about the pragmatics of hyperitext publishing--lessons that may be useful to ACM and others in future hypertext publishing and standardization projects. The Compendium currently includes:

- Proceedings of Hypertext '87, Hypertext '89, and ECHT '90
- About 30 hypertext-related articles from other proceedings and journals

- A comprehensive bibliography of the field (with approx. 1300 authors)
- Various indices and cross-references to meaningfully interlink the material

Subsequent editions of the Compendium will incorporate more of the prior literature, plus new works as they are published [Akscyn 1991]. Its design and functionality made what would have been a very difficult task, manageable. Our extensions to the original indexing were easy to implement and, in conjunction with this paper, should serve as an effective introduction to users wishing to learn more about potential hypermedia-software engineering symbiosis. For this research we have created a web (or associative index), linking together the existing on-line articles referenced in this paper. It is available from the authors as a HyperCard stack supplement to the *ACM Hypertext Compendium*.

Excel is a trademark of Microsoft Corporation.

IE:Advantage is a trademark of Information Engineering Systems Corporation.

### AUTHOR'S NOTES

The author's names are listed in random order to denote equal contribution. Please direct any correspondence to Peter Aiken at the address given or paiken@acm.org

### KEYWORDS

Code, Design, Hypermedia, Hypertext, Integration, Prototyping, Reusable Components, Software Engineering, Software Requirements Analysis, Software Productivity, Specifications, Test.

### REFERENCES

- <sup>1</sup> *ENGELBART, D.* Augmenting Human Intellect: A Conceptual Framework Summary Report. *Stanford Research Institute, on Contract AF 49(638)-1024, October 1962, 134 pp.*
- <sup>2</sup> *CONKLIN, J.* A Survey of Hypertext.. *MCC Technical Report Number STP-356-86, 1986.*
- <sup>3</sup> *American Heritage Dictionary, Second Collage Edition. Boston: Houghton Mifflin Company, 1982.*
- <sup>4</sup> *BREWER, B.* Compact Disc Interactive Audio, in *S. Lambert and S. Ropiequet, eds.* CD ROM: the new papyrus. *Redmond, WA: Microsoft Press, 1986, pp. 273-294.*

- <sup>5</sup> NELSON, T. *Computer Lib*. Redmond, WA: Microsoft Press, 1987.
- <sup>6</sup> HALASZ, F. *Slides from Seven Issues Revisited*. Hypertext '91 Keynote Talk. Third ACM Conference on Hypertext, San Antonio, TX, December 15-18, 1991.
- <sup>7</sup> NYCE, J. and KAHN, P. *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. Boston: Academic Press 1991.
- <sup>8</sup> CONKLIN, J. *Hypertext: An Introduction and Survey*. IEEE Computer, September 1987, 20(9):17-41.
- <sup>9</sup> SMITH, J. and WEISS, S. *Hypertext*. Communications of the ACM, July 1988, 31(7):816-819.
- <sup>10</sup> NIELSEN, J. *Hypertext and Hypermedia*. New York: Academic Press, 1990.
- <sup>11</sup> BOLT, R. *The Human Interface: Where people and computers meet*. Belmont, California: Lifetime Learning Publications, 1984.
- <sup>12</sup> AKSCYN, R., MCCRACKEN, D. and YODER, E. *KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations*. Proceedings of the ACM Conference on Hypertext. Pittsburgh, Pennsylvania, November 1989, (*Hypertext '89*).
- <sup>13</sup> MARSHALL, C. *Exploring Representation Problems Using Hypertext*. Proceedings of the ACM Conference on Hypertext. Chapel Hill, North Carolina, (*Hypertext '87*), November 13-15, 1987.
- <sup>14</sup> ENGELBART, D. *A Conceptual Framework for the Augmentation of Man's Intellect*. Vistas in Information Handling, pp. 1-29.
- <sup>15</sup> ENGELBART, D. *The Augmented Knowledge Workshop*. In: Adele Goldberg, ed. *A History of Personal Workstations*, Reading, MA: Addison-Wesley Publishing Company, 1988, pp. 187-232.
- <sup>16</sup> ENGELBART, D. with HOOPER, K. *The Augmentation System Framework*. In: *Interactive Multimedia: Visions of multimedia for Developers, Educators & Information Providers* Ambron and Hooper - eds. Redmond, Washington: Microsoft Press, 1988, pp. 15-31.
- <sup>17</sup> BIGELOW, J. *Hypertext and CASE*. IEEE Software, March 1988, 5(2):23-27.
- <sup>18</sup> GARG, P. and SCACCHI, W. *A Hypertext System to Manage Software Life-Cycle Documents*. Reprinted from IEEE Software, May 1990.
- <sup>19</sup> Andrew LIPPMAN, A. as quoted In: S. BRAND, *The Media Lab: Inventing the future at MIT*. New York: Viking Penguin Inc., 1987.

- 20 *RITTEL, H. and KUNZ, W. Issues as Elements of Information Systems Working Paper 131. Institut fuer Grundlagen der Planung, Stuttgart, 1970.*
- 21 *MCCALL, R., BENNETT, P., D'ORONZIO, P., OSTWALD, J., SHIPMAN, F. and WALLACE, N. PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).*
- 22 *CONKLIN, J. and BEGEMAN, M. gIBIS: A Hypertext Tool for Team Design Deliberation. Proceedings of the ACM Conference on Hypertext. Chapel Hill, North Carolina, (Hypertext '87), November 13-15, 1987, pp. 247-252*
- 23 *CONKLIN, J. and BEGEMAN, M. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. Proceedings of the Conference of Computer-Supported Cooperative Work. Portland, Oregon, (CSCW '88), September 26-28, pp. 140-152.*
- 24 *SCHULER, W. and SMITH, J. Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).*
- 25 *STREITZ, N., HANNEMANN, J. and THURING, M. From Ideas and Arguments to Hyperdocuments: Travelling through Activity Spaces. Proceedings of the ACM Conference on Hypertext Proceedings. Pittsburgh, Pennsylvania, November 1989, (Hypertext '89).*
- 26 *BELEW, R. and RENTZEPIS, J. HyperMail: Treating Electronic Mail as Literature. ACM Conference on Office Information Systems. (COIS '90), Cambridge, Massachusetts, 1990.*
- 27 *FISCHER, G., MCCALL, R. and JANUS, M. : Integrating Hypertext with a Knowledge-Based Design Environment Proceedings of the ACM Conference on Hypertext Proceedings. Pittsburgh, Pennsylvania, November 1989 (Hypertext '89).*
- 28 *AKSCYN, R., MCCRACKEN, D. and YODER, E. KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations,. Hypertext '87 Papers, pp. 1-20.*
- 29 *YODER, E. Using Hypertext in a Law Firm, Second ACM Conference on Hypertext Proceedings. Pittsburgh, Pennsylvania, November 1989 (Hypertext '89).*
- 30 *SMOLENSKY, P.S., BELL, B., FOX, B., KING, R. and LEWIS, C. Constraint-Based Hypertext for Argumentation. Proceedings of the ACM Conference on Hypertext. Chapel Hill, North Carolina, (Hypertext '87), November 13-15, 1987.*

- <sup>31</sup> HOFMANN, M., SCHREIWEIS, U. and LANDENDORFER, H. *An integrated approach to knowledge acquisition by th hypertext system*. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).
- <sup>32</sup> CATLIN, T., BUSH, P. and YANKELOVICH, N. *InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration* Proceedings of the ACM Conference on Hypertext. *Proceedings Pittsburgh, Pennsylvania, November 1989 (Hypertext '89)*.
- <sup>33</sup> SCHNASE, J. and LEGGETT, J. *Computational Hypertext in Bibliogical Modelling*. Proceedings of the ACM Conference on Hypertext *Proceedings. Pittsburgh, Pennsylvania, November 1989 (Hypertext '89)*.
- <sup>34</sup> FAIRCHILD, K., MEREDITH, G. and WEXELBLAT, A. *The Tourist Artificial Reality*. Proceedings: ACM/SIGCHI Conference on Human Factors in Computing Systems. 1989.
- <sup>35</sup> LUCARELLA, D. *A Model for Hypertext-based Information Retrieval*. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).
- <sup>36</sup> CARD, S. and HENDERSON, A. *Multiple, Virtual-Workspace Interface to Support User Task Switching*. Proceedings: ACM/SIGCHI Conference on Human Factors in Computing Systems. 1987.
- <sup>37</sup> WALKER, J. *Document Examiner: Delivery Interface for Hypertext Documents*. Proceedings of the ACM Conference on Hypertext. Chapel Hill, North Carolina, (Hypertext '87), November 13-15, 1987.
- <sup>38</sup> BIEBER , M. *Issues in Modeling a "Dynamic" Hypertext Interface for Non-Hypertext Systems*. Proceedings of the ACM Conference on Hypertext. San Antonio, Texas (Hypertext'91), December 15-18, 1991.
- <sup>39</sup> LAUREL, B., OREN, T. and DON, A. *Issues in Multimedia Interface Design: Media Integration and Interface Agents*. Proceedings: ACM/SIGCHI Conference on Human Factors in Computing Systems. 1988.
- <sup>40</sup> SCHNIEDERMAN, B. and KEARSLEY, G. *Hypertext HandsOn! Addison-Wesley Publishing Company, 1989*.
- <sup>41</sup> GLUSHKO, R. *Design Issues for Multi-Document Hypertexts*. Proceedings of the ACM Conference on Hypertext *Proceedings. Pittsburgh, Pennsylvania, November 1989 (Hypertext '89)*.

- 42 *GLUSHKO, R. Transforming Text into Hypertext for a Compact Disc Encyclopedia. Proceedings: ACM/SIGCHI Conference on Human Factors in Computing Systems. (CHI '89) San Antonio, Texas, May 1989.*
- 43 *FOX, J. Dynamic Rules for User Interface Development. The MITRE Corporation, 1991.*
- 44 *NIEMEIER, J. and KIEBACK, A. Selecting methods and tools for information system improvement projects. In. Bullinger, H. ed. Human Aspects in Computing. Design and Use of Interactive Systems and Work with Terminals. Proceedings of the Fourth International Conference on Human-Computer Interaction Stuttgart, Germany Sept. 1-6 1991 p. 1151-9.*
- 45 *HOCHON, J. A. new generation of intelligent on-screen support: an architecture and a model. In. TOULOUSE '92. Fifth International Conference. Software Engineering and Its Applications Proceedings. Toulouse, France Dec. 7-11 1992 p. 25-33*
- 46 *INSTONE, K. The HCI Bibliography Project: A hypertext research perspective. ACM SIGLINK Newsletter, 2(3) December 1993, 7-8.*
- 47 *ANDRIOLE, S., EHRHART, L., AIKEN, P. and MATYSKIELA, W. Storyboard Prototypes for Group Planning & Decision-Making: Advanced User-Computer Interaction (UCI) & Multimedia Technologies. Accepted for IEEE International Conference on Systems, Man & Cybernetics. Cambridge, MA, November 14-17, 1989.*
- 48 *MILLER, D. Instant Prototyping Using HyperCard on the Macintosh. Proceedings of the Human Factors Society-32nd Annual Meeting. 1988, pp. 362-366.*
- 49 *MADSEN, K. and AIKEN, P. Some Experiences Using Cooperative Interactive Storyboard Prototyping. Communications of the ACM, June 1993 36(4):57-67.*
- 50 *AKIGUCHI, C., REKIMOTO, J., SUGAI, M., YAMAZAKI, G., MORI, T., UCHIYAMA, A., SANO, T., TARUMI, H. and IGARI, K. User interface platform 'Canae'. NEC Technical Journal, 45(3) March 1992 9-18*
- 51 *GLUSHKO, R. Using Off-the-Shelf Software to Create a Hypertext "Electronic Encyclopedia". Technical Communication, First Quarter 1990 (as listed in the ACM HyperText Compendium).*
- 52 *KAINDL, H. The missing link in requirements engineering. SIGSOFT Software Engineering Notes, April 1993 18(2) 30-9.*

- 53 *KAINDL, H. and SNAPRUD, M. Hypertext and Structure Object Representation: A Unified View In. Proceedings of the Third ACM Conference on Hypertext. San Antonio, TX December 15-18, 1991, pp. 345-358.*
- 54 *GARG, P. and SCACCHI, W. On Designing Intelligent Hypertext Systems for Information Management in Software Engineering. Hypertext '87 Papers, pp. 409-433.*
- 55 *GARG, P. and SCACCHI, W. Ishys Design an Intelligent Software Hypertext System. IEEE Expert, Fall 1989.*
- 56 *AIKEN, P. A Hypermedia Workstation for Requirements Engineering. Dissertation: George Mason University, 1989*
- 57 *LEVENE, M., POULOVASSILIS, A. and OFFEN, R. The hypernode model as the basis for a CASE information repository. In. R. Norman, and R. Van Ghent eds. Fourth International Workshop on Computer-Aided Software Engineering. Advance Working Papers. Dec. 5-8, 1990, Irvine, CA, p. 216-18.*
- 58 *SMITH, T.J. READS: A requirements engineering tool. In. Proceedings of IEEE International Symposium on Requirements Engineering Conference. San Diego, CA Jan. 4-6 1993, 94-7.*
- 59 *SMITH, K., CATLIN, L. and GARRETT, N. Hypermedia Templates: An Author's Tool. Proceedings of the ACM Conference on Hypertext. San Antonio, Texas (Hypertext'91), December 15-18, 1991.*
- 60 *THEBAUT, S., and INTERRANTE, M. Computer-aided requirements engineering: an operational modeling approach In. Staudhammer, J.; Qunsheng Peng eds. Second International Conference on Computer-Aided Design and Computer Graphics Hangzhou. China, Sept. 23-26 1991, p. 491-6.*
- 61 *BECHTEL, B. Inside Macintosh as Hypertext. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).*
- 62 *MARTIN, J. Hyperdocuments and how to create them. Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1990.*
- 63 *LEASEM, M., LIVELY, M. and LEGGETT, J. Using an issue-based hypertext system to capture the softare life-cycle. Hypermedia, 2(1) 1990, 29-46.*
- 64 *ARCIERI, F., BOZZO, A. and NARDELLI, E. A hypertextual editor for unified management of information system diagrams. In. Third International Workshop. Software Engineering and its Applications Proceedings. Dec. 3-7, 1990 Toulouse, France p. 605-18*

- 65 *MAROVAC, N. Guidelines for embedded software documentation. SIGSOFT Software Engineering Notes, April 1994 19(2) 22-8.*
- 66 *AIKEN, P. and BACH, J. Document Management As A Prerequisite To Quality Assurance. 1990 IEEE International Conference on Systems, Man & Cybernetics Conference Proceedings. November 4-7, 1990 Los Angeles, CA, pp. 739-741.*
- 67 *BIGELOW, J. and RILEY, V. Manipulating Source Code in Dynamic Design. Proceedings of the ACM Conference on Hypertext. Chapel Hill, North Carolina, (Hypertext '87), November 13-15, 1987.*
- 68 *PUTTRESS, A. J. and GUIMARAES, B. M. The Toolkit Approach to Hypermedia. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).*
- 69 *SCHNASE, J. and LEGGETT, J. Computational Hypertext in Bibliogical Modelling. Proceedings of the ACM Conference on Hypertext Proceedings. Pittsburgh, Pennsylvania, November 1989 (Hypertext '89).*
- 70 *SCHREIWEIS, U., KEUNE, A. and LAGENDORFER, H. An integrated Prolog programming environment. SIGPLAN Notices, April 1993 28 (4) 21-8.*
- 71 *HARRIS, R. Support technologies for avionics operational flight programs. In: Proceedings of the IEEE 1991 National Aerospace and Electronics Conference. NAECON 1991 Dayton, OH, USA 20-24 May 1991 p. 660-4.*
- 72 *BIGGERSTAFF, T.J. Hypermedia as a Tool to Aid Large Scale reuse. MCC Technical Report Number STP-202-87, July 1987.*
- 73 *CREECH, M.L. , FREEZE, D.F. and GRISS, M.L. Using Hypertext in Selecting Reusable Software Components. Proceedings of the ACM Conference on Hypertext. San Antonio, Texas (Hypertext'91), December 15-18, 1991.*
- 74 *PINTADO, X. and TSICHRITZIS, D. SaTELLite: Hypermedia Navigation by Affinity. Proceedings of the European Conference on Hypertext. Versailles, France (ECHT '90).*
- 75 *BUSH, V. As We May Think. Atlantic Monthly, July 1945, pp. 101-108.*
- 76 *JORDAN, D. Facilitating the Development of Representations in Hypertext with IDE. Griss "Using Hypertext in Selecting Reusable Software Components" Proceedings of the ACM Conference on Hypertext. San Antonio, Texas (Hypertext'91), December 15-18, 1991.*
- 77 *ANDRIOLE, S.J. Storyboard Prototyping: A New Approach to User Requirements Analysis. QED Information Sciences, Wesley, MA, 1992.*

- 78 *NIELSEN, J. Prototyping user interfaces using an object-oriented hypertext programming system. Proceedings of the NordDATA'89 Joint Scandinavian Computer Conference. (Copenhagen, Denmark, 19-22 June), pp. 485-490.*
- 79 *GREITZER, F., WUNDERLICH, D. and WEINBERG, M. Hypermedia-based rapid interface prototyping. Journal of the Society for Information Display, Jan. 1993 1(1) 111-119.*
- 80 *BROOKS, F.P. No Silver Bullet: Essence and Accidents of Software Engineering. IEEE Computer, April 1987, pp. 10-19.*
- 81 *O'HARA-SCHETTINO, L. and GOMAA, H. Addressing Volume and Complexity in Software Engineering: A Hypertext Approach. technical working paper. Center for Software Systems Engineering, George Mason University 1992.*
- 82 *SCHLUMBERGER, P.C. Shadow: Fusing Hypertext with AI. IEEE Expert, Winter 1989, 65-78.*
- 83 *CYBULSKI, J. and REED, K. A hypertext based software-engineering environment. IEEE Software, 9(2) March 1992 62-68.*
- 84 *PALMER, J.D. and AIKEN, P. Utilizing Interactive Multimedia To Support Knowledge-based Development Of Software Requirements. Proceedings of the 5th Annual RADK Knowledge-based Software Assistant Conference. September 24-28, 1990 Saracuse, NY, pp. 105-120.*
- 85 *PALMER, J.D., AIKEN, P. and FIELDS, A. A Computer Supported Cooperative Work Environment for Requirements Engineering and Analysis. Proceedings of the Requirements Engineering and Analysis Workshop. March 12-14, 1991, Software Engineering Institute Special Report SEI-91-SR-12.*
- 86 *GARDNER, L. and PAUL, R. Developing a hypertext geographic information system for the Norfolk and Suffolk Broads Authority. Hypermedia, 1993 5(2) 119-43.*
- 87 *MULLERBURG, M., MEYERHOFF, D. The METKIT CAI system. Information and Software Technology, Feb. 1993 35(2) 120-6.*
- 88 *KACMAR, C. Suporting Hypermedia Services in the User Interface. Hypermedia, 5(2) 1993, 85-102.*
- 89 *RUGABER S. Requirements for a hypertext software maintenance system. Proceedings of Compcon. Spring'89 San Fransisco, California February 1989.*

- <sup>90</sup> RUGABER S. *Hypertext and software maintenance system*. Proceedings of the CASE'88: Second International Conference on Computer-Aided Software Engineering. Cambridge, Massachusetts July 1988
- <sup>91</sup> MINCH, R.P. *Application And Research Areas For Hypermedia In Decision Support Systems* Journal Of Management Information Systems, Winter 1989-90, 6(2):119-138.
- <sup>92</sup> WOLFE, M. *Development of the city of quality: a hypertext-based group decision support system for quality function deployment*. Decision Support Systems, March 1994, 11 (3) 299-318.
- <sup>93</sup> MALCOLM, K.C., POLTROCK, S.E. and SCHULER, D. *Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise*. Proceedings of the ACM Conference on Hypertext. San Antonio, Texas (Hypertext'91), December 15-18, 1991.
- <sup>94</sup> GARTSHORE, P., YOUNG, J. and NEWLAND, P. *A multiwindow hypermedia environment for the delivery of information to architects, engineers, and product designers*. Hypermedia, 3(2) 1991, 119-132.
- <sup>95</sup> SPEWAK, S.H. *Enterprise Architecture Planning* QED Publishing, Boston, 1993.
- <sup>96</sup> CONNALL, D.G. and BURNS, D.R. *Reverse Engineering: Getting a Grip on Legacy Systems*. Data Management Review, October 1993, pp. 24-27.
- <sup>97</sup> AKSCYN, R. ed. *The ACM Hypertext Compendium*. ACM Press, 1992.