Tool-supported Evolutionary Web Development: Rethinking Traditional Modeling Principles

C. BAUER
bauerc@cbs.curtin.edu.au
School of Information Systems
Curtin Business School
Perth, Australia

A. SCHARL
scharl@wu-wien.ac.at
Information Systems Department
University of Economics & BA
Vienna, Austria

***Abstract* – In this paper, an evolutionary Web information systems development methodology is proposed, together with a set of modeling tools with special regard to the document-oriented visualization of individual and accumulated Web access patterns. The requirement for evolutionary development is derived from the evolutionary nature of Web-based Information systems, but is not sufficiently addressed in current development techniques. The paper suggests suitable development methods and tools for design, implementation, usage and analysis of Web information systems to advance commercial development.**

I. INTRODUCTION

Commercial applications for the World Wide Web are deployed in and impacted by an environment that is constantly redefined by evolutionary change. A natural response to these new challenges is to adopt an evolutionary approach in developing Web information systems. The development of Web information systems includes all organizational and technology-related aspects from initiating the project to deploying and maintaining Web applications of variable complexity. It has to be pointed out that Web information systems development shows substantial differences compared to traditional information systems development [2]. In the context of such a framework, the development process is split into four interconnected phases: design, implementation, usage, and analysis including visualization.

Many companies still struggle with the sustainable and consistent implementation of Web information systems throughout their whole organization. The authors propose that these struggles are due to the evolutionary nature of (commercial) Web information systems that is not appropriately reflected in current Web information systems development methodologies and consequently methods. However, integrated software architectures applicable in commercial environments require the availability of appropriate methodologies and tools. Ideally, the latter should support a complete evolutionary feedback loop in order to facilitate successful deployment in commercial environments. This paper provides a framework of development methods for the design, implementation, usage and analysis of Web information systems. Furthermore, the integration of such methods with each other and with "case-tools" for efficient organizational deployment is outlined.

II. EVOLUTION PROCESSES OF WEB INFORMATION SYSTEMS

An evolutionary approach to software development faces the challenge of conceptually describing, analyzing, and designing complex systems that feature emergent behavior and evolve unpredictably from an initial set of simple elements. The process of modeling Web information systems requires creativity and intelligent planning during the initial analysis, design, and implementation which is referred to as "internal (r)evolution" in Fig. 1. For that reason, Martin introduces the term "intelligent evolution" [14]. With special emphasis upon corporate business behavior, he compares three types of evolution with the classic Darwinian evolution based on the survival of the fittest:

(a) Internal (r)evolution during the pre-deployment phase:

- First order evolution, modifying a product or service (Web-based application) within a cyclical, predesigned process and taking into account the existing corporate structure; this concept is rather similar to *product innovation* as defined by [26]. Frequently, multiple product innovations are required at the same time to be adopted by the envisioned target group [20, 25].
- Second order evolution, modifying the process, methodology, or fundamental design of work (development methodology for Web-based applications). Some authors use the term *process innovation* to describe this organizational change, which usually aims at reducing costs or raising quality.

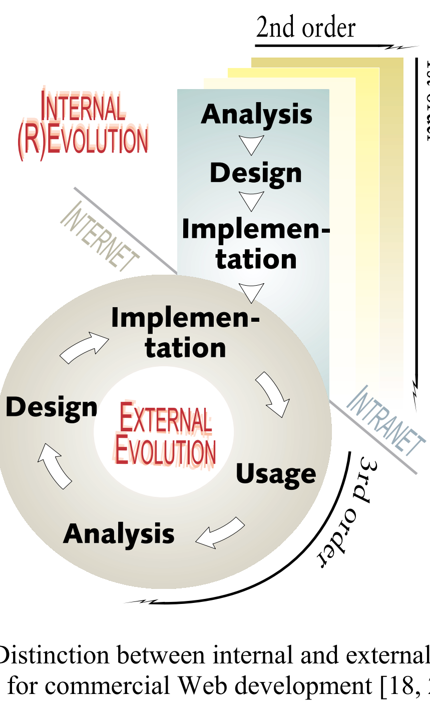

Fig. 1. Distinction between internal and external evolution for commercial Web development [18, 22]

(b) External evolution:

- Third order evolution, considering external factors outside the corporation (e.g., relationships with customers, other companies, governmental institutions, standardization committees, etc.).

First order evolution mirrors the classic approach of system analysis and design methodologies. Frequently, predesigned processes and static corporate structures are considered as independent variables not subject to evolutionary change.

Limitations of this traditional approach – e.g., lack of continuity or treating system evaluation and quality control as post-hoc processes [1] – might be overcome by propagating second order evolution as well, optimizing and extending development methodologies, reengineering the corresponding processes, and questioning established concepts and theories regarding the organizational integration and strategic importance of Web information systems. Especially the last category, third order evolution, relies on the concept of a business ecosystem comprised of several independent, but closely cooperating organizations [15, 16]. All members of such an ecosystem are responsible for the prosperity of this particular system. For establishing a new business ecosystem, even competitors have to cooperate – they are allies in the competition with other business ecosystems but rivals within the boundaries of their own system. The evolutionary context of business eco-systems and first, second and third order evolution of commercial Web information systems has already been discussed elsewhere in further detail [2].

III. Integrated Case-Tools for Developing Complex Web Applications

The emphasis on third order evolution of Fig. 1 in this paper is a logical consequence of the strategic and operational importance of permanently improving and restructuring commercial Web information systems. In most cases, regular maintenance and incremental (re-)development outweigh the costs and efforts of the initial start-up phase by far. The concentration on the external evolution in a business-to-consumer (B2C) scenario provides a significant reduction in modeling complexity and allows for a clear and intuitively comprehensible explanation of the Web information systems development process.

Consequently, Fig. 2 reduces the more complex, multidimensional evolutionary development process of Fig. 1 to external evolution, which is split into four sequential phases:

- Design,
- Implementation,
- Usage, and
- Analysis.

The *design* phase includes various general aspects of Web information systems, for example provision of content material, Web layout and guidelines, navigational structure, and so forth. Ideally, the design will completely define the actual *implementation* without requiring further specifications and intervention.

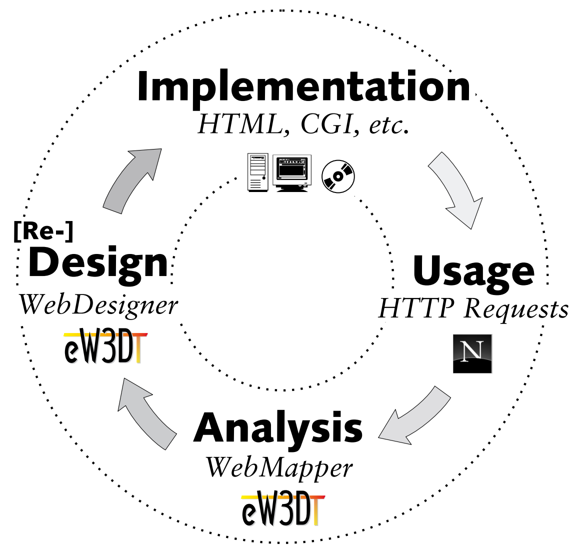

Fig. 2: The on-going Web development process in a cybernetic feedback loop

Once accessible, users (customers of commercial Web information systems, and – more specifically – consumers in a business-to-consumer context) will begin to utilize the Web information system. Their behavior in the *usage* of on-line resources can be directly observed. Additional, explicit user feedback about these activities can be obtained within and outside of the Web information systems. In commercial environments, the *analysis* of usage patterns, strengths, and weaknesses as well as the statistical interpretation of the current offerings are a vital business necessity. Necessary improvements and critical issues can be identified and spark *(re-)design* of Web information systems. The modeling cycle, therefore, closes a cybernetic feedback loop by feeding the outcomes of the analysis back into the design phase.

While almost all developers are performing the tasks associated with these four phases, the execution of these functions is not always manifested in formal models, but instead performed through mental models and rather informal constructs. Examples of more rigid models and concrete implementations are presented in Fig. 2. In the following sections, each of the four phases are explained in more detail and implications and early tools are introduced.

A. Design

The systematic design of commercial Web information systems requires suitable development methodologies and modeling languages. Web information systems, in contrast to traditional information systems, are undergoing constant change and evolution. They are rarely considered to be in a “finished”, or completed, state. Ongoing maintenance and re-development induce significant change to the structure and content of on-line systems. Analogous to traditional systems development approaches inadequately dealing with the dynamic evolution of Web information system, traditional modeling techniques are only of limited use for representing and reflecting the diversity and permanent transformation of associatively linked hypertext documents. The introduction of new modeling techniques and system representations, therefore, represents a necessity for conceptual Web information system design. These techniques have to reflect structural interdependencies and the emphasis on content as far as commercial Web information systems are concerned.

Several modeling techniques have been proposed for Web information system design, for example the *Hypermedia Design Model | HDM* [7], the *Object-Oriented Hypermedia Design Model | OOHDM* [24], the *Relationship Management Methodology | RMM* [11], or the *(extended) World Wide Web Design Technique | (e)W3DT* [5, 23]. However, in order to be commercially successful and to support the Web information system development of companies competing on time, any modeling technique needs to be embedded into a Web site management tool supporting user-friendly authoring of HTML documents. Some of these modeling techniques have already been implemented in prototypes, but usually without sufficient control and integration of the actual HTML implementation, e.g., RMM in *RM-Case*, or W3DT in *WebDesigner*. In contrast to database-oriented concepts, eW3DT was built from scratch to support the requirements of unstructured, hierarchical sets of hypertext documents and to visualize them from a recipient's perspective. Models of Web information systems should support structured as well as unstructured information and are aimed at reducing the communication gap between domain experts and system professionals, providing an agreed semantics for the conceptual data and navigational model [10]. They serve as interpretative guideline for people with very heterogeneous technical expertise and professional responsibilities. Conceptual models relying on eW3DT are a user-centric combination of structural and process diagrams [12], which requires an explicit explanation of symbols. This explanation respectively notation will be presented in the following paragraphs.

Fig. 3. Standard symbolic element of eW3DT

Every eW3DT data object type represents a specific variation of a standard symbolic element depicted in Fig. 3 and is equivalent to an atomistic unit of the Dexter Hypertext Reference Model [9]. Together with an (optional) differentiation by color, the sub-symbol (S) on the right side of the object name signals the basic type of the information object. The hierarchical level where the document in question usually can be found within a hypertext application has to be specified in the bottom left field (x). The second digit (y) describes optional sub-components. An interaction, implemented as part of the homepage, would receive the value 1.1. The eW3DT meta-model distinguishes between technical and content-specific responsibilities for designing, implementing, and maintaining Web information systems. Two abbreviations next to the hierarchical level refer to functional units responsible for content (CNT) and technical implementation (TEC). In the bottom right field, one to three '☆'-symbols represent the maintenance intensity of information objects (initial efforts to implement documents are not considered). Interfaces to existing (marketing) databases influence this value substantially. Independent of iconic similarity and real equivalence to a given object (hypertext compound document), every information object type defines a general profile for describing the characteristic attributes of this object.

Fig. 4 categorizes the object types of eW3DT into the three functional segments Information, Navigation, and Structure. With the help of these elements it is possible to visualize Web-based hypertext structures of variable complexity, no matter if they are intended for a real organization (implementation model) or for an industry-specific analysis (reference model). While an actual example of an eW3DT diagram will be shown in section D (Fig. 8), please refer to [22, 23] for a more detailed description of eW3DT object types.

B. Implementation

Presumably, future applications will feature the same functionality and integration between models and implementations for commercial Web information systems as it is already common for Upper- and Lower-CASE tools in traditional systems development. Given syntactically correct Web information systems models with complete specifications of the content information, navigational relationships and layout, the files for a representation of a Web information system at the implementation level can be generated automatically. Whether this generation takes place after each update and results in a set of static files on a Web server, or the HTML files are generated on-the-fly according to the specifications of the underlying model, is only of secondary concern in the context of the conceptual modeling strategy. The information architecture of the actual implementation has to be determined according to issues such as scalability, technical infrastructure, and update frequency.

One of the shortcomings of HTML is the mixture of content, structural, navigational and formatting information, and the tags of the markup language. The lack of separation between these inherently different types of data adds to the complexity of developing corporate Web sites with consistent layout, up-to-date information and syntactically correct document representation. The HTML syntax is designed for hypermedia representation and optimized for the document rendering requirements of early Web information systems (and has been extended with forms, frames, tables, and other more advanced formatting constructs). As a consequence, HTML provides rich facilities for display, but no standard way to manage meta-data, or exchange semi-structured information in wide area networking. The XML efforts of the W3C (http://www.w3.org/XML) are the most notable attempt to overcome these limitations of HTML.

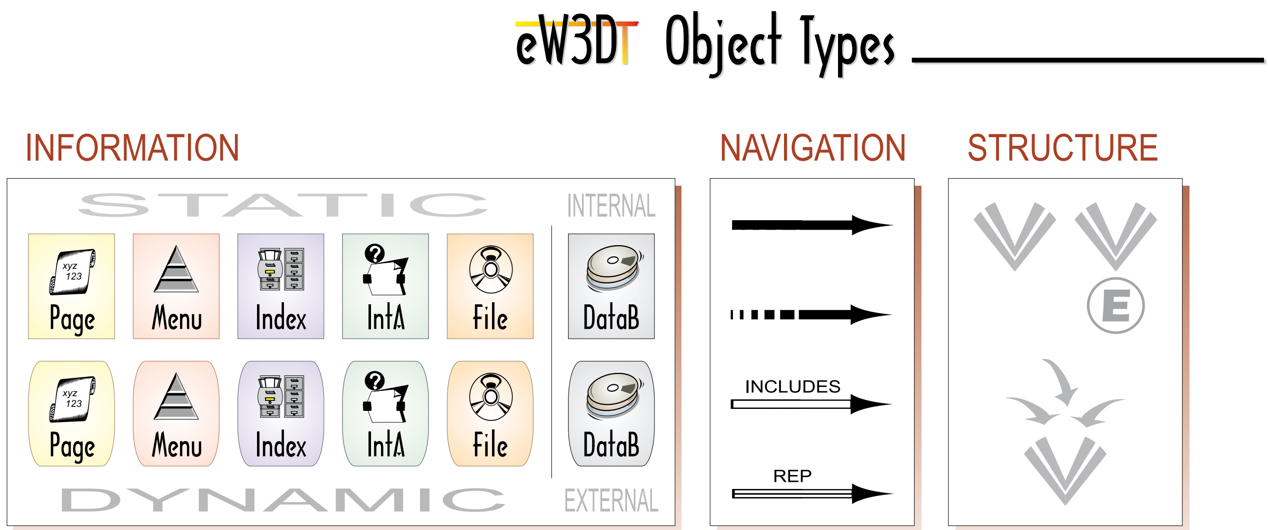

Fig. 4. Object types of eW3DT, categorized into the functional segments *Information*, *Navigation*, and *Structure*

They introduce a variety of standards such as the Extensible Stylesheet Language (XSL), or the XML Linking Language (XLink) to provide a common platform for representing structured data on the World Wide Web. Whereas HTML as presentational markup language imposes a lowest common denominator for document rendering and inextricably mixes presentation and representation (content and structure), XML as a semantic markup language is extensible, validatable by external modules, and provides self-documenting tags [8]. It forces the separation between presentation and representation by storing the definitions for non-standardized tags in a separate Document Type Definition (DTD) file. A DTD is the formal specification for documents of a given type, describing the constituting elements, their attributes, and the order in which they have to appear. Unfortunately, the tool support of XML is still in its infancy. The viewing of XML documents in common browsers has not been implemented yet (Netscape *Communicator*, Version 4), or does not support it completely (Microsoft *Internet Explorer*, Version 5). The authoring support is even poorer, commercially available *WYSIWYG (What You See Is What You Get)* editors are just about to be introduced into the market (for example SoftQuad's XMetaL – http://www.xmetal.com/).

Until the XML standard is widely accepted and adopted amongst Web developers and users worldwide, corporate Web implementations will have to rely on HTML for document delivery. The Web information system's server-side physical representation, on the other hand, is not bound to these restrictions, and can provide a more adequate structure of the information given a conversion process as outlined above. An early example of such an approach is the *WebDesigner*, a prototype based on a Prolog-style database converting visual W3DT models into HTML and CGI files. A more recent implementation, LifeWeb [17], separately stores the information as content material, link information and presentation formats in XML files. A set of Java classes and servlets parses these XML files and responds to the HTTP request in HTML format. A similar approach that includes the specific requirements of corporate development environments for the World Wide Web is envisaged for the implementation of eW3DT models. Fig. 5 presents the top level of the class hierarchy in the meta-model of eW3DT in an object-oriented notation [19, 21]. Essentially, it is a formal description of eW3DT, consistent with the more verbal description provided in section A.

A site is composed of at least one document, which is defined by its content (material), link structure, presentation layout, and several organizational and technical attributes. The content material is defined in either one of six categories: simple pages, menus for hierarchical refinements, indices, interactive modules, special file formats, or database interconnections. If all this information is provided during the design phase of Web information systems development, direct translation into Web-compatible data formats – e.g., HTML, XML, or Adobe's Portable Document Format (PDF) – and the automated generation of executable application processes such as CGI-scripts or JAVA-Applets become feasible.

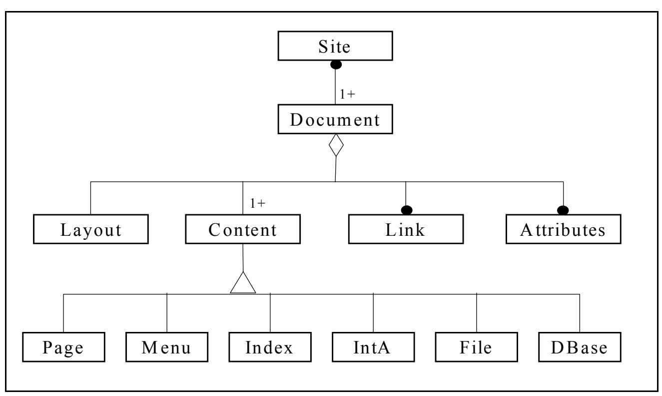

Fig. 5: Object model notation of eW3DT's top-level hierarchy

C. Usage

The usage phase establishes the direct contact to the customer. Therefore, it is the only source of direct feedback for the Web information systems developer. Once the implementation has been released to the public (in the case of business-to-consumer Electronic Commerce), the organization's customers will use the Web information system. An important success criterion to guarantee the quality of the user feedback is a critical mass of users requesting information, performing transactions, and/or participating in online evaluations. Since this criterion belongs to the general business objectives for successful Electronic Commerce applications, it does not lead to a dilemma of resource allocation. The information of implicit and explicit customer feedback through the usage of Web information systems is invaluable for all organizations.

In order to help the information provider to map and classify the customer's behavior, a prototypical, platform-independent Java software tool called *WebMapper* can be employed to provide a visual framework for analyzing access patterns of actual and potential on-line customers. In its early stages, this „clickstream" application exclusively focuses on the processing of HTTP log-files, enhancing the representations of commercially available analysis software. In these log-files, the user's IP address and computing platform, the referring document, and the exact time of access are recorded. Derived from the IP address, the domain name, workplace and approximate geographical location can easily be determined. Some examples of tracking and analysis tools based on HTTP log-file data are [6, 13]:

TABLE 1: COMMERCIAL WEB-TRACKING SOFTWARE PACKAGES

Product	Company	URL
ARIA	Andromedia	http://www.andromedia.com
Bazaar Analyzer Pro	Aquas	http://www.aquas.com
GuestTrack	GuestTrack	http://www.guesttrack.com
HitList Professional	Marketwave	http://www.marketwave.com
I/PRO	Nielsen Interactive Media	http://www.nielsenmedia.com
Net.Analysis Pro	Net.Genesis	http://www.netgen.com
NetIntellect	WebManage Technologies	http://www.webmanage.com
NetTracker	Sane Solutions	http://www.nettracker.com
WebTrends	WebTrends	http://www.webtrends.com

These tools, however, only provide statistically oriented representations embedded in various reports (including tables, bar charts, etc.) usually being generated directly in HTML or in a file format compatible with popular word processing software. In contrast to that, WebMapper will provide a graphical overview based on eW3DT and analogous to traditional customer tracking which is quite common for real-world retailing outlets [4]. The WebMapper is an example of an analysis tool with a systematic modeling language, and represents different types of hypertext compound documents in the rectangular symbols (Fig. 6). However, these WebMapper symbols incorporate a completely different set of attributes in comparison with the eW3DT design meta-model (compare Fig. 3). While the color respectively the shading of objects signifies their number of HTTP requests during a certain period (N_Hits), the style of connecting links between the documents represents the frequency with which these links where followed by customers. In addition to that, the average viewing time of documents in seconds is displayed in the field (Avg_Vtime). With the (Info) button, detailed information about the object in question is accessible (e.g., a list of host names or IP addresses of the most important visitors, aggregated number of entries and exits, and so forth). Being part of the user interface, the two arrow symbols in the bottom right corner do not represent an attribute of the object but provide the analyst with the option to move between lower-level and upper-level diagrams.

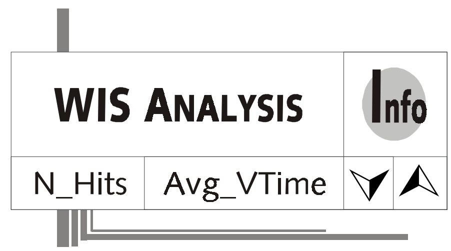

Fig. 6. WebMapper standard symbolic element

Such a usage map provides an interesting complement by showing sequence at a higher level, exposing the kinds of experiences which users can get from Web information systems. [27] denotes two main application areas: Guidance of new users and Web application design. In the latter case, the experience users are having is compared with the designer's model(s) of how the Web information system would be experienced. The structure of the site's hierarchical document tree is automatically generated either from meta-information of the design phase or from the hyperlink information found within existing HTML documents. WebMapper will help companies running commercial Web information systems in their efforts to map and classify individual as well as aggregated customer behavior. It will enable them to predict future trends, to advertise more effectively, and to maximize the customer delivered value of electronic transactions.

D. Analysis

Web developers try to overcome one-way communication, which is normally a characteristic of broadcast-oriented mass media. They do so by looking at two types of user response – i.e., feedback in a cybernetics system approach as introduced by Norbert Wiener in "Cybernetics or Control and Communication in the Animal and the Machine" [28], the term being derived from the ancient Greek word kybernêtês (steersman):

1. On-line forms and early CGI-based programs are used to gather **explicit** customer information (continuous arrows Fig. 7).

2. Log-file analysis, visualizations of user clickstreams, and persistent client state HTTP cookies provide **implicit** information. Additionally, more sophisticated technologies will gradually become popular (Fig. 7; dotted arrows).

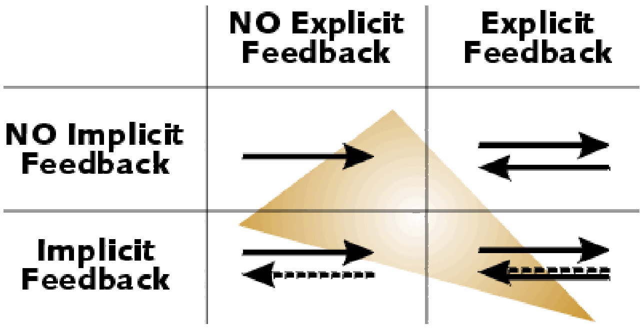

Fig. 7: Dominant communication models

Some implicit information is always transmitted with every HTTP request. However, developers may choose not to exploit this source of information due to various reasons, which range from lack of competencies to incurred total cost or privacy concerns. Therefore, the utilization of implicit information is defined as gathering such data, analyzing it, and reacting accordingly. Future additions will include sets of attributes for user profiling as well as additional object types required for modeling adaptive system components. Summarizing this visual framework for analyzing access patterns of on-line customers and comparing it to the design model presented in section B, Fig. 8 depicts a map-like view similar to customer tracking in traditional retailing outlets [4].

Besides the implicit and explicit feedback gathered online from the Internet as summarized in Fig. 7, developers can also rely on explicit information from their customers that is gathered via other communication channels, e.g. from past customer records or third-party surveys. Another important analysis tool is benchmarking the Web information system against competitor's efforts.

An approach into this direction is the *WebAnalyzer* as presented in [3]. The WebAnalyzer provides the numeric input for the quantitative analysis by mirroring the publicly available content of Web information systems and parsing the retrieved HTML code. Several variables (e.g., images, file size, internal and external links, and so forth) are derived and a corresponding input vector for further processing in a neural network is generated. Subsequently, the evaluation tool will use an adaptive neuro-fuzzy architecture to determine appropriate categories for any given Web information system. The analysis of their industry and competitors' efforts provide decision makers with benchmark data about relevant differences and the relative performance of their own Web information system. Benchmarks indicating weaknesses should trigger and guide the immediate redevelopment and optimization of the application structure.

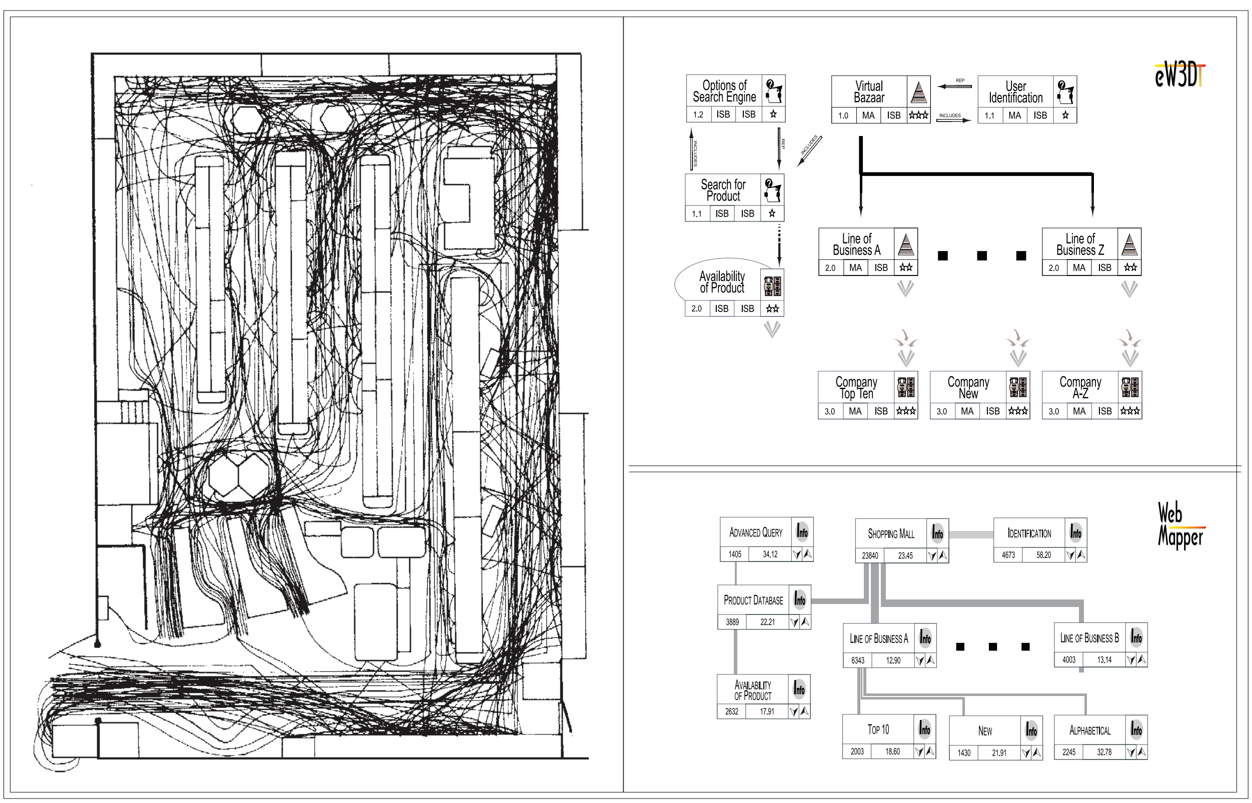

Fig. 8. Customer tracking for traditional retailing outlets versus eW3DT and WebMapper

IV. Conclusion

Changing Web technologies raise the need for corresponding modifications and extensions of Web modeling techniques and tools. While almost all developers of Web applications are performing numerous interconnected tasks associated with the four cyclical phases presented in this paper (design, implementation, usage, and analysis), more structured approaches are needed to model and visualize complex hypertext structures. Currently, no standard interfaces between most of the phases exist:

- **Design – Implementation:** The development of the extended World Wide Web Design Technique has been motivated by the lack of non-redundant, readable meta-models for Web information systems that do not ignore their hierarchical, recipient-oriented hypertext structure. Currently, there are no tools available that would integrate a visual navigational and content design method with an automated generation of a Web information system. ColdFusion's CFML (ColdFusion Markup Language) and dynamic server information architecture are the prime exemplar of the integration between design and implementation, but is still showing weaknesses in the design functionality in terms of visualization and modeling rigor compared to the standard of commercial CASE-tools for database applications. Even tools with a particular emphasis on design, for example Macromedia's DreamWeaver, do not offer strong support for the modeling of complete Web information systems, nor the automated creation of implementations from design models.
- **Usage – Analysis:** Web server log files and log analysis tools provide some, (pseudo-) standardized integration between usage and analysis. However, when information of higher granularity and visualization is required, these tools and data formats are not sufficient any more. More powerful tools are readily available, but do not integrate well with other phases and less structured information (for example from on-line customer surveys).
- **Analysis – Design:** The integration between these two phases is particularly weak, despite its overwhelming importance for an evolutionary approach, and for closing the feedback loop. Even tools such as ColdFusion, with rich and large information architectures, do not implement any connections between these two phases.

As it can be seen from the above reviews, integrated tools and/or interface standards are required to enable the commercial deployment of evolutionary Web information systems development. The notion of evolutionary development and the importance of user feedback needs to be implemented in the Web information system development infrastructure to outperform competitors. The conceptual models in this paper illustrate potential and infrastructural requirements for the implementation of an evolutionary Web information systems approach.

References

[1] B.L. Allen, *Information Tasks: Toward a User-Centered Approach to Information Systems*. Library and Information Science, ed. H. Borko. 1996, San Diego: Academic Press.

[2] C. Bauer, B. Glasson, and A. Scharl, *Evolution of Web Information Systems: Exploring the Methodological Shift in the Context of Dynamic Business Ecosystems*, in *Doing Business on the Internet: Opportunities and Pitfalls*, C.T. Romm and F. Sudweeks, Editors. 1999, Springer: London. p. 53-64.

[3] C. Bauer and A. Scharl. *A Classification Framework and Assessment Model for Automated Web Site Evaluation*. in *7th European Conference on Information Systems (ECIS-99)*. 1999. Copenhagen, Denmark: Copenhagen Business School.

[4] W. Becker, ed. *Beobachtungsverfahren in der demoskopischen Marktforschung*, ed. O. Strecker and R. Wolffram. 1973, Ulmer: Stuttgart.

[5] M. Bichler and S. Nusser. *Modular Design of Complex Web-Applications with W3DT*. in *5th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-96)*. 1996. Stanford, USA: IEEE Computer Society Press.

[6] D.D. Busch, *Count Your Blessings.* Internet World, 1997. **8**(6): p. 75-83.

[7] F. Garzotto, P. Paolini, and D. Schwabe, *HDM: A Model-based Approach to Hypertext Application Design.* ACM Transactions on Information Systems, 1993. **11**(1): p. 1-26.

[8] M. Ginsburg. *Annotate! A Tool for Collaborative Information Retrieval*. in *7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 1998. Stanford, USA: IEEE Computer Society Press.

[9] F. Halasz and M. Schwartz, *The Dexter Hypertext Reference Model.* Communications of the ACM, 1994. **37**(2): p. 30-39.

[10] C.J. Hemingway. *Toward a Socio-cognitive Theory of Information Systems: An Analysis of Key Philosophical and Conceptual Issues*. in *IFIP WG 8.2 and 8.6 Joint Working Conference on Information Systems: Current Issues and Future Changes*. 1999. Helsinki, Finland: IFIP.

[11] T. Isakowitz, E.A. Stohr, and P. Balasubramanian, *RMM: A Methodology for Structured Hypermedia Design.* Communications of the ACM, 1995. **38**(8): p. 34-44.

[12] G.L. Lohse, K. Biolsi, N. Walker, and H.H. Rueter, *A Classification of Visual Representation.* Communications of the ACM, 1994. **37**(12): p. 36-49.

[13] R. Malchow and K. Thomsen, *Web-Tracking.* Screen Multimedia, Sep 1997, 1997: p. 57-61.

[14] J. Martin, *Cybercorp: The New Business Revolution*. 1996, New York: Amacom.

[15] J.F. Moore, *Predators and Prey: A New Ecology of Competition.* Harvard Business Review, 1993. **71**(3): p. 75-85.

[16] J.F. Moore, *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. 1997, New York: Harper Collins.

[17] T.-L. Nguyen and H. Schmidt. *Creating and Managing Documents with LifeWeb*. in *5th Australian World Wide Web Conference (AusWeb-99)*. 1999. Ballina, Australia: Southern Cross University Press.

[18] H. Österle, *Business Engineering: Prozeß- und Systementwicklung, Band 1: Entwurfstechniken.* 1995, Berlin: Springer.

[19] Rational, *Unified Modeling Language: Resource Center*, 1999, Rational Software.

[20] E.M. Rogers, *Diffusion of Innovations*. 4th ed. 1995, New York: The Free Press.

[21] J. Rumbaugh, et al., *Object-oriented Modeling and Design*. 1991, Englewood Cliffs: Prentice-Hall.

[22] A. Scharl, *Referenzmodellierung kommerzieller Masseninformationssysteme: Idealtypische Gestaltung von Informationsangeboten im World Wide Web am Beispiel der Branche Informationstechnik (= Reference Modeling of Commercial Mass Information Systems: Referential Design of Applications for the World Wide Web with Special Regard to the Information Industry)*. Europäische Hochschulschriften: Reihe 5, Volks- und Betriebswirtschaft. 1997, Frankfurt: Peter Lang.

[23] A. Scharl. *Reference Modeling of Commercial Web Information Systems Using the Extended World Wide Web Design Technique (eW3DT)*. in *31st Hawaii International Conference on System Sciences (HICSS-31)*. 1998. Hawaii, USA: IEEE Computer Society Press.

[24] D. Schwabe and G. Rossi, *The Object-oriented Hypermedia Design Model.* Communications of the ACM, 1995. **38**(8): p. 45-46.

[25] A. Treloar. *Just Another Technology? How the Dynamics of Innovation Can Help Predict the Future of the Browser*. in *5th Australian World Wide Web Conference (AusWeb-99)*. 1999. Ballina, Australia: Southern Cross University Press.

[26] J.M. Utterback, *Mastering the Dynamics of Innovation*. 1996, Boston: Harvard Business School Press.

[27] A. Wexelblat. *History-based Tools for Navigation*. in *32nd Hawaii International Conference on System Sciences (HICSS-32)*. 1999. Hawaii, USA: IEEE Computer Society Press.

[28] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine*. 2nd ed. 1961, New York: MIT Press / John Wiley & Son.