Software Engineering Experiences While Implementing Internet-based Business Processes

David A. Marca
University of Phoenix
100 Grossman Drive
Braintree, MA 02184 U.S.A.

ABSTRACT

This paper presents a Software Engineering tool set for developing process-oriented Internet applications. This paper also presents experiences using this tool set to implement two industry quality business processes. These experiences provide practical Software Engineering principles for implementing Internet-based business processes: adaptable e-business connections, interconnected business-process-technology architectures, process-centric user interfaces, process conciseness and process personalization, and process branding.

Keywords: Software Engineering Environment, IDEF0, Process Modeling, Internet Application, Architecture and Hypertext.

1. ADAPTABLE E-BUSINESS CONNECTIONS

Experience #1: *When every part of a business process can quickly make an e-business connection to a customer or supplier, the operational aspect of that business has been maximized to its highest level of adaptability.*

Engineering Reduces Risk. If a business built plants to produce products without first designing a manufacturing process, the risk would be lack of capacity without significant plant redesign. Similarly, software developers who lack an approach and a tool set for developing Internet-based business processes before creating them, risk: 1) designing business partnerships incorrectly, 2) not implementing the process quickly enough, or 3) having operations that cannot adapt to changes in business direction. Therefore, correct Software Engineering is a key resource for the implementation of e-business connections.

E-Business connections can be defined by assessing their effect on the supply chain, that is, the network of suppliers and customers within which the business operates [1]. For example, a computer manufacturer has a chain of companies that supply it with disk drives, monitors, software, etc. It also has a chain of customers in its retailers, resellers and end users. Critical here is the potential for an e-business connection to implement any portion of that supply chain. Thus, an adaptable business process has the capability to allow a software engineer to make an e-business connection at any point in the entire business process.

Production-focused connections are the implementation objective for Internet-based business processes [2]. A production-focused connection is one that outsources a portion of a business process to a supplier. Customers see the outsource as adding value (i.e. lower costs, shorter cycle times). Since others perform the activities that create the value, production-focused connections require monitoring supplier performance [3]. With the right design, an e-business connection establishes standards for supplier performance and for performance monitoring.

Engineer Adaptive Connections. Therefore, the question for software engineers is: "How can business factors and rules of engagement for an e-business connection be built and managed at all levels by both parties?" The answer is by using a Software Engineering approach and tool set that:

- produces an architecture of the business relationship,
- creates processes within the business architecture [4],
- implements an adaptable technical infrastructure by
- exploiting key properties of the Internet [5].

"Adaptable" means that all designs constantly mirror and support the business. Internet technologies have made possible the kind of Software Engineering where supply chain implementation and management can be accomplished [6]. Let us look at key properties of the Internet that enable this possibility.

2. CRITICAL PROPERTIES OF THE INTERNET

Experience #2: *When taken together, Internet technology -- cheap computers, fast networks, and almost unlimited storage -- enables adaptability to be designed into both a business process and its technical implementation.*

Exploit Internet Properties. Software engineers can exploit the Internet if they can think of the Internet in a particular way: *The Internet is a process medium* -- not an advertising medium, not a transactional medium, not a communication medium [7]. To see the Internet in this way is to make a profound shift in how to utilize it during Software Engineering . This view unlocks hidden value that can enable a business to gain operational efficiencies without being constrained by regulatory standards, or to quickly adapt to changing market conditions without being held hostage by its application portfolio. Let us see why:

Property #1: Cheap Computers. Since the advent of the ARPanet, the cost of computers has decreased to the point where a person could buy the equivalent of an early mainframe computer and put it on their desk. This is the first Internet property - that MIPS are extremely inexpensive. This property enables software engineers to potentially dedicate a whole computer to running only a portion of an entire business process. In the extreme case, a whole computer can be, and has been [4], dedicated to running just a single business process rule.

Property #2: Fast Networks. The second key property is the reality of very fast, pervasive computer networks. Today, you can go down to the local computer store and buy all the wire and components to make your own home network. Local area networks have become very fast, cheap and easy to build. They are so fast that a corporate wide area network can run an application over its network, and with proper architecture, the application performance feels almost like it is running on your very own computer. This enables software engineers to physically distribute a business process across the entire network.

dmarca@email.uophx.edu

Property #3: Unlimited Storage. The third Internet property is a dramatic increase in information storage capability accompanied by dramatic decrease in its cost. This lets a person take a company's entire process and operational history wherever he or she needs to go. The effect is that people can carry as much information as they need in order to make decisions right at the place where a business event happens (e.g. handling an insurance claim at the accident scene). Software engineers can thus concentrate on providing all the right instructions and information to support real-time decision-making in the field.

Exploit All Properties Simultaneously. However, these three properties alone are insufficient. What is most important is that all three of these technical properties of the Internet arose in roughly the same period of time. Taken together, these properties are a potent cocktail for developing highly adaptable business processes and highly adaptable technical infrastructures.

3. WEB PAGES MODULARIZE A PROCESS

Experience #3: *If a Software Engineering tool set can manage the modularization of the process to the level of one web page per process step, then generating Internet-based processes can happen directly from commercial process modeling tools.*

Isolate Process Rules. When taken together, the fundamental properties of the Internet allow software engineers to create extremely deep and modular web structures [8]. In addition, hypertext languages [9] allow software engineers to distinguish process keywords from process narrative. Therefore, traditional process modeling tools, which support decomposition, modularity and data typing, can be used to generate web-based process descriptions. In particular, since IDEF0 process modeling tools [10] can distinguish rules from data flow, they are strong tool candidates for business architecture and design. As we will see in Section 9, these tools also provide data semantic information that is critical to customizing the final user interface to support novice, intermediate and expert users of the process.

One Web Page = One Process Step. The OpenProcess™ tool itself [11] is used to translate IDEF0 models into a large number of very small web pages. Each page is a process that is viewable and "executable" via an Internet browser. This tool also links the resulting Internet-based processes to software applications whose modularity is modeled after a 4-tier logical and 3-tier physical client/server architecture [5].

One Web Page = Many Connections. Section 4 presents the OpenProcess™ tool that generates web page versions of business processes. Potentially, each of the resulting web pages can enable one or more e-business connections. Section 4 outlines a way to use the tool in conjunction with standard commercial IDEF0 tools to design adaptability into e-business connections. Section 4 uses a contrived business process to explain step-by-step tool usage, and cites Software Engineering principles that ensure adaptability. Section 5 through 9 give Software Engineering experiences that resulted from using the tool to implement two commercial quality processes: a 4-tier contract labor management program and an accounting department.

4. ARCHITECTURE IS KEY TO ADAPTABILITY

Experience #4: *Software Engineering principles that focus on architecture and well-architected designs are the elements of an approach that can produce adaptable e-business connections.*

Focus on Architecture. The goal of building adaptable e-business connections can be reached if one uses an approach that emphasizes architecture. The architecture of the business-to-business partnership must address all business factors [4]. It must adhere to the constraints imposed by the business architecture [4]. It must also adhere to the constraints imposed by the process architecture [12]. In other words, adaptable e-business connections arise when the following are created in order:

1. Business Architecture for the partnership.
2. Business Design that defines all essential policies.
3. Process Architecture designed for adaptability.
4. Process Design for the activities of the connection.
5. Technical Architecture that enables adaptability.
6. Technical Design that supports process activities.

Business Architecture defines: the boundaries among major business activities, the fundamental business policies, and the external market forces. A boundary describes the roles of partners in a supply chain and the activities each must perform. The policies establish the cycle time for the activity. External factors are external events that affect the cycle time. Customers, suppliers, substitute products, rivalry, new competition, regulatory shifts, economic cycles, etc can appear in the architecture [13].

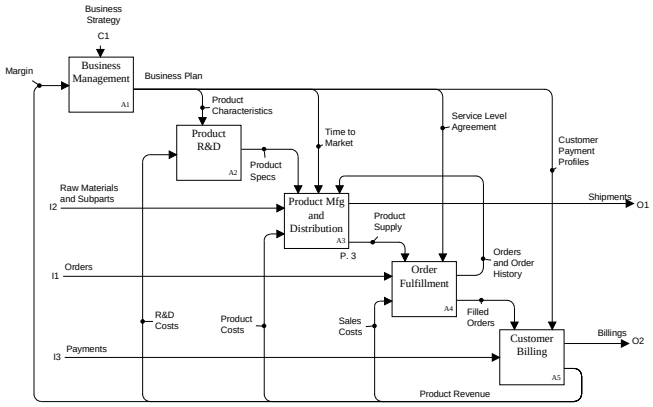


Figure 1: A Product-Oriented Business Architecture

Achieve Modularity. The value of modularity and well-defined sub-system interfaces is widely known [14]. These Software Engineering principles have great applicability when developing a business architecture. For example, Figure 1 is a business architecture for a product-oriented company of five departments. Let us assume that the executives choose to outsource "Product Manufacturing and Distribution." Box three in the diagram thus defines how both companies will accomplish this collaboration [15] via the business-to-business interface.

Business Design defines how the business responds to events. This design is created within the context of a business architecture. Thus, events and their responses are already identified as the parameters [16] of the business-to-business interface.

The architecture also provides the design with rules that govern how responses are met. These rules include service level agreements which define response times. This way, business policies are carried out in a time frame that ensures competitiveness.

Support Decision-Making. Figure 2 is a business design for a supplier of "Product Manufacturing and Distribution." Service level agreement "Time to Market" defines the time before manufacturing begins. The design shows the supplier receiving "Sales Volumes" for adjusting capacity, and "Sales Trends" for adjusting inventory. Since the implementation can deliver such process control data in near real-time speeds, this helps suppliers respond faster to changing market conditions [3].

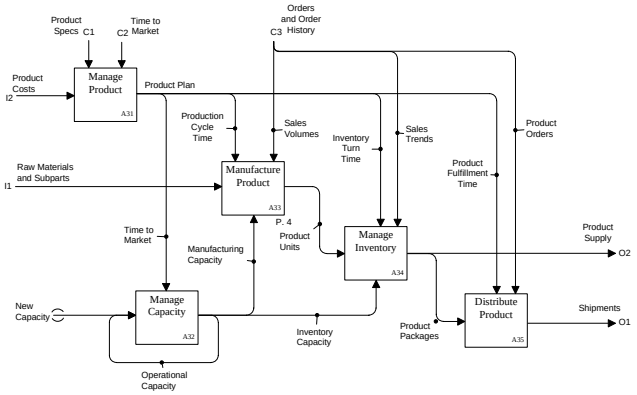


Figure 2: A Business Design with Service Levels

Process Architecture. We have defined business units and activities, made decisions as to where e-business connections will occur and decided which policies define expected performance. The connection can still fail, if either company cannot adapt its operations quickly and easily to respond to changes in business events, business cycles or market conditions - even if those changes occur elsewhere in the supply chain [17]. Adaptability is achieved when an operation has distinct steps, each fully capable of responding to a single business event.

Interconnected Architectures. A process architecture [18] is created for each box in the business design. Each is an IDEF0 model whose decomposition isolates operational steps that fully respond to one low-level event. Each architecture is adaptable because: 1) the loss of a business event means the loss of only one operating step. 2) When a service level agreement changes, the steps governed by it can be identified for alteration [19].

Define Events and Controls. Figure 3 is an architecture for Produce Detailed Parts. Its event is Product Materials and its response is Parts. Part Instructions control how parts are made. Part Production Schedule, the performance metric, is tied directly to the Time to Market SLA via arrow decompositions and transformations in the IDEF0 models. At this point, IDEF0 tool usage stops, and OpenProcess™ tool usage now begins.

Process Design. Partnering companies can fail to anticipate conflicts or lack of accountability. If an e-business connection lack a defined process, outcomes become counter-productive. This can range from gridlock or stalemate, to an unimaginative compromise where no one is really committed. To avoid such outcomes, a process design model is built to define policies, constraints, and response times for each process step. Traditionally, such models are built to work with hypercode [20], but the OpenProcess™ tool creates the model as hypercode.

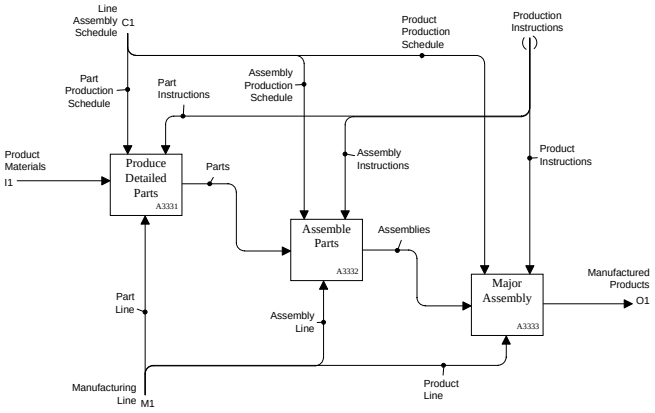


Figure 3. Process Architecture for Low Level Events

Process Hierarchy and Categories. A hierarchy is created by first loading the IDEF0 models into the OpenProcess™ tool. The hierarchy "top" comprises business units, and the "bottom" comprises process steps. A description template [10] is created, and applied to each process step during the load. The template defines aspects of the process -- description categories. These categories organize a description into small, easy-to-read-and-learn units. Some aspects of product-oriented connections are policy, exceptions, roles, quality, controls and metrics. Section 10 details how categories also act as process navigation aides.

Hyperlinks to Live Data. Figure 4 shows the tool being used after an IDEF0 model load. It identifies an internal metric [21] "Parts Per Day" for "Produce Detailed Parts." Part production indicates: 1) if the "Time to Market" SLA is being met, and 2) if production capacity is being exceeded. This exemplifies how an Internet-based process links to live operational data. This feature allows the software engineer to exploit the existing corporate knowledge base. By linking to live data, far more concise process descriptions can be written.

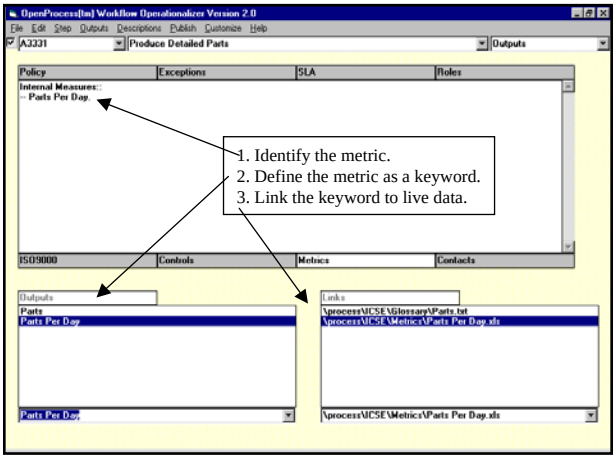


Figure 4: Defining Process for One Operational Step

5. FLEXIBLE TECHNOLOGY ARCHITECTURE

Experience #5: *The rules for designing adaptive client/server solutions can be used to create a flexible technical platform.*

Technical Architecture. Software Engineering principles for adaptable client/server architecture are well-known [5]: 1) Design business units so they do not cross the operational boundaries. 2) Place the user interface on the client machine. 3) Allocate one server for the operating rules for each business unit. 4) Create one stored procedure for each operating rule. 5) Isolate the data access logic from stored procedures. 6) Allocate one computer for both the data access logic and the database.

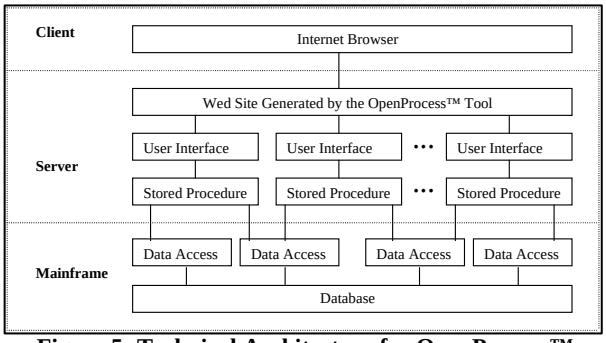


Figure 5: Technical Architecture for OpenProcess™

Highly Granular User Interface. The OpenProcess™ approach refines the client/server method by requiring the software engineer to do the following: 1) Place the resulting web pages for a business unit on the server dedicated for that business unit. 2) Create many small user interfaces -- one small user interface for just one process step. 3) Link process keywords to the small user interfaces or to files of live operational data. Figure 5 depicts the resulting architecture. Section 6 details how a large number of small user interfaces can be easily created.

Figure 6: Internet-Based Process Shown as "Cards"

Technical Design. The last step in the approach is to create small application user interfaces that display live data, and to then link process keywords to those user interfaces. Once this is done, people "run" the process by simply accessing the web site created by the tool via their Internet browser. They then simply click on a highlighted keyword that is a hyperlink to the application, repository artifact, or live data that is relevant to the process step they are currently viewing [22].

Consistent Visual Metaphor. Figure 6 shows how "Produce Detailed Parts" would appear in an Internet browser after the tool creates its web pages. Notice how the IDEF0 model has been translated into a set of "cards" - one card for each process step. Also notice how each card has one tab for each process aspect chosen in the Process Design step of the approach. Lastly, notice the "Parts Per Day" metric is encoded as a hyperlink to permit drill-down [23].

Pinpoint Data or Application GUI Usage. When users click on "Parts Per Day," they access the live data to which this link refers. In this case, it is a graph (Figure 7). The click activates a new browser that is then be used to examine production data over time (e.g. when production fails to meet the service level agreement or when it exceeded capacity). This way, users only see the application user interface, the repository artifact, or the production data relevant to the process step they are on.

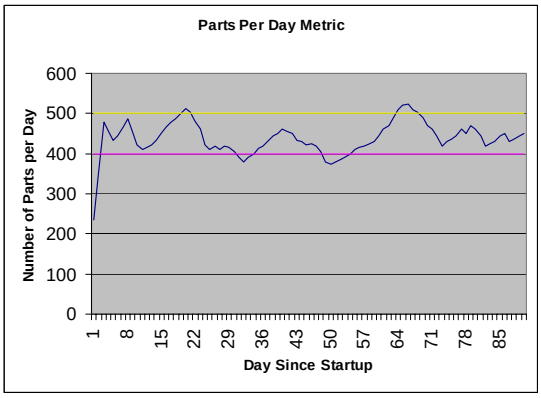


Figure 7: A Hyperlink Accesses Live Process Data

6. AN OPEN APPLICATION ARCHITECTURE

Experience #6: *When a business process is separated from its business application portfolio, the software engineer can wrap the process around each application screen, e-form and object.*

More Open Applications Are Needed. Experience during the implementation of an Internet-based process for a contract labor management program and an accounting group found traditional e-applications wanting in their architecture. Specifically, no application was “open” to the extent that an Internet-based process could hyperlink directly to a specific screen of its user interface, or to an e-form of a data object. This level of openness would likely boost productivity, because people would not need to traverse the application’s menu hierarchy to access the detailed functionality [24] for the process step they are on.

Process Wraps Around the User Interface. Applications that allow hyperlinks to detailed GUI screens or e-forms thus allow an Internet-based process to “wrap” itself around software user interfaces. This aids the software engineer because applications are written as a collection of related functions [24]. This

eliminates the concern for embedding the usage metaphor in the software. In this setting, good application design becomes:

- meaningful functional decomposition,
- compact GUI screens for each function, and
- modular create/read/write/delete code for objects [25].

7. BOOK-LIKE INDEX IS SUPERIOR TO SEARCHING

Experience #7: *If the tool set maintains process keywords and process step titles, then it can generate book-like indexes to let users to jump anywhere in the process with three mouse clicks.*

Avoid Search Functions. Experience with traditional search functions on today's web sites indicate that users are not very adept at choosing good search terms, and that search engines do not match well against search terms [26]. One way to solve this problem is to replicate the tried and true method used in physical books -- the index. A good index is a carefully designed mechanism for going right to the page of the book that meets the reader's need on a particular subject. Similarly, an index for an Internet-based process should enable the user to go to any process step with a minimal number of key clicks and no typing.

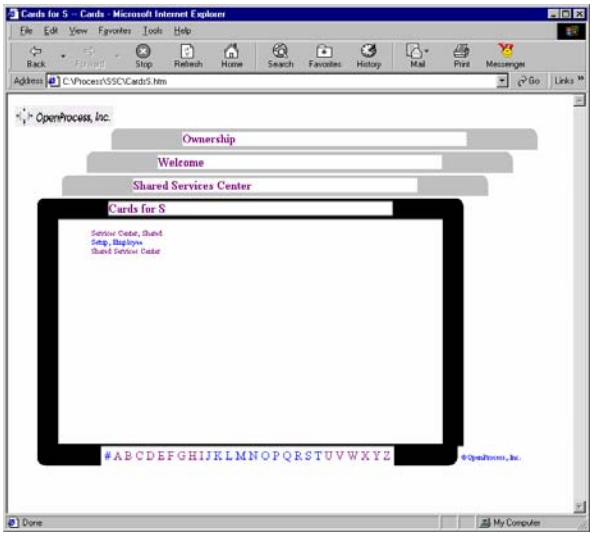
Implement Book-like Indexes. To accomplish this, the index into an Internet-based process should alphabetically present the names of process steps and keywords, and should collect these into higher level groups. Like a book index with many groups and many subjects within a group, a well-designed index of an Internet-based process needs to:

- give users insight into important aspects of the process,
- allow users to rapidly find a process step even though they may only have one word to start the search.

Use Keyword-in-Context Sorting. The OpenProcess™ tool set implements the equivalent of a book index by applying a "keyword in context sort" algorithm [27] to all keywords. It then puts all permutated keyword phrases onto web pages that correspond to the starting letter of the first word of each phrase. The algorithm is repeated using the titles of each process step. The result is a keyword index and a navigation index (Figure 8). This allows users who may remember only one word of a keyword phrase or process step to go to that keyword or step with just two mouse clicks. Experience suggests that users immediately grasp this concept, find it easy to use, and prefer it to today's search functions where one has to remember all words.

Provide Multiple Navigation Aides. The OpenProcess™ tool set also automatically generates two additional navigation aides. It generates a traditional hierarchic menu of process steps. It also generates a web page for each particular portion of a process step using description categories. In the "cards" metaphor, these categories appear as tabs (Figure 6). In a more traditional, menu-driven user interface, these categories appear as buttons or tabs above the description (Figure 9). Therefore, the tool set generates a menu, process category name buttons, a navigation index and a keyword index. The combination of these four navigational aides gives the user a comprehensive set of ways to traverse, jump and locate process steps.

Figure 8: Sample Navigation Index Page for "S"



8. CONCISE PROCESS DESCRIPTIONS

Experience #8: *When embedded within a consistent, modularized process, lists and very short sentences or paragraphs are easier to understand and follow than lengthy process narrative.*

Consistent Description Category Names. Experience with web sites and software user interfaces show that users have difficulty when data or narrative goes beyond one visible screen [28]. The solution to this problem has two parts. First, define consistent names for description categories. These names often appear at the top of web pages and have an appearance akin to a button. This breaks the process narrative into several small sections. When designed well, these number up to eight -- keeping within the rules of user comprehension and retention [28]. Sections 4 and 7 describe this mechanism in more detail.

Lists, Short Sentences, Pithy Paragraphs. After description categories are defined, write pithy narratives for each category of each process step. The following guidelines came about while writing narratives for a 4-tier contract labor management program and an accounting department:

- Bulleted lists for process step activities.
- One to three sentences for policy statements.
- Embed links to GUIs and e-forms in the narrative.
- Separate description categories for forms and templates.
- Last word in an artifact name defines its type, such as Form, Policy, Document or Template.

Implications for Contract Writing. Figure 9 shows what a process description would look like after following the above guidelines. Notice the description category tabs at the top of the page, and how very succinct policy rules were defined in the "Policy" tab. This experience uncovered important implications for contract writing [4]. An e-business connection has contractual underpinnings – two firms usually create a contract to define their business arrangement. If legal language is kept in the base agreement and terse appendices are used to define the ad-

ministration details of the e-business connection, then the contract becomes the narrative for the process description, and contract rules can then easily drive process implementation.

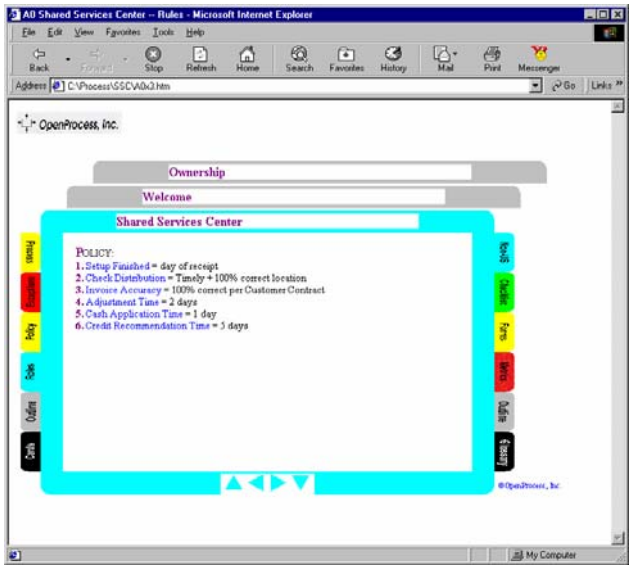


Figure 9: Concise Policy Statement of Service Levels

9. SEMANTIC KNOWLEDGE IN HYPERLINKS

Experience #9: *If the tool set associates process semantics with keywords, then standard browser customizations can alter the look-and-feel of the hyperlinks to accommodate user expertise.*

Tie Color to Semantics. Hyperlink semantics are very difficult to design [29], implement [30] and use [31]. The tool solves this problem by creating web pages from a rigorous process model having four arrow types: input, control, output and mechanism. The tool creates a hyperlink for each arrow, and lets the software engineer assign a color to each arrow type. The result is a web page that codifies data knowledge into multi-colored links: data used by a process step, a business rule, data generated by a process step, or a work instruction [32].

Browser Customization Matches User Level. Novice users (e.g. new workers) use the multi-colored links as they learn the process. Infrequent users (e.g. managers) create reminders by overriding the multi-color scheme to see all links in the same color. Expert users (e.g. day-to-day users) turn off link colors and activate the hover color. This causes process descriptions to read like plain text, yet the mouse "lights up" a keyword when it touches it. This powerful use of standard browser customization for portraying hyperlink semantic knowledge consistently across all user populations is rarely supported by current Internet applications [30] or by current workflow technology [15].

10. PROCESS BRANDING BECOMES FLAWLESS

Experience #10: *If the tool set separates the process content from its brand (i.e. its look-and-feel), then each outsourced part of the process can easily adopt the brand of the supplier.*

Separate Content from Appearance. Software Engineering has well-known and robust principles for separating the user interface from application functionality [24]. These principles translate directly to Internet-based processes. Experience has shown that it takes straightforward engineering to separate the content of a process (e.g. process steps, rules, links) from the look-and-feel of that process (e.g. menu location and behavior, selective disclosure [28] of process tabs). This is a critical principle to apply if a business process is to be dynamically and selectively opened or closed to customers or business partners.

Create Look-and-Feel Templates. The OpenProcess™ tool set implements this principle of separating process content from its appearance. The definition of process steps, rules, links, etc. are maintained separately from the look-and-feel template [33]. This way, the software engineer can define a template for the company and utilize a different template for each outsourcing arrangement. The result is a single process definition that potentially can be displayed differently for each supplier that is implementing a portion of the whole process.

Enable Process Branding. Look-and-feel templates enable process branding. That is, regardless of who defines the process, the firm that executes the process can make its own brand visible to the user [34]. Of course, the outsourcing agreement may not permit such branding, but if it does, the technology can support it. Example: Staples, Inc. lets customers create business cards. Selecting this service is done within the Staples brand. Then, the firm that handles the business card creation, makes its brand is made visible to the Staples customer. This way, the customer is aware of who is providing the actual service.

Isolate the Brand Design. To visually see how the look-and-feel can change without affecting the content of the process, let us return to the paper's running example of "Product Manufacturing and Distribution." In Section 5, we used the user interface metaphor of "cards" to present the process and give the user a way to navigate. By choosing a different template, that look-and-feel is replaced with one more akin to the menu-driven user interfaces very common on today's web sites. Figure 10 shows the result of applying a different branding template.

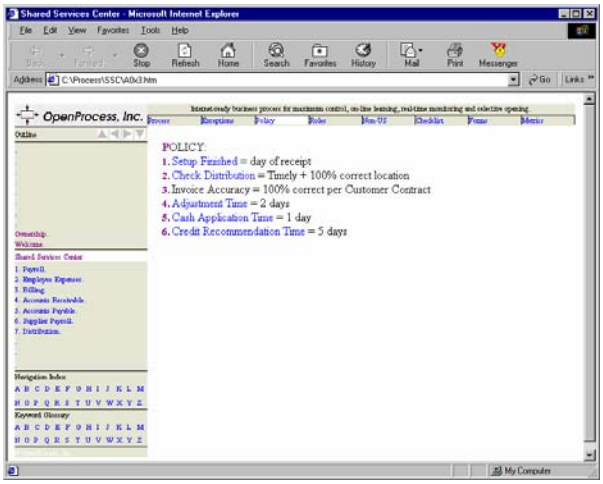


Figure 10: Menu Organization of the Process

11. SUMMARY AND CONCLUSION

Business Adaptability. The experiences given in this paper give evidence for Internet-based business processes being a critical component in a company's operations. Internet technology provides the potential for making an e-business connection to a customer or supplier at any point in the business process. This has powerful implications for a business' ability to adapt to changing markets and to take advantage of new suppliers.

Interconnected Architectures. This potential can be realized if the Internet-based business process is highly modular, implements one process step with one web page, and isolates business rules from data flow. Architecture is key – process and technology architectures are both designed within the business architecture that is designed to respond to external events.

Process-centric User Interfaces. At the technology level, Internet-based business processes must be highly modular and granular. This permits them to link to low-level GUI screens and e-forms. Designed in this way, processes can "wrap" themselves around application functionality and data object methods. The significant implication is that applications will then migrate to collections of closely related collections of small functions, while user interfaces will migrate to process-centric metaphors.

Consistent, Concise and Easy-to-Use. When designed well, Internet-based processes have several characteristics:

- process branding at each e-business connection point,
- consistent descriptions and process keyword names,
- pithy narratives and rules,
- colored keywords embedded in descriptions,
- book-like indexes for quick searching,
- multiple navigation aides on every web page,
- browser customizations accommodate different user levels.

Conclusion. Branding, personalization, conciseness, process-centric user interfaces, interconnected architectures, and business adaptability are all possible when Internet-based business processes are designed according to the experiences given in this paper. Business process management and e-business process modeling are taking on importance for researchers and practitioners. The insights gained while using the OpenProcess™ tool set to implement two industry quality Internet-based business processes are germane to these fields.

12. REFERENCES

[1] Gomes-Cassares, Benjamin, **The Alliance Revolution**, Harvard University Press, 1998.

[2] Moore, G., **Crossing the Chasm**, Harper Business, 1999.

[3] Poirier, Charles C., **Advanced Supply Chain Management**, Publishers' Group West, 1999.

[4] Marca, D., Perdue, B., "Business-to-Business Connection" **Software Eng Technical Council**, Vol 15, No 3, 1997.

[5] Gold-Bernstein, B., Marca, D. **Designing Client/Server Systems**, Prentice-Hall, 1997.

[6] Kalakota, R., Whinston, A., **Frontiers of Electronic Commerce**, Addison-Wesley, 1996.

[7] Tolhurst, W., et. al. **Using the Internet**, Que Corp, 1994.

[8] Singh, M. "Deep Web Structure" **IEEE Internet Computing**, September-October 2002.

[9] Ceri, S., et. al, "Conceptual Modeling of data-Intensive Web Applications" **IEEE Internet Computer**, Jul-Aug 2002.

[10] Marca, D., McGowan, C., **IDEF0: Business Process and Enterprise Modeling**, Eclectic Solutions, 1992.

[11] Marca, D., Perdue, P. "A Software Engineering Approach and Tool Set for Developing Internet Applications" in **22nd International Conf On Software Engineering**, 2000.

[12] Smythe, C., **Internetworking: Designing the Right Architectures**, Addison-Wesley, 1995.

[13] Jayachandra, Y., **Re-Engineering the Networked Enterprise**, McGraw-Hill, 1994.

[14] Britton, K., et al, "A Procedure for Designing Abstract Interfaces for Device Interface Modules" in **5th International Conf On Software Engineering**, Mar 1981.

[15] Marca, D. "Requirements Elicitation and Specification for CSCW" in **Software Engineering Productivity Handbook**, Keyes J., McGraw-Hill, 1992.

[16] Beck, K., et al, "Industrial Experience with Design Patterns" in **18th Int Conf On Software Eng**, Mar 1996.

[17] Taylor, David A., **Business Engineering with Object Technology**, John Wiley & Sons, Inc., 1995.

[18] Hitomi, A., et al, "A Process System Infrastructure" in **19th International Conf On Software Eng**, May 1998.

[19] Leonhardt, U., et al, "Decentralised Process Enactment in a Multi-Perspective Development Environment" in **17th International Conf On Software Eng**, April 1995.

[20] Kaiser, G., et. al., "An Architecture for WWW-based Hypercode Envs" **19th Int Conf On Software Eng**, 1997.

[21] Randall, R., **Randall's Practical Guide to ISO9000**, Addison-Wesley, 1995.

[22] Ackerman, M., Halverson C., "Reexamining Organizational Memory" **CACM**, Vol 43, No 1, 2000.

[23] Jern, M., "Information Drill-Down Using Web Tools" in "Virtual Worlds on the Internet" by Vince, J., Earnshaw, R., **IEEE Computer Society Press**, 1998.

[24] Alencar, P., et. al. "A Logical Theory of Interfaces and Objects" **IEEE Trans on Software Eng**, June 2002.

[25] Cheng, B., Wang, E., "Formalizing and Integrating the Dynamic Model for Object-Oriented Modeling" **IEEE Trans On Software Engineering** , August 2002.

[26] Hearst, M., et. al. "Finding the Flow in Web Site Search" **CACM**, September 2002.

[27] Knuth, D, **Sorting and Searching**, Addison-Wesley, 1973.

[28] Nielsen, J., **Usability Engineering**, Academic Press, 1993.

[29] Jourdan, M., et al, "Authoring Techniques for Temporal Scenarios of Multimedia Documents" in **Handbook of Internet and Multimedia**, IEEE Press, 1999.

[30] Wynblatt, M., et al, "Multimedia Applications on the Internet" **Handbook of Internet and Multimedia**, 1999.

[31] Trigg, R., "Tools for Communicating in a Hypertext Environment" in **Conference on CSCW**, September 1988.

[32] Jackson, M., Zave, P., "Deriving Specs from Reqts" **17th International Conf On Software Engineering**, Apr 1995.

[33] Billus, D., et. al., "Adaptive Interfaces for Ubiquitous Web Access" **CACM**, May 2002.

[34] Ardissono, L., et. al. "Personalization in Business-to-Customer Interaction" **CACM**, May 2002.