Capturing and Shaping Shifting Requirements using XML and XSLT: A Field Study

Charmayne Cullom
University of Northern Colorado

Tod Sedbrook
University of Northern Colorado

Follow this and additional works at: http://scholarworks.lib.csusb.edu/jitim

Part of the Management Information Systems Commons

Recommended Citation
Available at: http://scholarworks.lib.csusb.edu/jitim/vol16/iss1/6

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Journal of International Technology and Information Management by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
Capturing and Shaping Shifting Requirements using XML and XSLT: A Field Study

Charmayne Cullom
Tod Sedbrook
University of Northern Colorado

ABSTRACT

This paper explores Extensible Mark-up Language (XML) and Extensible Stylesheet Language Transformations (XSLT) for authoring, presenting and managing system requirements. A field study is presented that explores the influence of XML schema and XSLT rendering and modeling templates on stakeholder communications. The study is of an e-commerce project where an evolving business model and changing partnerships forced the requirements team to continually adapt XML and XSLT tools to capture requirements.

Coding procedures categorized resulting repositories of XML documents, XML schema, XSLT templates, stakeholder interviews, field notes, e-mails, and business documents. Qualitative techniques are applied to derive a model summarizing the influence of XML schema, and XSLT rendering and modeling templates. Implications for researchers and practitioners are discussed, including how XML tools support stakeholders by customizing presentations, assisting negotiations and enhancing traceability.

INTRODUCTION

Over half of information systems projects do not meet the expectations or total needs of the stakeholders according to a variety of sources. Inefficient communication is often noted as a primary culprit. Within the communication context, deficient capturing of changing requirements and lack of presentation of requirements in a shape or form compatible with the audience’s frame of reference contribute highly to the communication failures. Capturing and shaping these changing requirements present an immense challenge to the requirement analyst.

In Ovid’s epic poem, Metamorphoses, Peleus (the father of Achilles) finds the shape shifter Thetis asleep in a sea-cave and binds her with every noose and snare. Held fast, she changes into a bird, a tree, a tigress, and a hundred different forms and tells a hundred different stories. Peleus’s need to capture Thetis mirrors a requirement analyst’s need to capture a system’s functionality. Like Thetis, a system analyst believes the requirements are captured but appear in multiple forms to stakeholders.

Analysts commonly capture system requirements through interactions with stakeholders to develop use cases. While use cases capture regular, standard stories, there are roles that are not easily attached to the system. These roles and functions are considered to be detached. Because detached stories do not describe a regular, standard role, such stories must be bound and compiled into many forms to assist clients, developers, managers, and other system stakeholders integrate, inspect, and validate requirements. Detached stories are the changing and shifting requirements of any system. These stories typically represent the requirements that are difficult to capture and shape. Like Thetis, these stories may change form with the function or role being hidden. Form follows function.

This research describes application of Extensible Mark-up Language (XML) to facilitate categorization and customization of detached use case and related requirement artifacts. Extensible Stylesheet Language Transformations (XSLT) templates then capture, bind and transform XML source documents to provide customized and crosscutting views of evolving system requirements. Capturing and documenting changing system requirements using XML with XSLT templates becomes the Thetis for requirements analysts.
The next section reviews literature concerning usage of templates for requirements gathering and the utility of XML templates to structure requirements. The attributes of XML making it a suitable vehicle for template construct is addressed. Subsequent sections explore application of XML and XSLT templates in the context of a qualitative field study. The analysis section explores the field studies through a model summarizing the influence of XML schema, and XSLT rendering and modeling templates on stakeholder communications. We conclude by discussing implications of the research for practicing analysts.

**CAPTURING REQUIREMENTS WITH TEMPLATES**

Requirements describe the behaviors or services desired to be provided by a system. Behavior models specify actions, test cases manage verification, and change cases capture business change. A complete set of use cases should define all behaviors of the system establishing the system boundary. Use cases are bundles of commingled stakeholder viewpoints.

**Stakeholders Views of Requirements**

Stakeholders require sets of customized models that closely match their requirements concerns. Presentation of requirements adapted to particular stakeholder viewpoints improves communication and coordination. Poorly shaped requirements models lack sensitivity to individual concerns which causes misinterpretations resulting in immature designs and implementations.

Drawing out and inspecting logical inconsistency among artifacts improves requirements traceability. The formation of flexible combinations of use cases for stakeholders to highlight and zoom in on the impact of changes on business rules and operations are critical to discussions. (Nuseibeh & Easterbrook, 2000; Zhang & Xie, 2001)

**Using Requirement Templates to Facilitate Communication**

Informal models such as diagrams and use case stories simplify communication among clients and analysts. Analysts then study the resulting informal narratives to develop formal models to cleanly communicate detailed requirements. Improvised conferences and ad hoc methods to translate informal models into formal requirements documents introduce inconsistencies and ambiguities (Dawson & Swatman, 1999) as individual analysts impart interpretations on the facets of the models. One mechanism for controlling and structuring requirements elicited by use cases is the formulation of templates. Templates provide a consistent approach to the solicitation of requirements from stakeholders.

**Guiding the Elicitation Process**

Use case templates direct the creation of use cases by enforcing content and style patterns to assure consistency, correctness and completeness of requirement specifications. Templates focus an analyst’s attention on actor goals, flows, variations and triggers, and consider measures of relevance, accuracy, and ambiguity in use cases (Anda & Sjoberg, 2002).

Templates assist in identifying goals and in capturing likely scenarios, alternatives and potential failures. Requirements templates help organize sets of specifications by summarizing objectives, boundaries, and domain objects (Adolph & Bramble, 2003; Armour & Miller, 2000). Guided techniques capture higher quality knowledge within knowledge-based diagnosis, debugging and interpretation. Use case templates make it easier to identify detached cases during the formal modeling activities as the content is more easily comparable.

It is common to apply a word processor to document requirements gathered within the template structure. But resulting text documents are difficult to change, query and combine into summary reports. Integrating word processors with databases supports traceability by making it easy to create links among requirements artifacts (Gottesdiener, 2003). Even with traditional database tools, it is still difficult to customize documents for groups of clients, managers and developers. Reporting tools, based on XML, are needed to supplement databases to improve
readability by personalizing documents and other artifacts for stakeholder groups (Davies et. al., 2001). The efficacy of XML tools for capturing requirements and reporting is discussed next.

**XML BASED TOOLS**

Synergy results from requirement templates that construct intuitive representations that conform to a stakeholder’s mental models or frame of reference. XML based technologies provide opportunities for creating and adapting source documents to serve the needs of different stakeholders (Weber, Kienle, & Muller, 2002). XML allows efficiently sharing and managing document versions. XML links document revisions and provides services to rebuild and present document changes. Self-describing XML documents through metadata flexibly represent elements and attributes to express rich semantic relationships (Chien, Tsotras, & Zaniolo, 2001; Kappel, Kpasammer, & Retschitzegger, 2001).

**Logical Structure Difference**

XML data is described by its logical structure not the physical structure as a word processing document. Therefore, the emphasis is upon what the pieces of information are, and not the physical items (Opara, 2004). XML provides reuse without multiple storage of the same piece of information as well as the ability to identify and manipulate individual information components. (Wagner, Hilken, & Chung, 2004).

XML easily integrates with common storage, documentation, and modeling tools. XML schema promotes flexible standards for exchanging documents among stakeholders by defining data types, common names, attributes and relationships. XML templates offer an opportunity for augmenting requirement determination tasks by providing semantically rich representations and presentations (Frankel, 2003; Sutcliffe, 2002).

The following presents an interpretive field study of a requirement determination project. This investigation explores the organization’s usage of XML and XSLT to provide multiple interpretations and support richer views for stakeholders in requirements analysis.

**FIELD STUDY METHODOLOGY**

The name and details describing the participating organization which might reveal the actual company are fictionalized to assure anonymity. For purpose of this research, we have titled the organization CYAN. The field study explored CYAN’s project to gather requirements for development of a web based e-commerce system to market nearly new vehicles for a group of international partners.

**Qualitative Research Approach**

The research approach is one deemed to be interpretive and action oriented as we participated in creating, evolving and interpreting XML requirement artifacts (Klein et. al., 2002). One of the authors directly participated with and observed development teams as they developed schema, captured requirements in XML source documents and developed XSLT presentation tools to render use cases and other requirements artifacts.

Qualitative coding procedures categorized repositories of XML documents, XML schema, XSLT templates, stakeholder interviews, field notes, e-mails, and business documents to determine the impact of XML and XSLT on the requirements process. Interviews and data collection focused on gathering and categorizing qualitative reactions of stakeholders and classifying attributes of requirement artifacts. Multiple interviews were conducted with project managers, business managers, clients, technical projects leads, and system architects to assess utility of XML schema and resulting models for capturing and communicating requirements. Requirements inputs included business rules, domain objects, and the like derived from stakeholder interviews and business documents.
Field Study Context

CYAN, a medium sized U.S. based consulting company, is the development partner responsible for designing and implementing an internationalized internet commerce site to sell nearly new vehicles. The managing partner is located in the Belgium with the responsibility to refine the business model and coordinate partner negotiations. CYAN is responsible for requirements analysis, design and implementation and contracts directly with the Belgium partner. CYAN’s requirements team consists of architects, technical leads, project managers and business relationship managers. Belgium-based client participants include senior management representatives, business leads and customer service managers.

The overall project and business characteristics can be summarized as:
► Multi national with virtual teams
► Multi company involvement with many partners
► Multi national customers for products
► Products (nearly new automobiles) located multi nationally
► Multi national banking regulations to be addressed
► Dynamic development environment with rapidly changing requirements.

Characteristics of Project

The business model includes providing facilities for assisting customers in researching vehicles, selecting vehicles, configuring options, processing trade-ins, obtaining warranties and financing, and scheduling delivery. The project participants include numerous international partners representing banks, data suppliers, vehicle suppliers, service companies and marketing associations.

In the initial requirements phase, the requirements team and clients used standard word processors with requirements templates and email to develop and review use cases. U.S. based team members also maintained a "war room" or library containing whiteboard illustrations, page and user interface mockups and requirements documentation. Project managers used Microsoft Project to capture requirements timelines. Technical leads developed and applied UML to develop static and dynamic analysis models. Several international visits were part of the requirements gathering to ensure communication.

A core set of about seventy use cases were documented during this period of requirements gathering. CYAN began design and development based upon these initial seventy use cases. Table 1 presents a sampling listing of the original use cases.

Table 1: Listing of Selected Original Use Cases.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddVehicleToRealInventory</td>
<td>This use case describes how a Supplier adds a vehicle to the Real Inventory.</td>
</tr>
<tr>
<td>CalculateVehiclePrice</td>
<td>Calculate a vehicle price based on the configuration. The trade-in value for a vehicle is factored into the quoted price.</td>
</tr>
<tr>
<td>CalculateTax</td>
<td>This use case describes the mechanism for finding the Tax Rates for a particular region and using them to calculate tax on a purchase</td>
</tr>
<tr>
<td>DeleteSoldVehicleFromRealInventory</td>
<td>This use case describes how a Supplier deletes a vehicle that has been sold from the Real Inventory</td>
</tr>
</tbody>
</table>
However, within a month of the initial development process, the design team had identified 186 outstanding requirements questions and issues. Project managers, team leaders, client relationship managers and architects also each maintained individual lists of outstanding issues and questions in various spreadsheet and text formats. Then, in mid development stream, the Belgium partner proposed a major change to the business model that reflected new banking partnership agreements and revised policies for English and German markets.

The evolving business model and changing partnerships forced the requirements team to continually create new use cases and refine existing documents. The numbers of detached stories escalated and were difficult to bind to any of the initial seventy standard use cases. Using traditional methods of document management, it became difficult to track the ever increasing changes via the large number and versions of documents depicting the outstanding requirements and detached stories identifying issues to be addressed.

Table 2 provides a sampling of the use case involving virtual inventory as distinguished from real inventory. This use case describes how a Supplier adds a vehicle to the virtual inventory. When reviewing the template showing the sequence of events, it becomes apparent that the introduction of virtual inventory was not easily attached to the original requirements understandings. The virtual inventory scenario reflected the new business partners to fast track vehicles from the rental car fleet before the vehicle was actually released to the real inventory.

Table 2: Example of Detached Use Case Add Vehicle To Virtual Inventory.

<table>
<thead>
<tr>
<th>Use Case Sequence of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The Supplier logs onto the system.</td>
</tr>
<tr>
<td>2. The supplier selects “Manage Virtual Inventory” from the Supplier’s page.</td>
</tr>
<tr>
<td>3. The Manage Virtual Inventory page allows the Supplier to view, add, or delete vehicles from the current list.</td>
</tr>
<tr>
<td>4. The Supplier selects the “Add” option.</td>
</tr>
<tr>
<td>5. The system request information on the vehicle. The information requested includes:</td>
</tr>
<tr>
<td>• Make (choice box with fixed list)</td>
</tr>
<tr>
<td>• Model (choice box with fixed list, context sensitive to the Make)</td>
</tr>
<tr>
<td>• Year (choice box with fixed list)</td>
</tr>
<tr>
<td>• Style (choice box with fixed list)</td>
</tr>
<tr>
<td>• Color (choice box with fixed list)</td>
</tr>
<tr>
<td>• Options</td>
</tr>
<tr>
<td>➢ Air Conditioning (check box)</td>
</tr>
</tbody>
</table>
| ➢ WHAT OTHER OPTIONS NEED TO BE LISTED???
| • Vehicle Supplier (defaulted, based on Supplier login) |
| • WHAT OTHER INFORMATION NEEDS TO BE LISTED???
| 6. The system validates the information. |
| 7. The system adds the vehicle information to the Virtual Inventory database, classifying the vehicle as virtual. |

Another example of detached stories was the concept of test drive. This story was extremely difficult as each of the stakeholders required different views. Table 3 depicts some of the variety of participants and the complexity of the events included in the use case of Set-upTestDrive. This example of a use case has six participants each requiring a different view because of interests. The detached story resulted from the new financial business partner with a variety of options for financing payment of the vehicle as well as the change in specifications that the search results should yield “near” or “close” matches to the prospective customer’s criteria. Criteria changed for the test drive scenario when the business partners determined it was not necessary for the customer to test drive the actual vehicle.
he/she might purchase but rather a similar model of vehicle. Additional modifications to the scenario were added when the business partners removed the restriction of the vehicle location being in close proximity to the customer.

Table 3: Detached Story Use Case Example of Set-up Test Drive.

<table>
<thead>
<tr>
<th>Preconditions:</th>
</tr>
</thead>
</table>
| The customer has configured a payment for a vehicle.  
| The customer has registered and provided information to the Customer Data System  
| The customer has reserved a vehicle and notified the CSR to search real inventory.  |

<table>
<thead>
<tr>
<th>Participants:</th>
</tr>
</thead>
</table>
| Customer  
| Customer Service Representative (CSR)  
| Workflow Management System (WFMS)  
| Vehicle Supplier  
| Personal Automotive Centre (PAC)  
| Vehicle Supplier  |

<table>
<thead>
<tr>
<th>Sequence of Events:</th>
</tr>
</thead>
</table>
| 1. The CSR has updated the Customer’s PAC.  
| 2. The Customer review and selects a vehicle from the PAC.  
| 3. The Customer confirms the financing options for the vehicle.  
| 4. The Customer completes a credit application (use case “Credit Application”)  
| 5. The Customer requests a test drive.  
| 6. The System present test-drive information including costs, delivery options, and mileage restrictions and refund policies.  
| 7. The Customer confirms the test drive request.  
| 8. The System presents the Customer a schedule form requesting  
| 9. Duration of test drive – weekend or weekday (Monday – Friday)  
| 10. Date of test drive  
| 11. Hand over method (pick-up or delivery)  
| 12. If the Customer request pick-up, the System requests the desired location for pick-up and the Customer may create a map to the pick-up location.  
| 13. If the Customer requests delivery, the System requests that the Customer supply a location for delivering the vehicle.  
| 14. The System requests Credit Card information – use case “Collect Credit Card Information”.  
| 15. The System request the Customer provides a license and license issuer and requests that the Customer confirm they are over 23 years of age.  
| 16. The Customer is alerted that the CSR phone with a test drive confirmation.  
| 17. The System updates the WFMS and Customer PAC.  
| 18. The CCCP processes the credit card and notifies the CSR of the funds capture – use case functional core “Capture a Transactions”  
| 19. The System requests that the CSR confirm the schedule of the test drive.  
| 20. The CSR contact the Vehicle Supplier to check the status of the Customer’s requested vehicle.  
| 21. If the test drive funds have been captured, the CSR reserves a test drive vehicle at the Supplier.  
| 22. The CSR updates the Customers PAC with the confirmation and notifies the Customer of the reserved test-drive vehicle by email.  
| 23. The Customer completes the test drive.  
| 24. The Supplier notifies the WFMS that the Customer has completed the test drive.  
| 25. The WFMS notifies the CSR that the Customer has completed the test drive.  
| 26. The CSR surveys the Customer and enters survey results in the Customer Data System.  
| 27. The CSR notifies the WFMS that the test drive is completed.  |
The partners then introduced a further detached scenario titled “FastTrack” to offer specials on automobiles in either the virtual inventory or real inventory which the partner companies desired to move quickly to sold. Each supplier could provide his/her own special rules or requirements for the FastTrack specials. The need for better control and collaboration of requirements analysis and verification rendered word process based templates inefficient and ineffective tools.

At this point XML and XSLT tools were introduced as innovative representations to capture and present requirements for the ever morphing business model. Team members developed XML template schema, and sets of authoring and XSLT presentation tools to capture changes after intensive interviews and discussions with stakeholders concerning each group’s particular informational need. The following relates field study events to the influence model to illustrate conditions bearing on XML technology for requirements determination. The purpose is to present a structured example of CYAN’s experiences with XML and XSLT tools. We consider each representational component as it influences customization, negotiation and traceability.

**APPLICATIONS of XML And XSLT SCHEMA TEMPLATES**

Semantic networks mirror the organization of human memory to flexibly evolve connections, track changes, articulate relationships and manage inconsistency. Communication among stakeholders is critical to organization of the semantic networks. Communication is also critical for negotiation and traceability.

**Qualitative Analysis**

Field study analysis began by categorizing stakeholder events. (Jacobson, et al., 1992;) within classifications of elicitation, modeling, and verification. Discussions with stakeholders led to the identification of communication attributes deemed to be important to the capture of requirements. These attributes were refined into categories of customization, negotiation, and traceability. Analysts of the requirements team developed XSLT presentation templates to meet the needs of stakeholders. Based upon feedback and experiences it became apparent that XML schema and XSLT templates influenced stakeholder communication. Discussion of the categories of communication and examples of the influence the XML and XSLT renderings are presented next.

**XML Schemas**

An XML schema defines the semantic contents of an XML document by specifying ordered hierarchies of elements, attributes and data types. An XML schema and its associated XML document instances match a semantic network’s ability to capture, document, reorganize and generalize requirements.

**Customization.** Basic sets of XML schema elements were agreed upon for FastTrack. Stakeholders for the XML schema were the analysts. Analysts customized XML schemas to reflect their requirement needs. One analyst added schema elements and attributes to capture action items and priorities. Another analyst updated schema to capture structured questions and answers, and another included new schema sections for measuring quality. Although analysts customized schema for their individual use, the self-describing nature of XML facilitated merging common elements across sets of XML source documents.

**Negotiation.** At CYAN, XML schemas supported semantic organization by identifying elements for describing use cases, business and technical goals, assumptions, rules, scenario steps, alternatives and issues. Interface schema detailed guides for creating HTML mock-ups of interface layouts and controls. Workflow schema named a hierarchy of lifecycles, stages, tasks and commands. Rule schema defined logical structures of initial facts, premises, actions and conclusions.

Constructing schema frameworks involved meta-discussions for organizing requirement artifacts according to semantic abstractions. CYAN analyst bartering resulted in new schema categories to reclassify individual use cases according to partner tiers. XML schema provided a coherent framework to focus negotiations and organize artifact content.
Traceability. The resulting XML schemas served as checklists to trace gaps and direct interviews with stakeholders. XML authoring tools processed the reusable schema to provide fields for editing and organizing XML documents. Analysts then applied XML schema templates to guide interviews and develop XML document instances.

The XML schema identified information gaps and prompted follow-up questions. During interview breaks, analysts added new XML source documents to previous versions and XSLT templates then formatted updated specifications for immediate interviewee review. An example of the XML schema for FastTrack is presented in figure 1. The entire schema is not presented for both brevity and confidentiality concerns. The entire FastTrack scenario includes over forty pages of scenarios details and concerns of stakeholders.

XSLT Rendering Templates

XML source document acted as raw material to XSLT rendering templates that output customized use cases, domain object models, workflows sequences and logic based rules. XSLT contained stakeholder specific formatting instructions to transform the content of XML documents. XSLT also contained query logic to merge sets of source documents to form cross-cutting summaries and indexes.

Customization. XSLT rendering templates produced documents customized for each stakeholder group. XSLT rendering templates promoted cognitive accessibility by summarizing, highlighting, clustering, and reordering XML source documents. Customized content, text, graphics and layout presented a common look and feel for stakeholder groups.

Figure 1: Extract of XML Schema for FastTrack.
Negotiation. Use cases, represented as XML source documents, were rendered by XSLT templates for document production. CYAN applied a medley of rendered documents to support negotiation including crosscutting views of business goals, workflows and actor scenarios, action sequences, alternatives, business rules and outstanding issues. These summaries allowed CYAN managers to identify inconsistencies and conflicts across sets of related use cases. Managers inspected collections of business rules and found missing logic for returning a contracted car within the FastTrack scenario.

Highlighting and color coding focused managers on client priorities and partner concerns. Summary outlines, supplemented with hypertext indexes, allowed managers to drill down to use case details. As managers reviewed summaries of use cases, analysts captured and added management questions to XML source documents. XSLT presentation templates then rendered documents highlighting management issues for follow-up responses. Discussions with partners, managers and architects suggested customizing requirement documents aided negotiations by improving comprehension, identifying conflicts, and supporting cross-referencing.

Traceability. Separating XML source data from rendering templates supported creation of customized sets of views for tracing concerns of external banking partners, data supplier and international partners. XSLT provided the ability to transform XML source documents into arbitrary structures. XSLT templates converted XML documents to standard HTML. XSLT templates may also transformed XML documents to produce partner specific XSLT templates.

An XSLT transformation converts an input XML document into a new XML or HTML document. The XSLT template that controls this transformation and restructuring is itself a valid XML document. By designing a controlling template it is possible to transform an initial XML document containing partner specific formats to produce a customized set of XSLT templates.

Meta-XSLT templates defined the common visual properties across a set of partner documents including table layouts, color, backgrounds, business logos, and language. Meta-XSLT templates processed XML resources to produce sets of country and partner specific XLST style sheets that were, in turn, applied to format use-cases. Partners reviewed use cases presented in documents customized according to their functional and regional concerns (Peterson & Kim, 2000).

The XSLT templates rendered the XML use-case data into HTML comment forms that were distributed by e-mail, intranets and virtual private networks to simplify requirement reviews among global partners. Returned comments were traced to the original XML source documents that were in turn updated and rendered for follow-up reviews.

DISCUSSION OF XML AND XSLT INFLUENCE MODEL

Figure 2 graphically presents the resulting influence model grounding the relationships of XML schema and XSLT templates within stakeholder communication structures.
XML/XSLT Influence Model

The outer ring of the influence model considers relationships among XML schema, XSLT presentation and XSLT modeling templates as they influence requirement team communications. Transformation relationships connect XML schema instances with XSLT presentation templates that reformat and customize requirements documents. Adaptive relationships relate XSLT rendering templates to XSLT modeling templates which format interactive models including domain models, user interfaces and simulations. Prototyping relationships relate XML schema instances to XSLT model templates to render changes in requirement prototypes.

The inner ring considers the effect of stakeholder communication as influenced by outer ring representations. Key communication attributes for stakeholders include customization, negotiation and traceability. Customization is supported by XSLT rendering templates targeted toward stakeholders’ concerns. Negotiation is supported through stakeholder effects to cooperatively construct and review XML schema, XSLT generated review forms, prototypes and interaction models. Traceability is supported by updating, merging, indexing and summarizing XML source documents though XSLT tools that connect schemas, formatting and modeling efforts.

Renderings influenced communication as partners were able to visualize via color coding and highlighting areas of conflicts of requirement specifications. Gaps were also more easily identified with the XSLT renderings of the XML schema applicable to each stakeholder. Concerns, gaps, and conflicts were more easily monitored and updated using the XML and XSLT tools.

Each stakeholder had her/his own agenda for FastTrack. XML schema rendered with XSLT provided partners a neutral view of the requirements scenario. Such renderings were observed to facilitate negotiation to resolve conflicts and gaps.

Members of management and project teams fully interacted with each other using the documents and views rendered by XSLT to negotiate changes and modifications. Management was able to view the interpretation of the requirements in a meaningful context and thus could clarify the requirements and close any gaps in understandings the development analysts had. By rendering the requirements in a view familiar and comfortable to the management teams and partners, the discussion of requirements was conducted in a non threatening environment. Management was not overwhelmed with technical jargon or diagrams so that the focus of the discussions could be on the requirements. Additionally, using XML schema and XSLT rendering techniques, the analyst team members could quickly make modifications and track changes to keep the discussion timely and relevant. Thus, the XSLT presentation greatly facilitated and influenced the behaviors of the management and partner stakeholders.
SUMMARY and CONCLUSIONS

Studies have suggested that requirement determination depends on information richness provided by communication techniques. Tools and techniques are objectively studied as they exert a causal influence on human information processing (Browne & Rogich, 2001). Recent theories however extend this perspective by arguing research should include critical reflections of participants as they interpret and engage in requirements determination. These theories argue that a system’s features can not be deduced from the separate analysis of a tool’s properties and participants; instead they have to be explained from causal interactions and relations among those elements.

Research Implications

As field study participants gained experienced with XML tools, their actions and interpretations changed to reveal new requirement directions. The study findings suggest XML representations improve requirement determination by organizing and creating representations that aid stakeholders in presenting, tracing and negotiating volatile business changes.

XML schema and meta-XSLT templates support abstraction and synthesis by providing tools to abstract and clarify requirement artifacts. Such linkages support the theories of meta-improvement where participants re-evaluate process knowledge and focus on ways to improve stakeholder understanding (Bieber et al., 2002).

This study complements a design theory that accounts for emergent properties by engaging users with flexible tools that promote customization, negotiation and traceability as stakeholders creatively adapt requirements. XML and XSLT presentation and modeling tools manage emergent artifacts to support a process that encourage participants to “step-outside” and critically evaluate the rationality of existing designs (Markus, Majchrzak & Gasser, 2002).

Organizations generate, validate and revise requirements within cultural and organizational frames. This study grounds requirement representations within the context of stakeholder communications. The findings of this study are consistent with theories of frame salience that recommend that requirements determination explicitly acknowledge socio-cognitive effects. Recent requirement models however lack explicit roles for representational support (Browne & Rogich, 2001; Davidson, 2002; Sutcliffe, 2002). This research contributes an influence model that articulates a central role for external representational, presentational and modeling forms as they impact stakeholder communications.

Practical Implications from CYAN Experiences

The organization experimented with XML methods where field study intervention took place after managers became concerned that current requirements gathering procedures were inadequate. Other organizations with established development procedures may be less receptive.

Developing requirements requires tools to support feedback as mutually interacting stakeholders explore problems, review artifacts and share concerns with other team members. This study suggests that XML and XSLT tools represent requirements in flexible and understandable forms. The tools support stakeholders as they adapt to changing requirements by customizing presentations, negotiating changes and enhancing traceability.

Sensitivity to volatile stakeholder concerns is especially important as global business partners increasingly demand compromises. Companies must understand and participate in shared business processes, but they must bargain to retain control of their relative competitive advantages. Requirements analysis increasingly demands flexible approaches for specifying, customizing and maintaining requirements. XML and XSLT tools support global partners through open, interoperable, and customizable representations.

The introduction of XML and XSLT tools to CYAN improved change management procedures as shifting requirements could be presented in a customized fashion and negotiated in a flexible environment with the ability to trace gaps and outstanding issues in a reliable environment. XML and XSLT support requirements determination by capturing, presenting and modeling use-cases, domain objects and rules. Specifically these tools (1) provide XSLT
meta-presentation tools to specify common visual structures customized for individual stakeholders, (2) maintain a simple and flexible XML schema for capturing scenarios, business rules and associated data elements and interfaces, (3) support rule-based modeling in an environment that integrates use cases with business and technical rule representations.

XML helps stakeholders share fast changing requirements by defining schema and capturing information in self-describing documents. XML standards promote partner cooperation, model extensibility, and rich semantic representations. XSLT targets stakeholder concerns by highlighting, clustering, indexing and actively modeling representations. Shape shifting representations mediate the dissonance of change to help stakeholders see, simulate and manipulate evolving requirements. XSLT provides the vehicle to filter the semi structured requirements and transform into documents according to frame of reference and context of the stakeholder.

The field study demonstrated that XML allows full capture and binding of detached stories and use case requirements that otherwise become unmanageable in a multi company with evolving partnerships and business models. Requirements determination is messy, untidy and full of contradictions. Effectively snaring and binding changing sets of requirements begins with the recognition of the multiple concerns, hopes, skepticism and fears of stakeholders. XML and XSLT tools flexibly allow requirement artifacts to shift shape to connect customized models to support technical, managerial and partner needs.

Additional Research

Based upon the field study, additional relationships of XML, XSLT are being mined. The focus of future work is the role of prototyping on influencing stakeholders in establishing project system requirements.
REFERENCES


