1-2016

A Work System Front End for Object-Oriented Analysis and Design

Steven Alter
University of San Francisco, alter@usfca.edu

Narasimha Bolloju
LNM Institute of Technology, Jaipur, India, narsi.bolloju@gmail.com

Recommended Citation

Alter, S. and Bolloju, N. "A Work System Front End for Object-Oriented Analysis and Design," International Journal of Information Technologies and Systems Approach, 9(1), 1-18, January-June 2016

A Work System Front End for Object-Oriented Analysis and Design

Steven Alter
University of San Francisco
San Francisco, USA
alter@usfca.edu

Narasimha Bolloju
LNM Institute of Information Technology
Jaipur, India
narsi.bolloju@gmail.com

ABSTRACT

This paper proposes that basic ideas from the work system theory (WST) and the work system method (WSM) might serve as a front end to object-oriented analysis and design (OOAD), thereby providing a path from business-oriented descriptions to formal, technical specifications. After describing the background motivation and summarizing work system concepts, the paper uses a hiring system example to show how two tools from WSM can be used as a front end for OOAD, in effect, a step before creating use case diagrams and other types of Unified Modeling Language (UML) artifacts. Potential benefits of this approach stem from a business-oriented question, "how can we improve this work system's performance," rather than an IT-oriented question, "how can we create a technical artifact that will be used?"

Keywords

Work systems theory, Work system method, Object-oriented analysis and design, Use cases, UML artifacts

AN ALTERNATIVE STARTING POINT FOR ANALYSIS AND DESIGN

This paper explores the possibility of using work system concepts as the front end of an object-oriented analysis and design (OOAD) process, thereby addressing a widely recognized problem related to difficulties in communication and collaboration between business and IT professionals. The problem is that many concepts and methods designed for IT professionals simply do not satisfy the needs of business professionals. The fact that some highly talented IT professionals may do fine with the existing toolset and approaches does not minimize the confusion and blank stares encountered by many others when trying to apply technically-oriented representations with end users.

This paper shows how the problem may be addressed by linking ideas from an analysis and design approach for business professionals with established analysis and design concepts and methods for IT professionals. The first group of concepts comes from work system theory (WST) and the work system method (WSM). WST is a theoretical basis that emerged from an effort to develop a systems analysis method for business professionals that was eventually called WSM. Various versions of WSM – based on WST - were developed and tried out with MBA and Executive MBA students over many years (Alter, 1995, 2003, 2006, 2013; Truex et al., 2010). WST and WSM are explained in the following section. The second set of concepts consists of use case diagrams and other UML artifacts associated with OOAD, which was developed as a

method for IT professionals attempting to produce software that meets requirements produced in collaboration with managers and other business professionals. The creators of UML asserted that any modern object-oriented approach to developing information systems must be (1) use case driven, (2) architecture-centric, and (3) iterative and incremental (Dennis et al, 2009, p. 18). OOAD produces formal specifications that help IT professionals produce well-designed software.

Establishing links between WST/WSM and OOAD addresses important problems in requirements determination, a process that is problematic and error-prone due to difficulties in communicating between business-oriented and IT-oriented worldviews. With a business-oriented worldview, the system of concern is a work system in which human participants perform work using information, technologies, and other resources to produce product/services for internal or external customers. This work system focus is directly related to topics that managers and business professionals care about greatly, i.e., how well their work systems perform and how to improve performance. In contrast, specifications for IT-based tools are more distant from both their understanding and their concerns. With an IT-oriented worldview, the system is an IT artifact that is used by users while performing work. Thus, without diminishing the importance of UML specifications for architecture-based software development and maintenance processes, there is no reason to assume that initial collaborations between business and IT professionals should be framed around concepts that drive object-oriented specifications for IT professionals. It is possible that interacting around use case terminology introduces an unnecessary bias because it focuses on uses of technology rather than work system improvement. Ideally, collaboration with business professionals should occur around concepts they understand fully. Subsequent efforts should generate the technical specifications that programmers need.

This paper is organized as follows. A background section summarizes the limitations of use case diagrams. The next section presents an overview of WST and WSM, including the definition of work system, the work system framework, work system life cycle model, work system method, and work system metamodel. A hiring system example illustrates two ways to summarize a work system: a work system snapshot based on the work system framework and a more detailed summary based on the work system metamodel. The more detailed summary is called an Activities, Resources, Triggers and Products (ARTP) table as it includes resources used by each activity along with relevant triggers, preconditions, and post-conditions including product/services that are produced. The final sections explain how information in the work system snapshot and ARTP summaries can be converted into use case diagrams and can lead to other UML artifacts such as use case descriptions, domain class diagrams, and activity diagrams.

BACKGROUND

Most tools and methods that IT professionals use for object-oriented analysis and design are ineffective for communication between business and IT professionals because they emphasize technical documentation that is mostly of interest to software designers and programmers. Those tools and methods deemphasize topics that typical business managers care about, such as business metrics, business performance improvement, solutions to organizational problems, management of work, and the production of product/services that customers can use efficiently and effectively.

A significant part of the widely discussed user involvement problem is the lack of effective analysis methods that business professionals can use. Typical systems analysis methods and tools such as diagramming tools, UML, and Business Process Model and Notation (BPMN) are designed for use by IT specialists. There is growing literature on the limitations of these tools and their use (e.g., zur Muehlen and Recker, 2008; Dobing and Parsons, 2004; Siau and Tan 2005; Topi and Ramesh 2002). Relatively few IT analysts can engage effectively with business professionals while using these tools.

The information systems literature recognizes that many project failures are attributed to the process of information system requirements determination (e.g., Hickey and Davis 2004, Browne and Ramesh 2002; Vessey and Conger 1994). That process includes gathering information about requirements from users by analysts, representation of elicited requirements by systems analysts, and verification or validation of the requirements with the users (Browne & Rogich 2001). As discussed by Markus and Mao (2004) and other research articles related to user participation, communication and knowledge issues are key determinants of successful requirements determination. The literature on communication and knowledge gaps goes back to Snow's (1961) discussion of the sciences and the humanities as two separate cultures of modern society. Beath and Orlikowski (1994) describe common biases in system-related interactions between business and IT professionals. Despite the general agreement on the importance of user involvement during systems analysis, the level and quality of user involvement are often inadequate (Kujala, 2003; Markus and Mao, 2004).

Unfortunately, requirements modeling methods that are designed for use by IT professionals, such as UML, involve formal notations that are difficult to comprehend for people with a limited background in technology (Antony et al, 2005; Glinz, 2000). Users often find it difficult to verify the accuracy and completeness of requirements expressed using unfamiliar formalisms. Despite those difficulties, modeling methods such as UML are frequently used for requirements specifications that need users' review and approval (Dobing and Parsons, 2008). The use of methods that are poorly matched to the interests and knowledge of business professionals contributes to difficulties in verifying requirements and, ultimately, to the development of software applications that are ineffective or difficult to explain and use. For example, Samuel et al (2015) describe how related types of impediments (e.g., not well-understood by analysts; excessively complex, not self-explanatory, and too technically focused for end users) affect the creation and comprehension of process models. Conclusions such as those support recent research related to collaborative requirements elicitation (e.g., Konaté et al, 2014; Azadegan et al, 2013; Lim and Finkelstein 2012), and tools for automating requirements elicitation and analysis (e.g., Meth et al 2013; Nguyen et al 2014). Our research takes a fundamentally different approach of identifying methods that create a link between business- and IT-oriented analysis and design.

We assume that most readers are familiar with OOAD and its reliance on use case diagrams, but that they may not be familiar with work system concepts. Therefore we identify recognized limitations of use case diagrams and then summarize aspects of WST and WSM.

Limitations of Use Cases

Although use cases are used widely (e.g., Dobing and Parsons, 2004, 2008), the creation and application of use cases encounters a number of problems whose existence supports the potential value of an alternative front end for OOAD.

Techno-centric nature of use cases. According to the latest specification of UML from the Object Management Group (OMG),, "A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system." (OMG, 2011, p. 606) In effect, a use case answers the following question: "which activities will use the IT artifact that is being built?" That is not the best question to ask business professionals whose main concern is improving the efficiency and effectiveness of work systems containing human participants, not just users of technology. More important questions concern how the current work system operates, how well it operates, and how work system changes could yield better performance. Those changes could involve new or existing IT artifacts and/or changes in business processes, information, skills, knowledge and incentives of participants, expectations of work system customers, and the surrounding environment.

Difficulties teaching use case modeling to novices. Use case modeling is relatively difficult to teach to novices. For example, an empirical study on the quality of commonly used UML artifacts (Bolloju and Leung, 2006) reported that more than half of the use case diagrams contained "manual operations listed as use cases." Siau and Loo (2006) identified other difficulties. Many novices have difficulty visualizing the business situation within which use cases will operate. A work system approach addresses that issue more effectively.

Practical limitations. Use case models have many practical limitations. Baekgaard (2005) notes unrealistic assumptions that the border between the IT-system and its environment is clear, and that activities of actors are well-understood and can be reduced to interactions with the IT system. Kim et al. (2006) argues that use-case driven analysis does not provide an adequate rationale for the various artifacts generated during the requirements analysis. Rational Software published an article about avoiding ten ways in which project teams misuse use cases (Gottesdiener, 2002).

Omission of important information. By design, use case diagrams identify actors, activities, and associations between actors and activities. While simplicity is beneficial, use case diagrams (without use case narratives) also omit important information, such as "nonfunctional" requirements, identification of information created, used, or updated, identification of product/services produced, and identification of customers for those product/services. A different, more detailed summary that is not overwhelming might be more effective.

Work System Theory

WST provides a perspective for understanding systems in organizations, whether or not those systems use IT intensely. WST consists of three main components: 1) the definition of work system; 2) the work system snapshot, which is a static view of a work system during a period when it is relatively stable; and 3) the work system life cycle model, a dynamic view of how a work system changes over time. Various versions of WSM that have been used are all based on the main ideas in WST but are not part of WST, which has been applied and extended into other

areas. The extensions of WST are various concepts, frameworks, methods, and theories that are based on WST and can be used whenever appropriate for thinking about work systems in general, about categories of work systems, and about specific systems in organizations. Examples of published extensions of WST include work system principles, work system design spaces, various versions of a work system metamodel (Alter, 2013, 2015), and applications to service and service systems. The metamodel plays an important role in this paper.

Definition of work system. A work system is a system in which human participants and/or machines perform processes and activities using information, technology, and other resources to produce specific product/services for specific internal or external customers. Almost all value chain systems (e.g., systems for inbound logistics, operations, sales and marketing) and support systems (e.g. systems for procurement and human resources) are IT-reliant work systems that use IT to operate efficiently and effectively.

A work system viewpoint differs from the more techno-centric viewpoint that underlies typical analysis and design textbooks, in which "the system" is a technical artifact (hardware and software) with human users, not a sociotechnical system with human participants. From a techno-centric viewpoint, a use case is "an activity that the system performs" (Satzinger et al., 2009, p. 160), a functional requirement is a "system requirement that describes an activity or process that the system must perform" (p. 122), and a nonfunctional requirement is a characteristic of the system other than activities it must perform or support, such as technology, performance, usability, reliability, and security." (p. 123). In contrast, the default view of a work system sees "the system" as a sociotechnical system with human participants. Work system analysis and design includes technology, process, participants, information, and other relevant factors. Work system concepts can be used by business professionals (Truex et al. 2010, 2011) and even freshmen undergraduates (Recker and Alter, 2012). It can help novice analysts develop use case diagrams.

Work system framework. The work system framework (Figure 1a) is a pictorial representation of a work system in terms of nine elements included in a basic understanding of the work system's form, function, and environment during a period when it is relatively stable, even though incremental changes may occur during that period. The arrows say that the specific elements of a work system should be in alignment. Of the nine elements:

- Processes and activities, participants, information, and technologies are completely within the work system.
- Customers and product/ services may be partially inside and partially outside because customers often participate and product/services take shape within the work system.
- Environment, infrastructure, and strategies are outside even though they have direct effects within the work system.

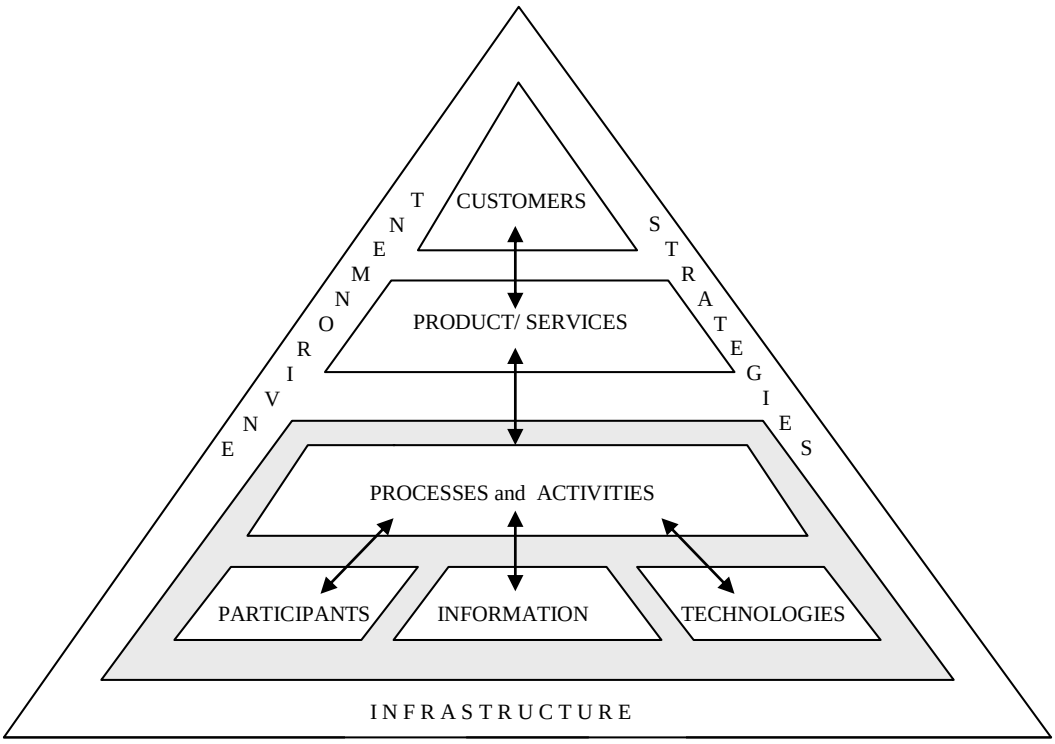


Figure 1. The work system framework

Work System Life Cycle Model. The work system life cycle model (WSLC) is the other central framework in WSM. As shown in Figure 1b, it expresses a dynamic view of how work systems change over time through iterations involving planned change and emergent (unplanned) change that occurs through adaptations, bricolage, and workarounds. (Alter 2006, 2013). The WSLC differs fundamentally from the "system development life cycle" (SDLC) because the SDLC is basically a project model and focuses primarily on building a technical artifact. Due to this paper's focus on an operational view of work systems, the WSLC will not be discussed further.

WORK SYSTEM METHOD

WSM evolved over many years and through many versions as a flexible systems analysis and design method devised for business professionals concerned with creating or improving work systems. WSM applies WST but is not part of WST. It starts with whatever problems, opportunities, or issues launched the analysis. The "as is" and "to be" systems are work systems rather than configurations of hardware and software. The work system analysis template summarized in Table 1 is an illustrative classroom version of WSM that was designed to accomplish a dual pedagogical purpose. Filling in the appendices provides experience in performing organized, business-oriented WSM analysis of a work system. Writing the management briefing reinforces the difference between performing the analysis and producing a management-oriented report.

Management briefing	1. Executive summary 2. Background 3. System and problem 4. Analysis and possibilities 5. Recommendation and justification
Appendix 1: Initial summary of the existing work system and the problem or opportunity	1. Name of work system 2. Main problem or opportunity 3. Significance of the work system 4. Constraints that limit the possible recommendations 5. Performance gaps related to processes, participants, information, or technology 6. Performance gaps related to customer perceptions of products and services
Appendix 2: Summary of the "as is" work system	1. Work system snapshot of the "as is" work system 2. Customer value and customer concerns (for primary customers) 3. Customer responsibilities (for primary customers).
Appendix 3. Summary of problems, issues, opportunities in the "as is" work system	Problems, issues, and opportunities: 1. for the system as a whole 2. for each step in the processes or activities 3. for specific work system elements (e.g., participants, information) 4. for specific types of activities (e.g., information processing, informing, communicating, controlling work, making decisions.)
Appendix 4: Summary of the recommendations and their likely impacts	1. Work system snapshot of the "to be" work system. Likely impact of recommended changes: 2. for the system as a whole 3. by step 4. related to specific types of activities

Table 1. Summary of a work system analysis template

Work System Snapshot

Table 2 is an example of a "work system snapshot," a tool mentioned in the work system analysis template in Table 1. This tool is a formatted one-page summary of a work system in terms of the six central elements of the work system framework. The specific example in Table 2 is related to hiring new employees. The requirement of not exceeding one page avoids excessive detail and helps focus attention on the system's scope. Work system snapshots require rigorous thinking because of internal consistency rules that are explained in Alter (2006), e.g., each product/service must be received and used by at least one customer group.

Truex et al. (2010, 2011) reports that many hundreds of MBA and executive MBA students produced work system snapshots when analyzing real world work systems. Recker and Alter (2012) discuss how freshmen undergraduates used work system snapshots to understand systems in organizations. As will be presented later, one of this paper's authors found that the introduction of a work system snapshot at the beginning of a previously used textbook example helped novice analysts produce an average of twice as many valid use cases as a previous class that had not seen work system snapshots. This evidence suggests that work system snapshots may be useful in the early stages of OOAD.

<table>
<tr><th>Customers</th><th>Products & Services</th></tr>
<tr><td>• Hiring manager
• Larger organization (which will have the applicant as a colleague
• HR manager (who will analyze the nature of applications)</td><td>• Applications (which may be used for subsequent analysis)
• Job offers
• Rejection letters
• Hiring of the applicant</td></tr>
<tr><th colspan="2">Major Activities and Processes</th></tr>
<tr><td colspan="2">• Hiring manager submits request for new hire within existing budget
• Staffing coordinator defines the parameters of the new position.
• Staffing coordinator publicizes the position.
• Applicants submit job applications.
• Staffing coordinator selects shortlisted applicants.
• Hiring manager identifies applicants to interview.
• Staffing coordinator sets up interviews.
• Hiring manager and other interviewers perform interviews.
• Hiring manager and other interviewers provide feedback from the interviews.
• Hiring manager makes hiring decisions.
• Staffing assistant sends offer letters or rejections.
• Successful applicant accepts or rejects job offer or negotiates further.</td></tr>
<tr><th>Participants</th><th>Information</th><th>Technologies</th></tr>
<tr><td>• Hiring managers
• Staffing coordinator
• Applicants
• Staffing assistant
• Other employees who perform interviews</td><td>• Job requisition
• Job description
• Advertisements
• Job applications
• Cover letters
• Applicant resumes
• Short list of applicants
• Information and impressions from the interviews
• Job offers
• Rejection letters</td><td>• New HR portal that is being built
• Word processor
• Telephones
• Email</td></tr>
</table>

Table 2: Work system snapshot of a recommended "to be" work system

WORK SYSTEM METAMODEL

The work system metamodel is a more recent extension of WST. Although the work system framework has proven useful for high-level summaries that help in defining a work system's scope, it omits many important relationships and details. For example, there is no arrow linking participants and technology or information and technology even though specific information and technologies are used by specific work system participants in specific activities within the work system. Also, both classroom discussions and written assignments produced by MBA and Executive MBA students revealed confusion and many ambiguities related to the work system framework when applied to specific situations. (see Alter, 2010, p. 8) A framework for deeper, more detailed analysis should provide greater clarity on concepts and more specific guidance about important relationships. Ideally, it should support more rigorous analysis without requiring abstruse UML terminology.

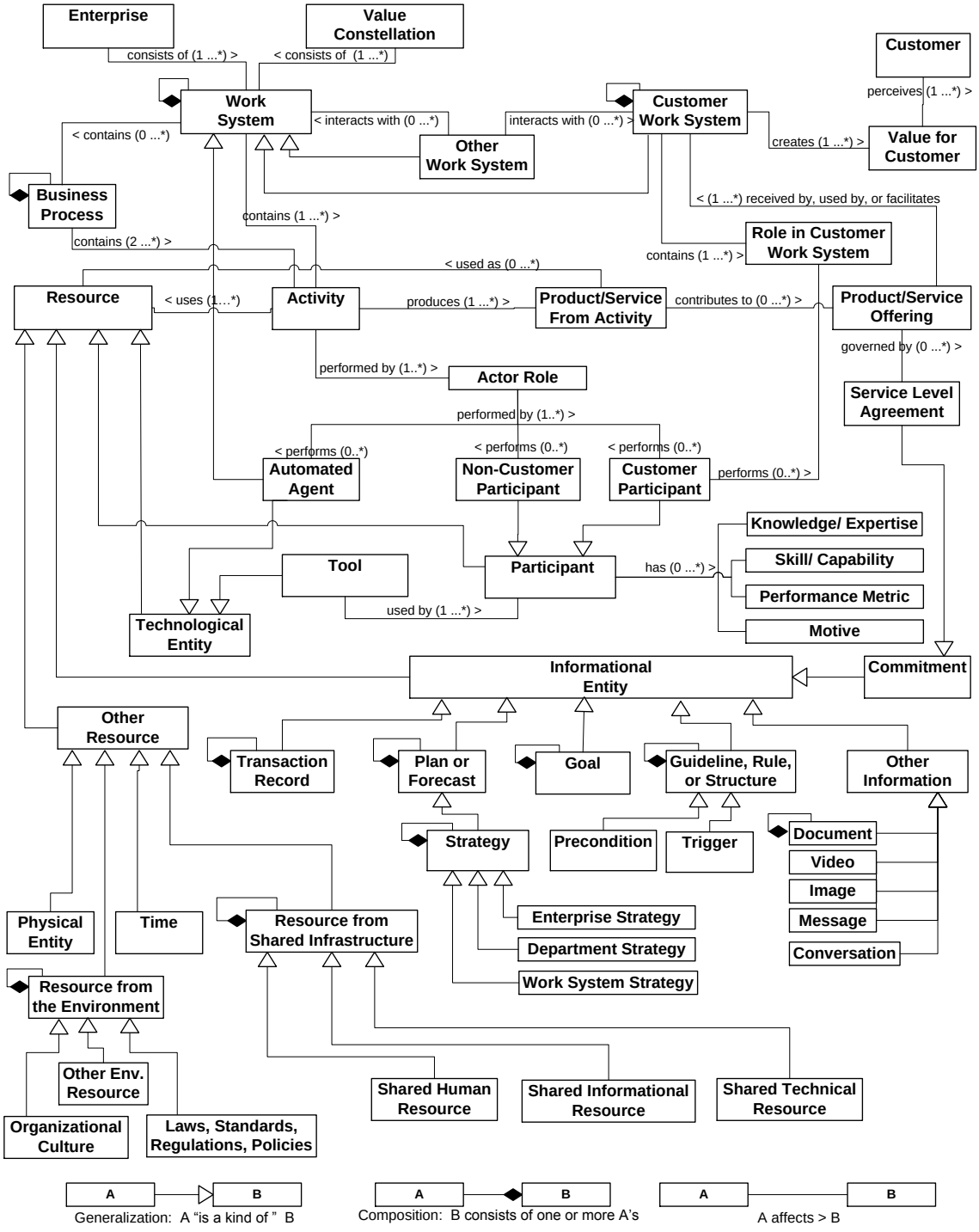
The work system metamodel is basically a more detailed specification of the work system framework, with each element re-interpreted in a more detailed way. The version in Figure 2 appeared in Alter (2015). In the metamodel, information becomes informational entity, technology becomes technological entity and is divided into tools and automated agents, activities are performed by three types of actors, and so on. "Uses" is a relationship between a participant and a tool. Attributes of entity types, such as goals, characteristics, metrics, principles, and other concepts are not shown, just as attributes of classes might not be shown in a summarized UML class diagram. Those attributes would be used while defining problems or opportunities, evaluating "as is" work systems, and justifying proposed improvements. Overall, the metamodel takes over where the work system framework provides insufficient detail. For example, every activity produces product/services that may be resources for other activities

and/or may be received and used by the work system's customers. Such relationships in the metamodel can be the basis of straightforward tools even though they do not appear explicitly in the less detail-oriented work system framework.

EXAMPLE ILLUSTRATING AN ALTERNATIVE FRONT END FOR ANALYSIS AND DESIGN

The summary of the work system analysis template in Table 1 calls for using a work system snapshot as a summary of the "as is" work system and the recommended "to be" work system. The example in Table 2 illustrates that type of summary, which is based on the work system framework (Figure 1).

The metamodel in Figure 2 provides a path for describing the work system in greater depth as a step toward more detailed analysis and design. The metamodel says that a given activity produces product/services by using human, informational, technical, and other resources. That general idea is the basis of the ARTP summary in Table 3, which is an extension of the work system snapshot in Table 2 and builds on the discussion of "service responsibility tables" in Tan et al. (2011). The columns for actor and activity came directly from the "processes and activities" section of Table 2. The columns for *information used* and *information created, updated, or deleted* are based on the information section of Table 2 and a relatively minor effort to fill in items that were omitted from Table 2. The technology column mentions the HR portal repeatedly because that is the new technical artifact that will be built. It also mentions other technical artifacts that the work system snapshot omitted. The columns for *trigger, preconditions, and post-conditions* (including product/services produced) combine aspects of the metamodel (e.g., that every activity produces product/services) and the fact that triggers, preconditions, and post-conditions are often included in use case narratives. While work system snapshots are a better starting point for requirements determination, ARTP summary tables provide additional information that is understandable to business professionals and also is useful to IT professionals who will develop technical specifications.



Note: Many elements in the conceptual model have goals, attributes, performance indicators, and related principles, patterns, and generalizations that do not fit into a one page representation, and that must be included in more detailed explanations.

Metamodel representing a more detailed version of the work system framework

CONVERTING WORK SYSTEM SUMMARIES INTO USE CASE DIAGRAMS AND OTHER UML DIAGRAMS

Thus far we have discussed two different versions of work system summaries. The work system snapshot in Table 2 is based on the work system framework; the more detailed ARTP summary in Table 3 is based on the work system metamodel. The next step in discussing the potential for a work system front end to OOAD involves a relatively mechanical way to convert each type of summary into a use case diagram. In both cases, the result will be the use case diagram in Figure 3. Each type of summary can also be a starting point for producing UML diagrams.

Activity	***Actors***	***Information used***	***Information created, updated, or deleted***	***Technology***	***Trigger***	***Preconditions***	***Post conditions (including products/ services produced)***
Submit request for new hire.	Hiring manager	Hiring budget	Job requisition	HR portal	Need for new employee	Sufficient hiring budget	Job requisition exists
Define parameters of the job.	Staffing coordinator	Job requisition, Hiring policies	Job description	Word processor, HR portal	Job requisition	Job requisition	Job description
Publicize the job opening	Staffing coordinator	Experience with advertising media	Advertisement	HR portal, Web site for selected media	Job requisition, Job description	Job requisition, Job description	Advertisement displayed on web sites
Submit application	Applicant	Job description	Cover letter, Job application, Resume	HR portal	Advertisement displayed on web sites	Advertisement displayed on web sites	Receipt of cover letter, job application, resume
Select shortlist	Staffing coordinator	Job application	Short list of best applicants	HR portal	Deadline for job applications	Availability of job applications	Short list available to hiring manager
Identify applicants to interview	Hiring manager	Short list of best applicants	List selected for interviews	HR portal	Short list available to hiring manager	Short list available to hiring manager	List selected for interviews
Set up interviews	Staffing coordinator	Schedules of interviewers	Interview schedule	Employee calendar system, HR portal	List selected for interviews	List selected for interviews	Interviews schedule
Perform interview	Hiring manager, other interviewers	Job description, Job application	Interview impressions	HR portal	Interview schedule	Interview schedule	Interview impressions
Make hiring decision	Hiring manager	Interview impressions	Hiring decision	HR portal	Completion of interviews	Completion of interviews	Hiring decision
Send offer letters or rejections.	Staffing assistant	Hiring decision	Job offer, Rejection letter	HR portal	Hiring decision	Hiring decision	Job offer, Rejection letter
Accepts or rejects job offer.	Applicant who was selected	Job offer	Applicant's response to offer	HR portal	Job offer	Job offer	Applicant's response to offer

Table 3. Activities, Resources, Triggers, and Products (ARTP) summary table

Converting from a Work System Snapshot or ARTP Summary to a Use Case Diagram

Steps listed under activities and processes in the work system snapshot can be viewed as tentative use cases. The process of creating a use case diagram from a work system snapshot includes:

- Assume the participants in the work system snapshot are actors in the use case diagram.
- Assume that the action part of each process or activity in the work system snapshot is an activity in the use case diagram.
- Think about which activities will be supported by the software that is being built. Place those activities inside of ovals within the boundary of computerized system and place the other activities inside of ovals outside of that boundary.
- Link each actor to the relevant activities.

Figure 3: Use case diagram corresponding to the work system snapshot in Table 2

Converting from an ARTP summary table to a use case diagram follows the same path. The first two columns of the ARTP summary table already accomplish the first two above steps. Performing the other two steps mentioned above will produce the use case diagram.

Implication of the two conversion processes. The mechanical nature of the two conversion processes above implies that use case diagrams can be produced from either work system snapshots or the more detailed representation in ARTP summary tables. If there are advantages to using either work system snapshots or ARTP summaries in collaboration with business professionals, there is no need to start with use case diagrams because use case diagrams can be derived from either work system snapshots or ARTP summaries. The opposite direction is not a practical path because both work system snapshots and ARTP summaries contain much more information than use case diagrams.

Regardless of whether use case diagrams are used in discussions with business professionals, it may be important to produce use case diagrams in order to make the programming effort more efficient through appropriate modularization and exploitation of reuse. For example, it may be useful to introduce <<include>> and <<extend>> relationships that are important for programming but of little interest to business professionals who are not concerned with whether information about applicants is partitioned into information about people in general and other information specific to the applicants. That type of modularization and reuse issue is important to programmers but should be invisible to business professionals. Moreover, some researchers (e.g., Genova et al, 2002) argue that ≪include≫ and ≪extend≫ relationships can be misleading, unnatural, and difficult to understand for typical practitioners.

Converting from a Work System Snapshot or ARTP Summary to Other Types of UML Artifacts

Both the work system snapshot and the activity summary table contain starting points for not only use case diagrams, but also use case descriptions, domain class diagrams, activity diagrams, and state machine diagrams. Consider how those narratives and diagrams can be produced directly from the ARTP table:

Use case descriptions. Use case descriptions or narratives corresponding to use cases that are identified can be created using almost all the information present in the rows of the ARTP table. In fact, many entries in this table (e.g., actors, triggers, pre-conditions and post-conditions) have an equivalent representation in use case descriptions. Entries related to informational entities contribute to step descriptions in the narratives. Table 4 presents expected functionality corresponding to the "select shortlist" use case.

Use case name	**Select shortlist**	
Primary actor	Staffing coordinator (SC)	
Pre-conditions	Availability of job applications	
Post-conditions	Shortlist available to hiring manager	
Trigger	Deadline for job applications	
Main success scenario	1	SC selects the job requisition for shorting of job applications
	For each job application submitted and for each element of essential criteria perform step 2	
	2	System determines if the application meets the selected criterion element by comparing it with the corresponding qualification in the job application
	For each job application meeting the essential criteria and for each element of desirable criteria perform step 3	
	3	System rates the application with respect to the selected desirable criterion element by comparing it with the corresponding qualification in the job application
	4	System computes total score
	5	System generates a sorted list of shortlisted applications
Extensions		
	2a	System cannot determine if application meets the selected criterion element
		1. System displays the application details corresponding to that criterion and the resume
		2. SC reviews the application details & resume, and assesses whether the essential criterion is met or not
		3. SC adds comments about his/her assessment
	3a	System cannot rate the desirable criterion
		1. System displays the application details corresponding to that criterion and the resume
		2. SC reviews the application details & resume, and assesses whether the desirable criterion is met or not
		3. SC adds comments about his/her assessment

Table 4: Use case description for "Select Shortlist"

Class diagrams. Class diagrams for domain classes can be produced as follows. Consider the columns for information used; information created, updated, or deleted; triggers; preconditions; and post conditions. Identify the entity types about which information is created, used, updated, or deleted. Those can be viewed as the names of tentative domain classes. Associations between the classes (e.g., 0 ... *) can be filled in based on general knowledge of the situation and confirmed by subject matter experts if there is uncertainty. Similarly, a first cut at attributes of each class can be filled in based on general knowledge (see Figure 4). More detailed analysis of the situation will probably find additional entity types and attributes.

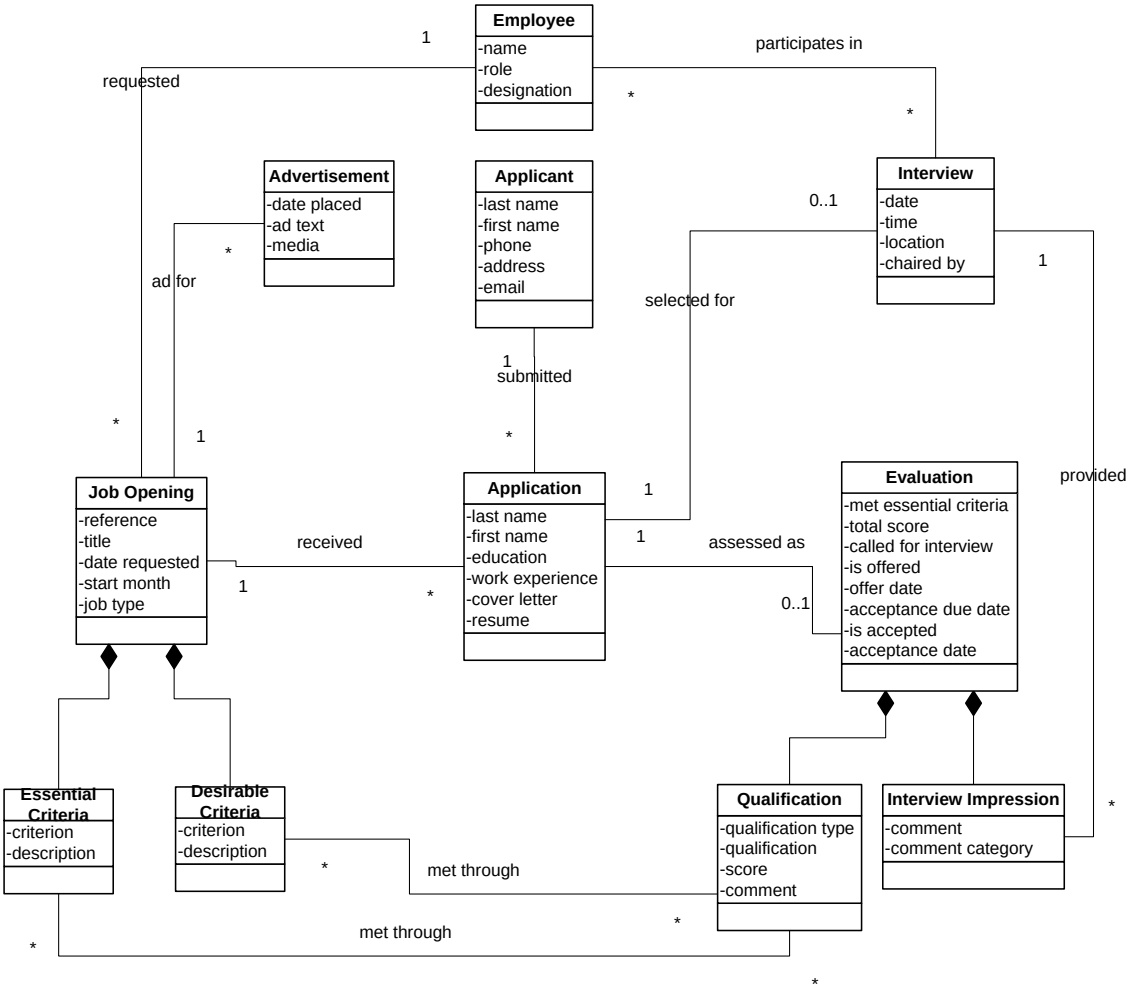


Figure 4: Class diagram based on the ARTP summary in Table 3

Activity diagrams. Creating an activity diagram for the entire work system is not totally mechanical, but can be guided as follows (see Figure 5). Insert each step in the activity column into a tentative activity diagram. Use triggers, preconditions, and post conditions from the ARTP summary to insert branching logic wherever it belongs.

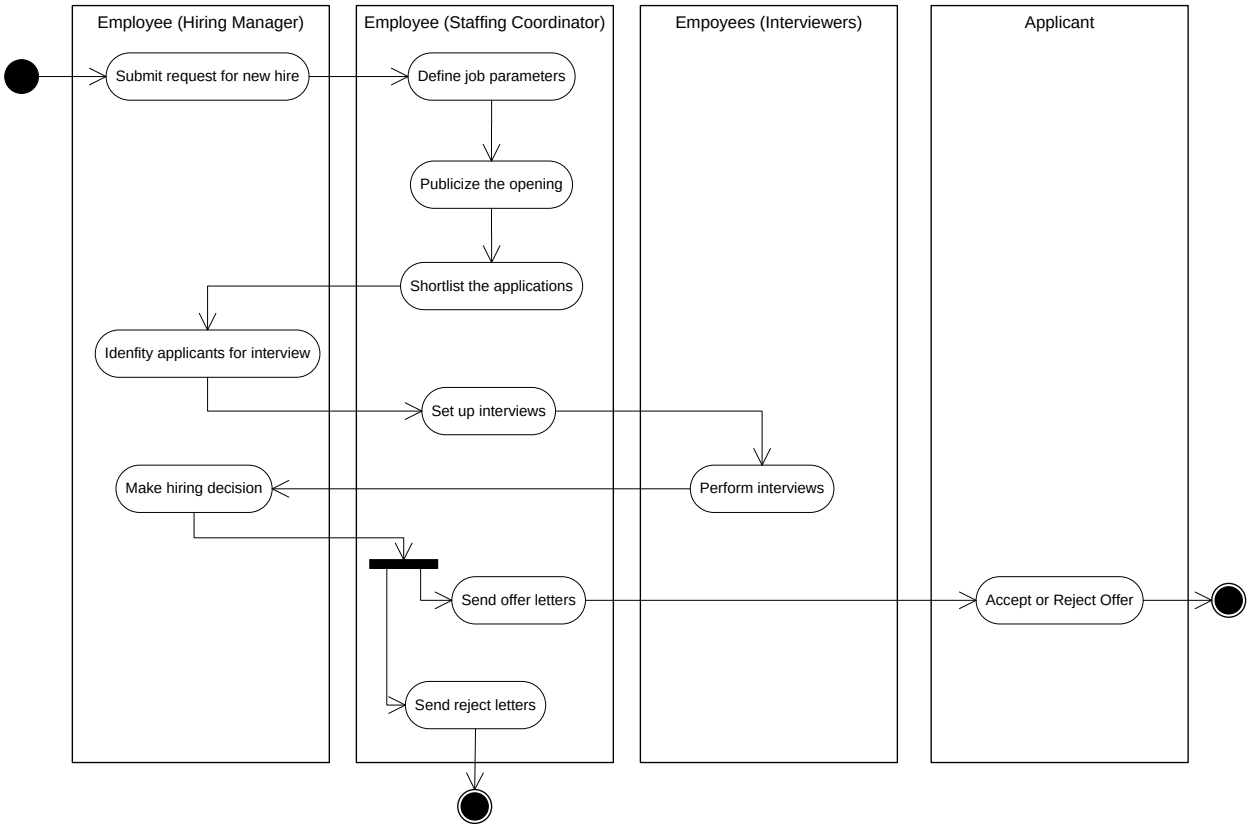


Figure 5: Activity diagram based on the work system snapshot in Table 2 and the ARTP summary in Table 3

For an activity diagram for the individual activities identified in the ARTP summary, start with triggers, preconditions, and post conditions from the activity summary table, and then fill in any missing details that would appear in a use case narrative.

State machine diagrams. For a state machine diagram, start with the domain class diagram previously produced (see Figure 6). Identify all possible states of objects in each class. Make sure that the ARTP summary and use case diagram include or correctly express all of the activities needed for transitions to and from all possible states of objects in each class.

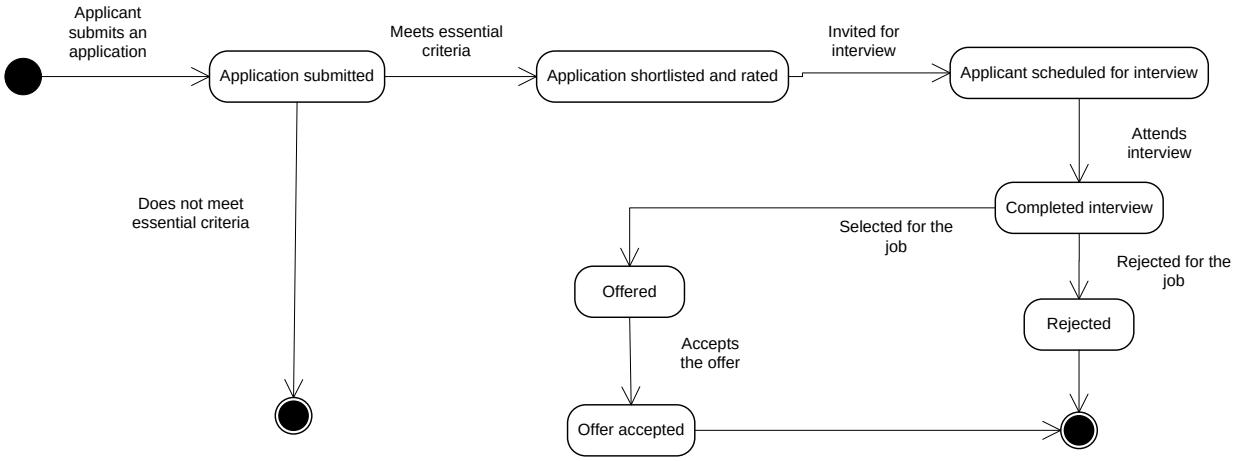


Figure 6: State machine diagram for the Application class based on the work system snapshot in Table 2 and the ARTP summary in Table 3

Other UML representations that are fundamentally about programming choices such as the structure and behavior of interface classes, control classes, and non-persistent classes cannot be derived directly from the work system snapshot or ARTP tables. Choices related to those UML representations are neither visible nor understandable to most business professionals.

CONCLUSION

This paper's purpose was to illustrate the possibility of using work system concepts to make the early parts of OOAD more effective, especially activities involving collaboration with business professionals. The approach here would not be appropriate for OOAD for purely technical artifacts such as internal components of computer systems.

The paper showed how a work system snapshot or ARTP summary can be converted directly into a use case diagram, thus illustrating that a type of business-oriented front end based on WST can be linked directly to existing OOAD techniques that start with use case diagrams. The paper also illustrated how other types of UML artifacts can be prepared based on the work system snapshot and ARTP summary tables. The potential advantage of this approach is that work system concepts are well suited to collaboration with business professionals because they focus on improving the performance of work systems, rather than specifying hardware/software artifacts that satisfy previously defined requirements supplied by others.

While the purpose of this research was not to try to replace use case diagrams, a later stage of this research might include experiments that would compare the relative efficacy of use case diagrams versus tabular representations based on work system concepts as communication and scoping tools near the beginning of an analysis and design effort. This paper does not attempt to demonstrate that the proposed approach is superior to use case diagrams in some general way.

Rather, it demonstrates that the proposed approach may be a useful alternative for early stages of collaboration with business professionals.

REFERENCES

Alter, S. (1995) "How should business professionals analyze information systems for themselves?" pp. 284-299 in E. Falkenberg, et al., *Information System Concepts: Toward a Consolidation of Views*, Proceedings of the IFIP WG 8.1 Working Conference on Basic Information System Concepts, Marburg, Germany.

Alter, S. (2003) "18 Reasons why IT-Reliant Work Systems Should Replace the IT Artifact as the Core Subject Matter of the IS Field," *Communications of the Association for Information Systems*, 12,23, pp. 365-394.

Alter, S. (2006) *The Work System Method: Connecting People, Processes, and IT for Business Results*, Larkspur, CA: Work System Press.

Alter, S. (2010) "Bridging the Chasm between Sociotechnical and Technical Views of Systems in Organizations," *Proceedings of ICIS 2010, the 31st International Conference on Information Systems*.

Alter, S. (2013) Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future. Journal of the Association for Information Systems, 14(2), 72-121.

Alter, S. (2015) "Work System Theory as a Platform: Response to a Research Perspective Article by Niederman and March," *Journal of the Association for Information Systems*, 16(6), 483-514.

Antony, S., Batra, D., & Santhanam, R. (2005) "The use of a knowledge-based system in conceptual data modeling," *Decision Support Systems*, 41(1), 176-188.

Azadegan, A., Papamichail, K. N., & Sampaio, P. (2013). Applying collaborative process design to user requirements elicitation: A case study. *Computers in Industry*, 64(7), 798-812.

Baekgaard, L. (2005) "From Use Cases to Activity Cases," *Proceedings of ALOIS*2005, Limerick, Ireland, 15–16 March 2005*, Ågerfalk, P. J., Bannon, L., and Fitzgerald, B. (eds.)

Beath, C.M., & Orlikowski, W. J (1994) "The contradictory structure of systems development methodologies: Deconstructing the IS-user relationship in information engineering," *Information Systems Research* 5, no. 4: 350.

Bolloju, N., & Leung, F.S.K. (2006). Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM* 49,7, pp. 108-112.

Browne, G.J. & Ramesh, V. (2002) "Improving information requirements determination: a cognitive perspective," *Information & Management*, 39(8), 625–645.

Browne, G.J. & Rogich, M.B. (2001) “An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques,” *Journal of Management Information Systems*, 17(4), 223–249.

Dennis, A., Wixom, B.H., & Roth, R. M. (2009) *Systems Analysis & Design with UML Version 2.0: an Object-Oriented Approach*, 3rd, ed., New York, NY: John Wiley & Sons, Inc.

Dobing, B. and Parsons, J. (2008) "Dimensions of UML Diagram Use: A Survey of Practitioners," *Journal of Database Management*, 19,1,, pp. 1-18.

Dobing, B., & Parsons, J. (2004) "How UML is Used," *Communications of the ACM* 49,5, pp. 109-113.

Génova, G., Llorens, J., & Quintana, V. (2002) “Digging into Use Case Relationships.” *Lecture Notes in Computer Science* (2002), pp. 115–127.

Glinz, M. (2000) “Problems and deficiencies of UML as a requirements specification language*,” Paper presented at the 10th International Workshop on Software Specification and Design (IWSSD-10)*, San Diego.

Gottesdiener, E. (2002) "Top Ten Ways Project Teams Misuse Use Cases - and How to Correct Them, *Rational Edge*, http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jul02/TopTenWaysJul02.pdf (Accesed on Mar. 23, 2012)

Hickey, A.M. & Davis, A.M. (2004) “A unified model of requirements elicitation,” *Journal of Management Information Systems*, 20(4), 65–84.

Kim, J., Park, S. & Sugarman, V. (2006) "Improving use case driven analysis using goal and scenario authoring: A linguistics-based approach," *Data & Knowledge Engineering*, 58, pp. 21-46.

Konaté, J., Sahraoui, A. E. K., & Kolfschoten, G. L. (2014). Collaborative Requirements Elicitation: A Process-Centred Approach. *Group Decision and Negotiation*, *23*(4), 847-877.

Kujala, S. (2003) “User involvement: a review of the benefits and challenges,” *Behaviour & Information Technology*, *22*(1), 1-16.

Lim, S. L., & Finkelstein, A. (2012). StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation. *Software Engineering, IEEE Transactions on*, *38*(3), 707-735.

Markus, M. L., & Mao, J. (2004) “Participation in Development and Implementation–Updating an Old,Tired Concept for Today’s IS Contexts,” *Journal of the Association for Information Systems*, *5*(11), 514-544.

Meth, H., Brhel, M., & Maedche, A. (2013). The state of the art in automated requirements elicitation. *Information and Software Technology*, *55*(10), 1695-1709.

Nguyen, T. H., Grundy, J., & Almorsy, M. (2014, August). GUITAR: An ontology-based automated requirements analysis tool. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International* (pp. 315-316).

OMG (2011) Unified Modeling Language, Superstructure Version 2.4.1, http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/ (Accessed Mar. 24, 2012).

Recker, J. & Alter, S. (2012) "Using the Work System Method with Freshman Information System Students," *Journal of Information Technology Education: Innovations in Practice*, 11, pp. 1-24.

Samuel, B. M., Watkins, L. A., Ehle, A., & Khatri, V. (2015) "Customizing the Representation Capabilities of Process Models: Understanding the Effects of Perceived Modeling Impediments," *IEEE Transactions on Software Engineering*, 41(1), pp. 19-39.

Satzinger, J., Jackson, R., & Burd, S. (2009) *Systems Analysis & Design in a Changing World*, Course Technology, Cengage Learning.

Siau, K. and Loo, P-P., (2006) "Identifying Difficulties in Learning UML," *Information Systems Management*, 23,3, pp. 43-51.

Siau, K., & Tan, X. (2005) "Improving the quality of conceptual modeling using cognitive mapping techniques," *Data & Knowledge Engineering*, *55*(3), 343-365.

Snow, C.P. (1959) *The Two Cultures and the Scientific Revolution*, Cambridge University Press, New York.

Tan, X., Alter, S. & Siau, K. (2011) "Using Service Responsibility Tables to Supplement UML in Analyzing e-Service Systems," *Decision Support Systems*, 51,3, June, pp. 350-360.

Topi, H. & Ramesh, V. (2002) "Human factors research on data modeling: a review of prior research, an extended framework and future research directions," *Journal of Database Management* 13, no. 2, 3–19.

Truex, D., Alter, S., & Long, C. (2010) "Systems Analysis for Everyone Else: Empowering Business Professionals through a Systems Analysis Method that Fits their Needs," *Proceedings of 18th European Conference on Information Systems*, Pretoria, South Africa.

Truex., D., Lakew, N., Alter, S., and Sarkar, S. (2011) "Extending a Systems Analysis Method for Business Professionals," *European Design Science Symposium*, Leixlip, Ireland, Oct. 2011.

Vessey, I. & Conger, S.A. (1994) "Requirements specification: learning object, process, and data methodologies," *Communications of the ACM*, 37(5), 102–113.

zur Muehlen, M. & Recker, J. (2008) "How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation," *CAiSE 2008*, Montpelier, France, pp. 465-479.

APPENDIX – LIST OF ACRONYMS USED

ARTP - Activities, Resources, Triggers and Products

BPMN - Business Process Model and Notation

OMG - Object Management Group (OMG),

OOAD - object-oriented analysis and design (OOAD)

SDLC - system development life cycle

UML - Unified Modeling Language

WSLC - work system life cycle model

WSM - work system method

WST - work system theory