

The University of San Francisco  
**USF Scholarship: a digital repository @ Gleeson Library |  
Geschke Center**

---

Business Analytics and Information Systems

School of Management

---

January 2012

# A Work System Front End for Object-Oriented Analysis and Des

Steven Alter

University of San Francisco, [alter@usfca.edu](mailto:alter@usfca.edu)

Follow this and additional works at: <http://repository.usfca.edu/at>

 Part of the [Business Administration, Management, and Operations Commons](#), [Management Information Systems Commons](#), and the [Technology and Innovation Commons](#)

---

## Recommended Citation

Alter, Steven, "A Work System Front End for Object-Oriented Analysis and Des" (2012). *Business Analytics and Information Systems*. Paper 20.

<http://repository.usfca.edu/at/20>

This Article is brought to you for free and open access by the School of Management at USF Scholarship: a digital repository @ Gleeson Library | Geschke Center. It has been accepted for inclusion in Business Analytics and Information Systems by an authorized administrator of USF Scholarship: a digital repository @ Gleeson Library | Geschke Center. For more information, please contact [repository@usfca.edu](mailto:repository@usfca.edu).

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

**Steven Alter**

University of San Francisco  
alter@usfca.edu

**Narasimha Bolloju**

City University of Hong Kong  
narsi.bolloju@cityu.edu.hk

## ABSTRACT

This paper proposes that basic ideas from the work system theory (WST) and the work system method (WSM) might become a front end to object-oriented analysis and design (OOAD). After describing the background motivation and summarizing work system concepts, it 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. 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 an IT artifact that will be used?"

## Keywords

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

## 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. Work system theory (WST) was developed as a byproduct of the development over two decades of various versions of the work system method (WSM), a systems analysis method for business professionals (Alter, 1995, 2003, 2006, 2008a). In contrast, OOAD was developed as a method for IT professionals attempting to produce software that meets requirements produced in collaboration with managers and other business professionals. OOAD emphasizes specifications that IT professionals need to produce well-designed software. 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. To the contrary, collaboration with business professionals should occur around concepts they understand and should be converted separately into a form that drives technical specifications.

This paper demonstrates that concepts from WST and WSM can serve as a front end for OOAD. 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, p. 18). This paper demonstrates the possibility of creating use case diagrams from either of two tabular work system summaries based on WST and WSM. Thus, it demonstrates linkage between well-articulated analysis and design methods for business professionals and well-articulated analysis and design methods for IT professionals.

Establishing this linkage addresses important problems in requirements determination, a problematic and error-prone process due to difficulties 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 products/services for internal or external customers. This work system focus is more natural for managers and business professionals because they care more about performing work effectively than about using IT-based tools. With an IT-oriented worldview, the system is an IT artifact that is used by users while performing work. The need for requirements does not imply that collaboration with business professionals should focus on technology. Interacting around use case terminology introduces an unnecessary bias because it focuses on uses of technology rather than work system improvement.

This paper is organized as follows. A background section summarizes limitations of use case diagrams. The next section presents an overview of components of WST and WSM, including 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 called an activities, resources, triggers and products (ARTP) table that includes resources used by each

activity, relevant triggers and preconditions, and post-conditions including products/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 therefore lead to other UML artifacts such as use case descriptions, domain class diagrams, and activity diagrams.

## **BACKGROUND**

Assuming that most readers are more familiar with OOAD than with work system concepts, we identify 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 OMG's latest specification of UML, "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.

**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 interaction 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 correcting 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 products/services produced, and identification of customers for those products/services. A different, more detailed summary that is not overwhelming might be more effective.

## **WORK SYSTEM THEORY**

Work system theory (WST) is a theory for analysis that provides a perspective for understanding systems in organizations, whether or not those systems use IT intensively. WST combines a static view of a work system during a period when it is relatively stable and a dynamic view of how a work system changes over time.

By default a work system is a sociotechnical system in which human participants and/or machines perform processes and activities using information, technology, and other resources to produce specific products/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. Most are not IT systems, however, because they are not about IT.

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

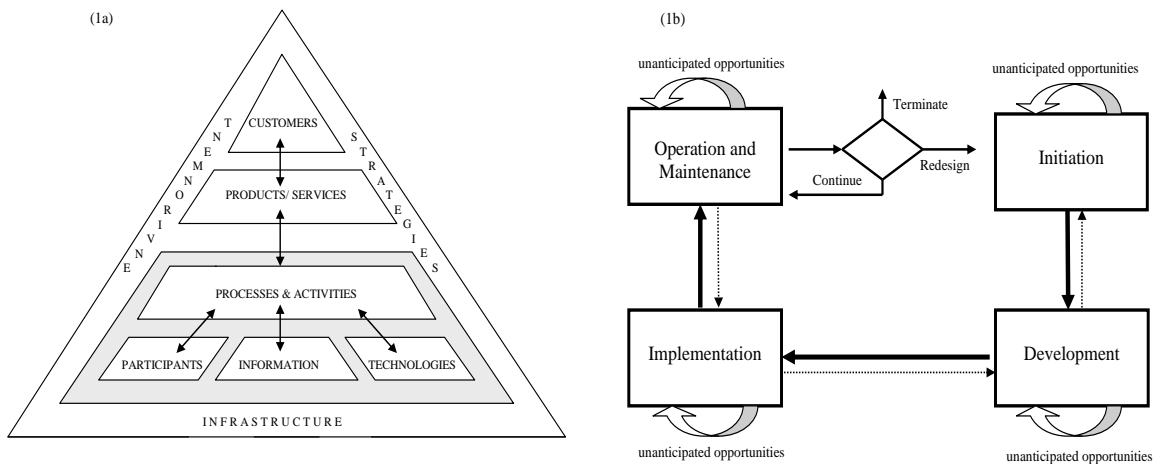
an activity or process that the system must perform" (p. 122), and a nonfunctional requirement is a characteristic of the system other than activities it must perform or support, such as technology, performance, usability, reliability, and security." (p. 123) In contrast, the default view of a work system 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 (Authors, 2012).

**Work system framework.** As explained in Alter (2006, 2008a) 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 products/ services may be partially inside and partially outside because customers often participate and products/services take shape within the work system.
- Environment, infrastructure, and strategies are outside even though they have direct effects within the work system.

### Work System Life Cycle Model

The work system life cycle model (WSLC) is the other central framework in WSM. 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, 2008a, 2008b). 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 length limitations, the WSLC will not be discussed further.



**Figure 1. Work system framework (1a) and work system life cycle model (1b)**

### WORK SYSTEM METHOD

WSM has evolved over many years and through many versions as a flexible systems analysis and design method designed for business professionals concerned with creating or improving work systems. 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 2 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	<ol style="list-style-type: none"> <li>1. Executive summary</li> <li>2. Background</li> <li>3. System and problem</li> <li>4. Analysis and possibilities</li> <li>5. Recommendation and justification</li> </ol>
Appendix 1: Initial summary of the existing work system and the problem or opportunity	<ol style="list-style-type: none"> <li>1. Name of work system</li> <li>2. Main problem or opportunity</li> <li>3. Significance of the work system</li> <li>4. Constraints that limit the possible recommendations</li> <li>5. Performance gaps related to processes, participants, information, or technology</li> <li>6. Performance gaps related to customer perceptions of products and services</li> </ol>
Appendix 2: Summary of the "as is" work system	<ol style="list-style-type: none"> <li>1. Work system snapshot of the "as is" work system</li> <li>2. Customer value and customer concerns (for primary customers)</li> <li>3. Customer responsibilities (for primary customers).</li> </ol>
Appendix 3. Summary of problems, issues, opportunities in the "as is" work system	<p><u>Problems, issues, and opportunities:</u></p> <ol style="list-style-type: none"> <li>1. for the system as a whole</li> <li>2. for each step in the processes or activities</li> <li>3. for specific work system elements (e.g., participants, information)</li> <li>4. for specific types of activities (e.g., information processing, informing, communicating, controlling work, making decisions.)</li> </ol>
Appendix 4: Summary of the recommendations and their likely impacts	<ol style="list-style-type: none"> <li>1. Work system snapshot of the "to be" work system.</li> </ol> <p><u>Likely impact of recommended changes:</u></p> <ol style="list-style-type: none"> <li>2. for the system as a whole</li> <li>3. by step</li> <li>4. related to specific types of activities</li> </ol>

**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

Customers		Products & Services	
<ul style="list-style-type: none"> <li>• Hiring manager</li> <li>• Larger organization (which will have the applicant as a colleague)</li> <li>• HR manager (who will analyze the nature of applications)</li> </ul>		<ul style="list-style-type: none"> <li>• Applications (which may be used for subsequent analysis)</li> <li>• Job offers</li> <li>• Rejection letters</li> <li>• Hiring of the applicant</li> </ul>	
Major Activities and Processes			
<ul style="list-style-type: none"> <li>• <b>Hiring manager</b> submits request for new hire within existing budget</li> <li>• <b>Staffing coordinator</b> defines the parameters of the new position.</li> <li>• <b>Staffing coordinator</b> publicizes the position.</li> <li>• <b>Applicants</b> submit job applications.</li> <li>• <b>Staffing coordinator</b> selects shortlisted applicants.</li> <li>• <b>Hiring manager</b> identifies applicants to interview.</li> <li>• <b>Staffing coordinator</b> sets up interviews.</li> </ul>		<ul style="list-style-type: none"> <li>• <b>Hiring manager</b> and <b>other interviewers</b> perform interviews.</li> <li>• <b>Hiring manager</b> and <b>other interviewers</b> provide feedback from the interviews.</li> <li>• <b>Hiring manager</b> makes hiring decisions.</li> <li>• <b>Staffing assistant</b> sends offer letters or rejections.</li> <li>• Successful <b>applicant</b> accepts or rejects job offer or negotiates further.</li> </ul>	
Participants	Information		Technologies
<ul style="list-style-type: none"> <li>• Hiring managers</li> <li>• Staffing coordinator</li> <li>• Applicants</li> <li>• Staffing assistant</li> <li>• Other employees who perform interviews</li> </ul>	<ul style="list-style-type: none"> <li>• Job requisition</li> <li>• Job description</li> <li>• Advertisements</li> <li>• Job applications</li> <li>• Cover letters</li> <li>• Applicant resumes</li> </ul>	<ul style="list-style-type: none"> <li>• Short list of applicants</li> <li>• Information and impressions from the interviews</li> <li>• Job offers</li> <li>• Rejection letters</li> </ul>	<ul style="list-style-type: none"> <li>• New HR portal that is being built</li> <li>• Word processor</li> <li>• Telephones</li> <li>• Email</li> </ul>

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

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 as to understand systems in organizations. Authors (2012) explains how the introduction of a work system snapshot at the beginning of a previously used textbook example helped novice analysts to 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 early stages of OOAD.

## **WORK SYSTEM METAMODEL**

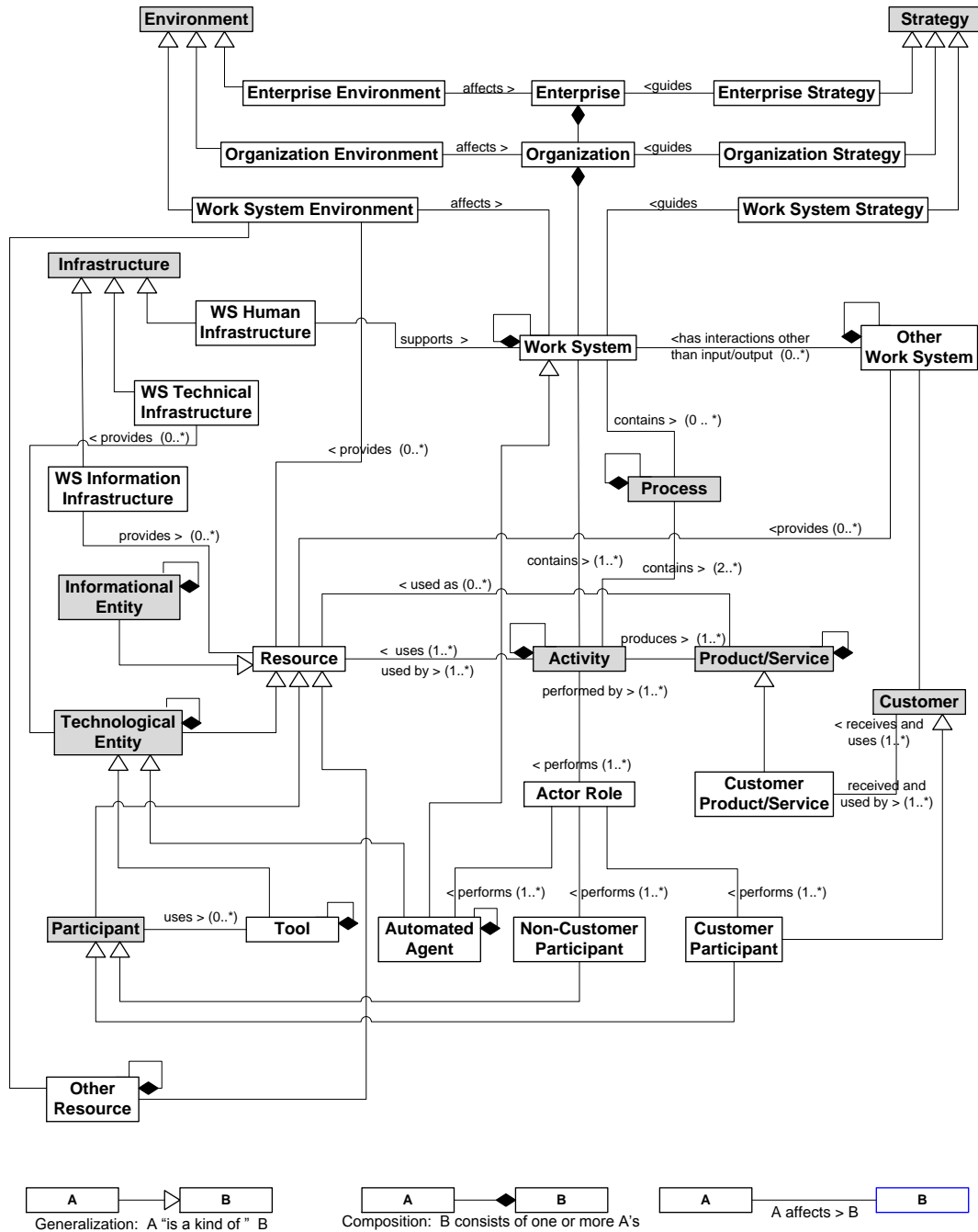
The work system metamodel is a recent extension of WST. Although the work system framework has proven useful for high level overviews, it omits many relationships and details that are quite important. For example, there is no arrow linking participants and technology, which is adequate for systems thinking by most business professionals, but sometimes leads to questions. Both classroom discussions and written assignments produced by MBA and Executive MBA students revealed a number of confusions and 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 about concepts and more specific guidance about important relationships. Ideally, it should support more rigorous analysis without requiring abstruse UML terminology.

The work system metamodel in Figure 2 (Alter 2010, p. 10) is basically a more detailed specification of the work system framework, with each element re-interpreted in a more detailed way. 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 products/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 are less clear 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 2 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 illustrated that type of summary, which is based on the work system framework (Figure 1a).

The metamodel in Figure 3 provides a path for describing the work system in greater depth as a step toward more detailed analysis and design. The lower part of the metamodel (Figure 2) says that a given activity produces products/services by using human, informational, technical, and other resources. That general idea is the basis of the "activity, resources, triggers, and products" (ARTP) summary in Table 3, which is an extension of the work system snapshot in Table 3 and builds on the discussion of "service responsibility tables" in Alter (2008b) and 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* were based on the information section of Table 3 and relatively minor effort to fill in items that were omitted from Table 3. 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 products/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 useful to IT professionals but is in a form that is understandable by business professionals.



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.

Figure 2. 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.

<i>Activity</i>	<i>Actors</i>	<i>Information used</i>	<i>Information created, updated, or deleted</i>	<i>Technology</i>	<i>Trigger</i>	<i>Preconditions</i>	<i>Post conditions (including products/ services produced)</i>
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	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. Activity, resource, 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 3**

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 show the result of the first two steps above. To produce the use case diagram, perform the other two steps above.

**Implication of the two conversion processes.** A direct implication from the mechanical nature of the two conversion processes above is 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 don't care whether information about applicants is partitioned into information about people in general and other information only about 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 UML Diagrams**

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 statechart 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.

**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 of class can be filled in based on general knowledge. More detailed analysis of the situation will probably find additional attributes.

**Activity diagram.** Creating an activity diagram for the entire work system is not totally mechanical, but can be guided as follows. 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.

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.

**Statechart diagram.** For a statechart diagram, start with the domain class diagram previously produced. 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.

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 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 involve experiments that would compare the relative efficacy of use case diagrams versus tabular representations based on work system concepts. The research in this paper does not attempt to demonstrate that the proposed approach is superior to use case diagrams in some general way. It merely demonstrates that the proposed approach may be a plausible alternative for purposes related to 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. (2008a) "Defining Information Systems as Work Systems: Implications for the IS Field." *European Journal of Information Systems* 17,5, pp. 448-469.
- Alter, S. (2008b) "Service System Fundamentals: Work System, Value Chain, and Life Cycle," *IBM Systems Journal* 47,1, pp. 71-85.
- 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*.
- Authors (2012) "Better Use Case Diagrams by Using Work System Snapshots," manuscript currently under review
- 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.)
- 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.
- Dennis, A., Wixom, B.H., and 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. (2004) "How UML is Used," *Communications of the ACM* 49,5, pp. 109-113.
- Dobing, B. and Parsons, J. (2008) "Dimensions of UML Diagram Use: A Survey of Practitioners," *Journal of Database Management*, 19,1,, pp. 1-18.
- Génova, G., Llorens, J., and Quintana, V. (2002) "Digging into Use Case Relationships." *Lecture Notes in Computer Science* (2002), pp. 115-127.
- 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> (Accessed on Mar. 23, 2012)
- Kim, J., Park, S. and Sugarman, V. (2006) "Improving use case driven analysis using goal and scenario authoring: A linguistics-based approach," *Data & Knowledge Engineering*, 58, pp. 21-46.
- 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. and 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.
- 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.
- Tan, X., Alter, S. and Siau, K. (2011) "Using Service Responsibility Tables to Supplement UML in Analyzing e-Service Systems," *Decision Support Systems*, 51,3, June, pp. 350-360.
- Truex, D., Alter, S., and 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.