RELATIONSHIP ANALYSIS: A TECHNIQUE TO ENHANCE SYSTEMS ANALYSIS FOR WEB DEVELOPMENT

Michael Bieber
bieber@njit.edu
http://www.cis.njit.edu/~bieber

Joseph Catanio
JosephCatanio@att.net

New Jersey Institute of Technology, Information Systems Department, College of Computing Sciences

forthcoming in

Web Engineering: Principles and Techniques, Woojung Suh (ed.), Idea Group, 2004

ABSTRACT

A significant aspect of systems analysis and design involves discovering and representing entities and their relationships. Neither structured nor object-oriented analysis techniques provide a formal process to identify relationships in a system being modeled. Existing techniques leave the relationship determination implicit; they are supposed to appear as a byproduct of the other analysis activities. We present a comprehensive, systematic, domain-independent analysis technique, Relationship Analysis (RA), which focuses exclusively on a domain's relationship structure. RA serves three major purposes. First, it helps users, analysts and designers develop a deeper understanding of the application domain through making the relationships explicit. It serves as an effective communication tool for the user and analyst to develop a shared understanding of the domain, and to work out differences in terminology, assumptions and viewpoints. Second, the domains relationships are thoroughly documented utilizing an RA template and an RA diagram. Third, RA results in fuller and richer application analyses and designs. RA significantly enhances the systems analyst's effectiveness, especially in the area of relationship discovery and documentation, which will result in the development of higher quality software applications that consistently meet user needs.

MOTIVATION

A significant aspect of Systems Analysis and Design involves discovering and representing entities and their relationships. There are some informal guidelines (identify nouns, etc.) and tools (Use Cases, CRC cards, etc.) to help with identifying entities or objects. However, no defined processes or templates (for example, in the Unified Process) or diagrams (for example, in UML) exist to explicitly and systematically assist in eliciting relationships or documenting them in Class Diagrams or ER Diagrams [Beraha & Su 1999]. The existing techniques leave the relationship determination implicit; they are supposed to appear as a byproduct of the other analysis activities.

As further evidence, a vital aspect of *hypermedia* system design is identifying relationships and implementing them as links [Fielding et al., 1998]. Yet even in hypermedia design methodologies [Christodoulou et al., 1998, Isakowitz et al., 1995, Koufaris, 1998, Lange, 1994, Schwabe et al., 1996] where links (which represent relationships) explicitly are modeled as "first class objects" (as objects with a set of rich attributes), no technique exists for eliciting relationships/links explicitly during the analysis stage [Yoo & Bieber, 2000b].

A domain's relationships constitute a large part of its implicit structure. A deep understanding of the domain relies on knowing how all the entities or objects are interconnected. Relationships are a key component of vital design artifacts such as ER diagrams and object-class diagrams. These diagrams capture an important, but often rather limited subset of relationships, leaving much of the domain's structure out of the design and thus out of the model of the system. While analyses and models are meant to be a limited representation of a system, we believe the incomplete relationship specification is not by design, but rather from the lack of any methodology to determine them explicitly. Many analyses thus miss aspects of the systems they represent, and often do not convey all the useful information they could when passed on to the designers. It seems that formally and rigorously identifying relationships early on in the development process has not been a primary concern of software engineers in the past.

A rich plethora of relationships surround many objects in the real world. E.g., a product may have several relationships to its customers, who can purchase it, recommend it to others, provide input for modifying it, make comments on it, transform it for other uses, dispose of it, trade it for other goods, etc. Often, a typical analysis would only capture the first of these. Figure 1 presents a subset of the relationships around a book, which one may wish to include, e.g., in a library support application. (The full set would be at least half again as large [Yoo and Bieber 2000b].) Note the presence of multiple relationships among objects.

So, how does one go about discovering the relationships between objects/classes? Is it possible? And once discovered, how does one communicate this discovery to the designer in a formal manner? Relationship Analysis (RA) specifically addresses these concerns and offers solutions that we believe fill a vital gap in systems analysis.

RA provides systems analysts with a systematic and rigorous technique for determining the relationship structure of an application, helping them to discover all potentially useful relationships in application domains and to document them effectively.

RA enhances users', analysts' and system developers' understanding of application domains by broadening and deepening their conceptual model of the domain. Developers can then enhance their implementations by including additional links and other representations of the relationships.

RA can be used either to thoroughly describe an existing application (or information domain) in terms of its relationships, or as part of a systems analysis to understand a new application being designed. It provides a comprehensive technique to perform a systematic analysis for identifying and modeling relationships in a generic domain.

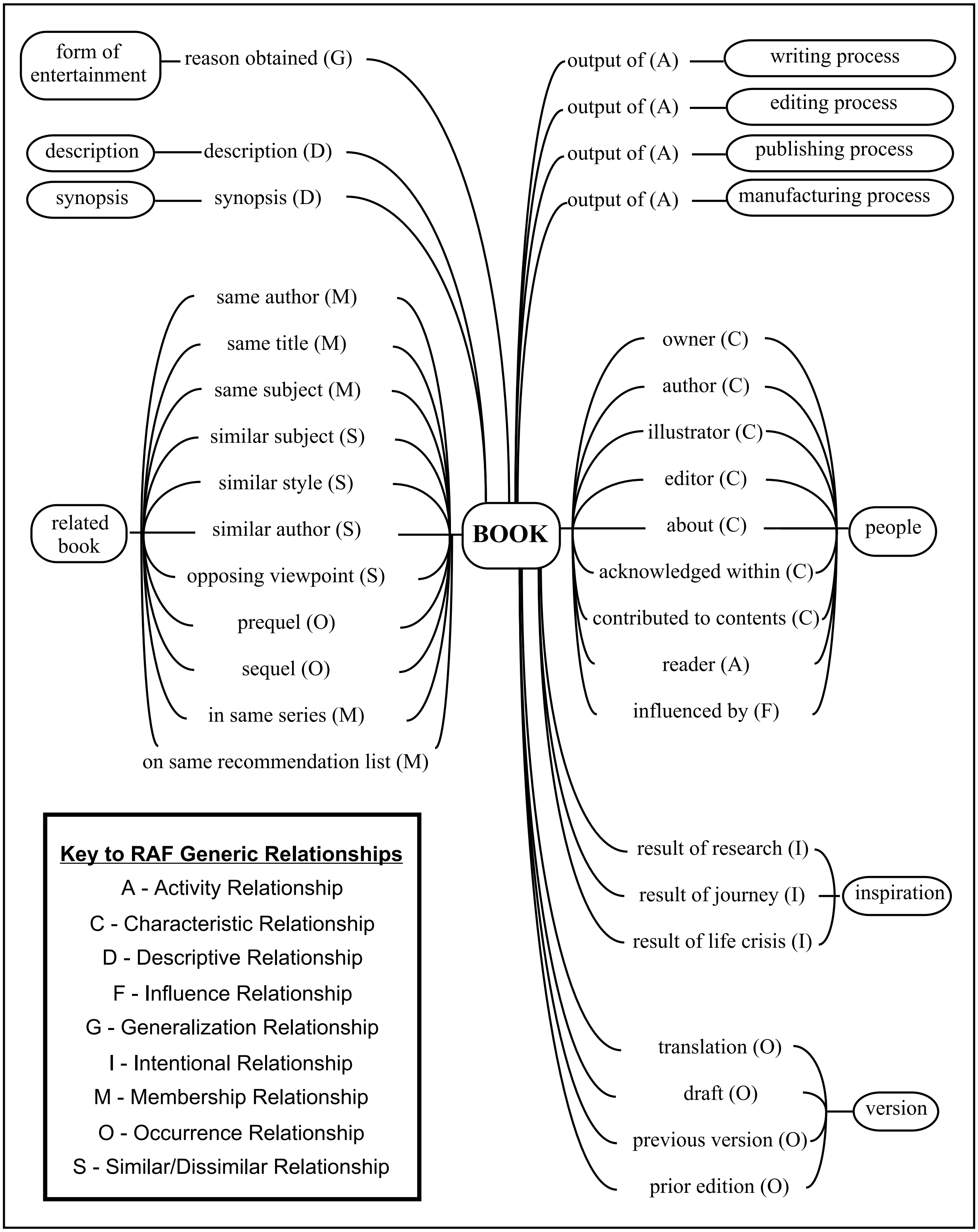

Figure 1. A subset of the relationships around books found through the relationship questions in Table 2

Generic Relationship Taxonomy

These relationship of figure 1 were discovered based on categories of a very extensive literature review [Yoo 2000] and strenuous trial-and-adjustment prototyping. We believe it to be fairly complete. [Yoo 2000] compares RA's taxonomy with 10 other domain-specific taxonomies in detail, with additional comparisons with over 20 others. RA's categories encompass all of these other taxonomies' relationships. This includes, for example, object-oriented analysis [Martin and Odell, 1995] (which provides RA's generalization/specialization, whole-part, classification/instantiation and association relationship classifications).

Generalization/specialization relationships concern the relationships between objects in a taxonomy [Borgida et al., 1984, Brachman, 1983, Smith and Smith, 1977]. Self relationships include characteristic, descriptive, and occurrence relationships.

Whole-part/composition relationships include configuration/aggregation relationships based on configuration aspect of the whole-part relationships, and membership/grouping relationships [Brodie, 1981, Motschnig-Pitrik and Storey, 1995] based on membership aspect of the whole-part relationships [Henderson-Sellers, 1997, Odell, 1994]. Classification relationships connect an item of interest and its class or its instance.

Comparison relationships break down into similar/dissimilar and equivalence relationships, involving such relationships as in thesaurus or information retrieval [Belkin and Croft, 1987, Neelameghan and Maitra, 1978]. Association/dependency relationships break down into ordering, activity, influence, intentional, socio-organizational, spatial and temporal relationships. The term association and dependency could be used interchangeably, because every association involves some concept of dependency [Henderson-Sellers, 1998]. Because association is defined as a relationship that is defined by users, there could be no fixed taxonomy for it. The association relationship taxonomy is fluid compared with other relationships. Current association relationship taxonomy is based on our observations, analyses, ontologies [Mylopoulos, 1998], and the existing classifications [Henderson-Sellers, 1998].

Ordering relationships involve some kind of sequence among items. Activity relationships are created by combining SADT activity diagrams [Mylopoulos, 1998] and case relationships [Fillmore, 1968] to deal with relationships associated with activities or actions abstractly. This relationship could cover any activities that involve input or output, and deal with agents and objects involved in the activities. Influence relationships exist when one item has some power over the other items. Intentional and Socio-organizational relationships could be identified in intentional and social ontologies respectively. Temporal [Allen, 1983, Frank, 1998] and spatial [Cobb and Petry, 1998, Egenhofer and Herring, 1990, Rodriguez et al., 1999] relationships deal with temporal and spatial perspectives, respectively.

Each relationship category can be further broken down into lower levels of detail. [Yoo 2000] details each of these and the literature from which each is derived.

Conducting a Relationship Analysis

RA begins with a stakeholder (role) analysis and "items of interest" (object or entity) analysis. (For the refined technique resulting from the proposed research, use cases will provide this and other contextual information.) For each item of interest identified by the domain expert or user, the analyst asks a series of questions to elicit the relationships around it, which actually often leads to discovering additional elements of interest these connect.

Table 2 gives a series of brainstorming questions that an analyst uses to elicit domain information from the user. Each set of questions is derived from the lower levels of detail for each relationship in the taxonomy, described in [Yoo 2000]. For the purposes of this paper, the questions in Table 2 are rather condensed and highly generic and should be tailored to each item of interest. For example, the descriptive relationship prompts analysts to ask whether an item of interest has "a definition, explanation, set of instructions or illustrations available within or external to the system." (These are all lower-level categories for the generic relationship "descriptive.") The analyst clearly should ask each of the questions individually, and in a way that makes the most sense to the particular domain expert.

Generalization/ Specialization	Is there a broader term for this item of interest? Is there a narrower term for this item of interest?
Characteristic	What attributes and parameters does this item of interest have?
Descriptive	Does an item of interest have a description, definition, explanation, or a set of instructions or illustrations available within or external to the system?
Occurrence	Where else does this item of interest appear in the application domain? What are all uses of this item of interest?
Configuration/ Aggregation	Which components consist of this item? What materials are used to make this item? What is it a part of? What phases are in this whole activity?
Membership/ Grouping	Is this item a segment of the whole item? Is this item a member of a collection? Are these items dependent on each other in a group?
Classification/ Instantiation	Is this item of interest an example of a certain class? If a class, which instances exist for this element's class?
Equivalence	What is this item of interest equal or equivalent to in this domain?
Similar/ Dissimilar	Which other items are similar to this item of interest? Which others are opposite to it? What serves the same purposes as this item of interest?
Ordering	What prerequisites or preconditions exist for this item? What logically follows this item for a given user's purpose?
Activity	What are this item's inputs and outputs? What resources and mechanisms are required to execute this item?
Influence	What items (e.g., people) cause this item to be created, changed, or deleted? What items have control over this item?
Intentional	Which goals, issues, arguments involve this item of interest? What are the positions and statements on it? What are the comments and opinions on this item? What is the rationale for this decision?
Socio-organizational	What kinds of alliances are formed associated with this item of interest? Who is committed to it in the organizational structure? Who communicates with it or about it, under what authority and in which role?
Temporal	Does this item of interest occur before other items? Does this item occur while other items occur?
Spatial	Which items is this item of interest close to? Is this item of interest nearer to destination than other items? Does this item overlap with other items?

Table 2. Sample brainstorming questions emanating from RA's generic relationships

EXPERIMENT

We conducted an experiment to compare RA with other systems analysis techniques. Object-Oriented Analysis (OOA) by Coad and Yourdon was used as the traditional OOA method. The subjects were undergraduate students enrolled in four sections of a software engineering course. Each section served as one group: one control group, one with RA, one with OOA, and one with both techniques. After a training session, the subjects were asked to identify the objects and relationships for an on-line bookstore.

The number of modeling objects plus the number of relationships was used as one of the measures of the output quality. More objects and relationships would indicate deeper understanding of the application and richer representation of the model. Another measure for the quality of output was subjective 1-7 scale judgments by four expert judges. The criteria of the judgment were the extent to which the modeling was relevant to the task and whether the modeling included important entities in the domain. After the experiment the subjects filled out a questionnaire about the usability of the analysis techniques.

The data analysis showed that RA resulted in a significantly higher output quality in terms of number of entities and relationships. The usability score of RA was significantly higher than OOA, which implies that RA is easier to use. The information sufficiency and adequacy of RA was also significantly higher than that of OOA. The results of the experiment confirm that RA can be a powerful and easier to use systems analysis technique. [Yoo, 2000] describes the experiment, analysis and conclusions in detail.

RA LIMITATIONS

While RA was crafted from an extensive literature review, and trial and error revisions, it has no theoretical basis. This opens RA up to two criticisms. First, while we believe it can characterize systems thoroughly, we cannot claim categorically that it's taxonomy is complete. Second, the taxonomy's categories are not distinct enough and relationships sometimes fall under more than one. In part this is because the relationships themselves are interrelated [Yoo 2000], especially within the lower levels. (For example, adjacent items found through the taxonomy's ordering relationship could also be found through the membership relationship if they are in the same group.) However, because RA is a brainstorming technique, it turns out not to matter whether the analyst or user discovers a particular relationship using questions from one category or another. What is important is they found the relationship in the first place.

Another limitation is that while RA has a prescribed order and set of guidelines for conducting the analysis, it has no templates or other well-designed, user-friendly tools to assist in elicitation. All note-taking during the analysis is *ad hoc*. Similarly, no prescribed format exists for recording the results of an RA analysis, including no way to cluster, organize or present the relationships and new objects found. RA simply is not a fully-developed analysis technique. Yet analysts still have found it extremely useful!

EXTENDING RA

In this section we describe our research agenda for extending RA. The proposed research will address the aforementioned limitations with RA and re-develop Relationship Analysis as a complete and fully usable analysis technique that can be integrated with the object oriented analysis methodology by developing the following four major components:

1. *Relationship Analysis Model (RAM)*: A theory-guided taxonomy described below will generate the categories and brainstorming questions, which will help the analyst "discover" all the possible relationships among objects and classify these.
2. *Relational Analysis Template (RAT)*: A form designed to capture elicited knowledge about the domain.
3. *Relationship Analysis Diagram (RAD):* A new design tool to help the analyst "formally" document all the discovered relationships and aid in communicating it to the designer who will, in turn, use it as the input to create the class diagram.
4. *Relationship Analysis Process (RAP):* A formal process to facilitate relationship discovery and documentation.

(1) Relationship Analysis Model: Theoretical Basis

We intend to develop a new relationship taxonomy grounded in theory. We have preliminarily chosen Guilford's Structure of Intellect (SI) theory [Guilford 1956, 1967, 1971, 1982] as the basis of our taxonomy.

SI is a general theory of human intelligence. SI has formed the basis for comparing and classifying the complete range of tests for intellectual ability. Guilford designed SI with a focus on measuring creativity [Guilford 1950], which is an integral aspect of the systems analysis and brainstorming activities in general. Because RA is a brainstorming elicitation technique, we believe that SI will help the analyst and user thoroughly explore a domain in a way that fits the way people conceptualize.

Thus we believe that a SI foundation will allow us to develop a complete taxonomy of relationships from a cognitive, human intellect viewpoint. Of course, not all relationships within a computer application domain have something to do with human intellect. But because SI is a complete taxonomy, we believe it will enable analysts to *elicit* as complete a set of relationships (and associated objects), as possible, within application domains.

The SI model classifies intellectual abilities into a cross-classification independent three-plane system comprised of contents, products, and operations [Guilford 1956].

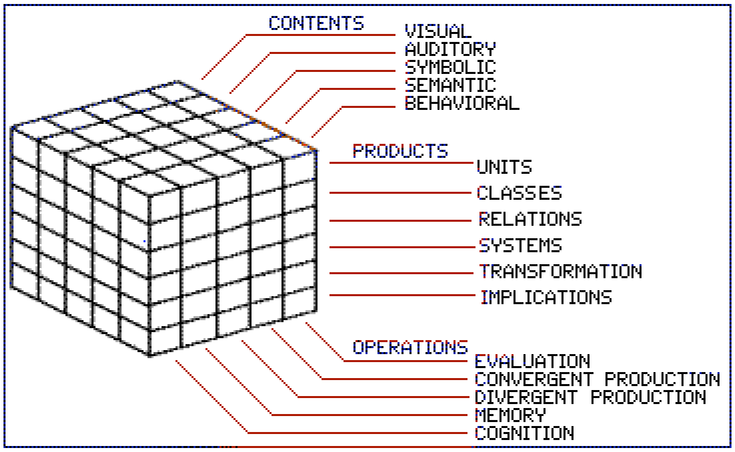

Figure 2 Guilford's Structure of Intellect Model

Figure 2 shows SI includes five kinds of contents, six kinds of products, and five kinds of operations. Due to the three independent planes, there are theoretically 150 different components of intelligence. The three dimensions of the model specify first, the operation, second the content, and third, the product of a given kind of intellectual act. Every intellectual ability in the structure is characterized in terms of the type of operation employed, the content involved, and the sort of resulting product. The convention (Operations, Contents, Products) is used to specify each factor. For example, (Cognition, SeMantic, Unit) or (CMU) represents cognition of a semantic unit. In this way the SI theory represents the major kinds of intellectual activities or processes as an interrelated three-dimensional model.

Turoff et al. apply SI to the computer application domain [Turoff et al., 1991] and argue that not all of the SI components are necessary for classifying computer application domains, they reduce it to two dimensions by classifying all SI types of content as one, namely semantic. The four SI contents; visual, auditory, symbolic, and behavioral are useful in classifying tests of intellect, but are not necessary for classifying application domains. In addition, the SI operations, evaluation and memory are also not necessary for classifying application domains [Turoff et al., 1991].

Extending from these aforementioned models, the Relationship Analysis Model (RAM) approach in classifying relationships of computer application domains is to develop a semantic classification model. Therefore, the resulting model is a two-dimensional model, products vs. operations.

A product represents the organization that information takes in the analyst’s processing of it [Guilford, 1967] [Meeker, 1969].

- Units – Most basic item. Things to which nouns are normally applied. Described units of information.
- Classes – Sets of items of information grouped by virtue of their common properties.
- Relations – Connections between items of information based on variables or points of contact that apply to them.
- Systems – Organized or structured aggregates of items of information.
- Transformations – Changes, redefinition, shifts, or modifications of existing information or in its function.

- Implications – Extrapolations of information. Emphasizes expectancies, anticipations, and predictions.

Operations represent major kinds of intellectual activities or processes that analysts perform with information [Guilford, 1967] [Meeker, 1969].

- Cognition – Discovery, awareness, or recognition of information by comprehension or understanding. Guilford views the cognition process as the classification of an object. Turoff et al. extend this concept to hypertext whereby cognition is represented by a node that classifies all the linked objects as related to a common concept or characteristic. Hypertext, at its core, concerns nodes (elements-of-interest) and links (relationships). These links or relationships among nodes are classified under convergent and divergent production properties. The RAM differentiates itself from the HMM in its application of cognition. The HMM represents cognition by a node and in hypertext terms: a node is an endpoint, and relationships exist among nodes or endpoints. In contrast, the relationships of each element-of-interest in the RAM represent by six cognitive focus perspectives.
- Convergent Production – Generation of information from the given information, where the emphasis is on achieving unique best outcomes. The given information fully determines the response. Guilford views convergent production as when the input information is sufficient to determine a unique answer. Turoff et al. extend this concept and a convergent link is a relationship that follows a major train of thought. This is referred to as a convergent relationship in the RAM.
- Divergent Production - Generation of information from the given information, where the emphasis is on variety and quality of output from the given information. Guilford views divergent production as fluency of thinking and flexibility of thinking. Turoff et al. extend this concept and a divergent link is a relationship that starts a new train of thought. This is referred to as a divergent relationship in the RAM.

The Relationship Analysis Model (RAM) applies these three operations to the six products defined in the previous section to categorize relationships. Similar to Turoff's Hypertext Morphology Model (HMM), each cognitive product becomes a focus point that classifies all the linked relationships pertaining to the particular cognitive focus. Thus, relationships of an element of interest are described by six cognitive focal points. Relationships of each focal point are classified under convergent and divergent operation properties. Therefore it is possible to classify the relationships of an element of interest in terms of six products each of which has convergent and divergent relationships. Table 3 depicts the model.

Cognition Focus	Convergent Relationship	Divergent Relationship
Unit	Specification	Elaboration
Collection	Membership	Aggregation
Comparison	Equivalence	Opposition
System	Path	Branch
Transformation	Modify	Transpose
Implication	Influence	Extrapolate

Table 3 Relationship Analysis Model (RAM)

Developing RA gave us the experience of developing brainstorming questions from relationship categories. We expect that the types of questions we shall develop using SI to be similar in spirit to those in Table 2. Turoff et al. provide several synonyms for each node and link category, which can form the basis of RA's corresponding set of questions. One difference is that the node synonyms could underlie additional brainstorming questions, whereas RA only had questions based on relationships. Node-based questions may pose a useful extension for RA.

(2) Relationship Analysis Templates

Based on our experience with RA, several kinds of useful information come to light during the elicitation process. These include relationships, characteristics (metadata) about the relationships, new objects (at the other end of the relationships), characteristics about these new objects, characteristics of the object being focused upon for relationship elicitation, as well as general comments reflecting insight into context, terminology, assumptions and viewpoints.

The Relationship Analysis Templates will have areas for recording each of these, as well as a place for recording comments. We may find it useful to provide another form for capturing the latter contextual information that arises from the focused communication between analyst and user, which RA provides.

(3) Relationship Analysis Diagrams

We envision the Relationship Analysis Diagrams to look somewhat similar to Figure 1. Each diagram will show all the relationships, metadata and prioritizations (see below) around a single object-of-interest (or for complex cases, perhaps split a single object's relationships and metadata among several sub-diagrams). One issue is how busy the diagram may become. We may need to prototype several versions before determining the most useful format.

Relationship Analysis Diagrams are the final output from RA, and a primary input into the systems design phase. Through prototyping and revisions we will determine whether (and how) to include all metadata (and relevant comments) gathered on the templates with the diagrams. Perhaps a version of the templates should accompany each diagram for the subsequent design phase.

(4) Relationship Analysis Process

We shall develop and refine a fully-useable Relationship Analysis Process (RAP) for conducting a Relationship Analysis. We believe it will encompass the following three stages, though these are open to refinement based on the evaluation described later.

(a) Context Analysis: The analyst starts with one or more use cases. This provides the background (context, actions and functional requirements) as well as a starting set of objects.

(b) Relationship Elicitation: The analyst will work together with the users to elicit the domain relationships derived from the new Structure of Intellect-based taxonomy. The analysis will use the new Relationship Analysis Templates to ask appropriate brainstorming questions and record elicited information. The elicitation will produce a Relationship Analysis Diagram for each object showing all its relationships to other elements. We also need to develop accompanying full guidelines for conducting this analysis, completing the RA Templates and drawing the RA Diagrams.

(c) Prioritization: The analyst and users should feel cognitively unbounded during the Relationship Elicitation stage, in order to come up with a comprehensive map of the domain relationships [Gause & Weinberg 1989]. While very useful for understanding the domain fully, in practice the designer may need to prune the relationships in the subsequent systems design phase. Some relationships may be unnecessary to the final application; others may be too costly or difficult to implement. To help the designer in these decisions, the analyst and user work together to prioritize each element (relationship, object, metadatum) in the Relationship Analysis Diagram. To motivate the user to prioritize, he or she could be told that the designer may need to constrain the number of relationships (and objects) for budgetary reasons. They then assign each a ranking between 1 and 5, where 5 is the most important and should be implemented if at all possible, and 1 is the least important and can be left out of a final design with no detriment. This will provide important feedback to the designer as to the importance of each element in the diagram.

DISCUSSION: INTEGRATION INTO CURRENT ANALYSIS TECHNIQUES

The research and solutions we propose here can be seamlessly integrated into both current object-oriented (OO) and structured analysis processes to fill the vital gap of identifying relationships.

Object-oriented analysis techniques like Unified Process (UP) and Unified Modeling Language (UML) certainly provide real benefits to the critical early stages of application development. A formal process and support to identify and document all relationships of interest in a domain, however, is not one of them. The UML depicts the interactions between the use-cases and the actors utilizing use-case diagrams. Subsequently, class diagrams are developed to depict the relationships between the classes that implement the use-cases. We believe that a step is missing and that the transition is too abrupt.

This also is the case with the structured analysis method. One of the most popular analysis tools used in structured analysis to capture relationships is the Entity Relationship (ER) diagram. Although an excellent technique for portraying the resulting relationships in a domain, just as with OO class diagrams, no formal techniques exist for identifying the relationships to include.

Thus, existing techniques leave the relationship determination implicit. Relationship analysis fills this void by providing a systematic technique to determine the relationship structure of an application. Relationship analysis (RA) is geared towards discovering and representing entities and their inter-relationships. The relationship analysis process (RAP) provides a relationship analysis diagram (RAD) that explicitly depicts these discovered relationships using the standard Unified Modeling Language (UML) notation. The RAP can be integrated into the UML technique between the use-case and class diagram identification steps. Thus, RA adds a step to the UML process but provides a technique to explicitly determine and depict the application's relationship structure, thereby enhancing the UML.

CONCLUDING NOTES

We begin this concluding section by summarizing some of the things that RA is not.

RA is not a design technique. Rather it is a method-independent analysis technique, which provides useful input to the systems design phase.

RA does not provide algorithms to generate relationships. RA is an elicitation technique embodied in a systematic procedure (RAP) to support the analysis phase. In follow-on research we hope to investigate automatic generation of design documents from the analysis documentation.

RA and the associated support tools presented here are intended to provide a high degree of support to the analyst and NOT to replace the analyst by totally automating the relationship discovery and documentation process. There can be no substitute to the quality and expertise provided by the human analyst. However, we believe that RA and the corresponding support mechanisms can significantly enhance the effectiveness of the human analyst. Contributions and Potential Impact

This research addresses a major shortcoming in today's analysis techniques. Neither structured nor object-oriented analysis techniques provide a formal process to identify relationships in a system being modeled. RA is the only systematic, domain-independent analysis technique focusing exclusively on a domain's relationship structure. RA will provide a theoretically-based procedure and tools for conducting a systematic analysis.

RA serves two major purposes. First, it helps users, analysts and designers develop a deeper understanding of the application domain (through making the relationships explicit). Second, RA should result in fuller and richer application analyses and designs.

RA also provides the analyst with another tool for working with the user to better understand the application domain. Because of its brainstorming/elicitation approach, RA should serve as an effective communication tool for the user and analyst to develop a shared understanding of the domain, and to work out differences in terminology, assumptions and viewpoints. RA will provide a foundation for users and system analysts to communicate throughout systems analysis process.

We expect that RA will become an invaluable tool in the toolkit of the analyst irrespective of the software engineering approach taken during analysis. Since RA is methodology-independent, it should be equally effective in identifying relationships between entities when using the traditional structured approach to analysis and identifying relationships between objects using object-oriented methodologies. RA could very easily become a standard extension to the other tools and techniques currently available for analysis. While the analyst is working with the user in creating use cases to understand the functionality required of the system, e.g., he or she also could be conducting RA and documenting it as part of the elicitation process.

Some object-oriented "gurus" hold that spending too much effort in trying to identify relationships is counter-productive. E.g., while discussing guidelines to creating domain models, Larman [2002] states:

"Associations are important, but a common pitfall in creating domain models is to spend too much time during investigation trying to discover them... Too many associations tend to confuse a domain model rather than illuminate it. Their discovery can be time-consuming, with marginal benefit."

We address these concerns by providing the tools and techniques to make an extensive relationship analysis useful and practical. We believe that using RA will produce a richer understanding of relationships in less time than the comparable informal processes currently followed. Further, our prototyping of the tools will address whether a plethora of relationships tends to confuse or enlighten. Finally, our evaluation should show that RA significantly improves the software development process.

One thing that became clear from using RA was that many applications (with and without Web interfaces) had many fewer links that users would find useful. This occurs for several reasons [Bieber and Vitali, 1997; Bieber and Yoo, 1999]. Few analysts explicitly think in great detail about their applications' interrelationships. In part, few existing applications have a rich link structure that could be an example for analysts and designers. In part, few tools exist that help system developers to think of an application in terms of its relationships [Bieber, 1998]. Until the advent of recent World Wide Web standards such as XLINK, Web browsers did not support the easy display of multiple links from a single link anchor (e.g., underlined blue text in Netscape). With time, this now will become more commonplace. We believe that RA will provide the tools and help change the mindset of analysts and designers to include multi-headed links in applications.

RA will significantly enhance the systems analyst's effectiveness, especially in the area of relationship discovery and documentation, which will result in the development of higher quality software applications that consistently meet users' needs.

ACKNOWLEDGEMENTS

We gratefully acknowledge partial funding support for this research by the Alfred P. Sloan Foundation, the NASA JOVE faculty fellowship program, the United Parcel Service, the New Jersey Center for Multimedia Research, the National Center for Transportation and Industrial Productivity at the New Jersey Institute of Technology (NJIT), the New Jersey Department of Transportation, the New Jersey Commission of Science and Technology, the National Science Foundation under grants EISA-9818309 and DUE-0226075, and NJIT's SBR program.

REFERENCES

[Allen, 1983] Allen, J. "Maintaining Knowledge about Temporal Intervals," *Communication of the ACM,* 26(11), 1983, 832-843.

[Belkin and Croft, 1987] Belkin, N. and Croft, W. (1987) "Retrieval Techniques," *Annual Review of Information Science and Technology (ARIST),* Vol. 22, 1987, Chapter 4, pp 109-131.

[Beraha & Su 1999] Beraha, S. and Su, J., 1999. "Support for modeling relationships in object-oriented databases," *Data & Knowledge Engineering,* vol. 29 no. 3, 1999, pp 227-257

[Bieber 1998] Bieber, M. "Hypertext and Web Engineering," *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia,* ACM Press, 277-278, 1998.

[Bieber 2001] Bieber, M. "Supplementing Applications with Hypermedia," Technical Report, IS Department, NJIT, 2001.

[Bieber and Vitali 1997] Bieber, M. and Vitali, F. "Toward Support for Hypermedia on the World Wide Web," *IEEE Computer,* 30(1), 1997, 62-70.

[Bieber and Yoo 1999] Bieber, Michael and Joonhee Yoo "Hypermedia: A Design Philosophy," ACM Computing Surveys, 31(4es), 1999.

[Booch et al. 1999] Booch, G., Rumbaugh J. and Jacobson, I., 1999. *The Unified Modeling Language User Guide*, Addison-Wesley, MA

[Borgida et al., 1984] Borgida, A., Mylopoulos, J., and Wong, H. "Generalization/Specialization as a Basis for Software Specification," *On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases, and Programming Languages,* Ed. Brodie, M., Mylopoulos, J., and Schmidt, J., pp 87-117.

[Brodie, 1981] Brodie, M. (1981) Association:A Database Abstraction for Semantic Modelling. Entity-Relationship Approach to Information Modeling and Analysis, P.P. Chen (ed.), ER Institute 1981, 583-608.

[Brachman, 1983] Brachman, R. "What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks," *IEEE Computer*, October 1983, pp 30-36.

[Christodoulou et al., 1998] S. Christodoulou, G. Styliaras and T. "Papatheodourou, Evaluation of Hypermedia Application Development and Management Systems," *Proceedings of ACM Hypertext '98 Conference*, Pittsburgh, 1998, 1-10.

[Coad & Yourdon 1991] Coad, P. and Yourdon E., 1991. *Object-Oriented Analysis*. Yourdon Press, Englewood Cliffs, NJ

[Cobb and Petry, 1998] Cobb, M. and Petry, F. "Modeling Spatial Relationships within a Fuzzy Framework," *Journal of the American Society for Information Science*, 49(3), pp 253-266, 1998.

[Egenhofer and Herring, 1990] Egenhofer, M. and Herring, J. "Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases," Technical Report, Department of Surveying Engineering, University of Maine.

[Fielding et al., 1998] Fielding, R., Whitehead JR., E. J., Anderson, K., Bolcer, G., Oreizy, P., and Taylor, R. (1998) Web-Based Development of Complex Information Products. Communications of the ACM, 41(8), 84-92.

[Fillmore, 1968] Fillmore, C.J. "The case for case" in Universals in Linguistic Theory.

[Fowler & Scott 2000] Fowler, M., and Scott, K. 2000. *UML distilled: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley, MA

[Frank, 1998] Frank, A. "Different Types of "Times" in GIS," *Spatial and Temporal Reasoning in Geographic Information Systems*, 1998, Eds. Egenhofer, M. and Golledge, R. Chapter 3, pp 41-62.

[Gause & Weinberg 1989] Gause, D. C. and Weinberg, G. M. *Exploring Requirements: Quality before Design*, Dorset House Publishing: New York, 1989.

[Guilford, 1950] Guilford, J.P. (1950). *Creativity.* American Psychologist, 5, 444-454.

[Guilford, 1956] Guilford, J.P. (1956). *The Structure of Intellect.* Psychological Bulletin 53(4), 267-293.

[Guilford, 1967] Guilford, J.P. (1967). *The Nature of Human Intelligence.* New York: McGraw-Hill.

[Guilford & Hoepfner, 1971] Guilford, J.P. & Hoepfner, R. (1971). *The Analysis of Intelligence.* New York: McGraw-Hill.

[Guilford, 1982] Guilford, J.P. (1982). "Cognitive psychology's ambiguities: Some suggested remedies," *Psychological Review,* 89, 48-59.

[Henderson-Sellers, 1997] Henderson-Sellers, B. "OPEN Relationships-Compositions and Containments," *Journal of Object-Oriented Programming,* November/December 1997, pp 51-72.

[Henderson-Sellers, 1998] Henderson-Sellers, B. "OPEN Relationships-Associations, Mappings, Dependencies, and Uses," *Journal of Object-Oriented Programming*, February 1998, pp 49-57.

[Isakowitz et al., 1995] Isakowitz, T., Stohr, E. and Balasubramanian, P. (1995) RMM: A methodology for structuring hypermedia design. Communications of the ACM 38(8), 34-44.

[Jacobson et al. 1999] Jacobson, I., Booch G., and Rumbaugh, J.1999. *The Unified Software Development Process,* 1999, Addison-Wesley, MA

[Jarvenpaa et al., 1988] Jarvenpaa, Sirkka L.; Rao, V. Srinivasan; and Huber, George P. "Computer support for meetings of groups working on unstructured problems: A Field experiment," *MIS Quarterly*, December 1988, pp. 645-666.

[Koufaris, 1998] Koufaris, M. (1998) Structured design of WWW and Intranet applications. The Seventh International WWW Conference Tutorial.

[Lam, 1997] Lam, Simon S.K. "The effects of GDSS and task structures on group communication and decision quality," *Journal of Management Information Systems*, Spring 1997, vol. 13, no.4, pp. 193-215.

[Lange, 1994] Lange, D. (1994) An Object-Oriented Design Method for Hypermedia Information Systems. Proceedings of the Twenty-seventh Annual Hawaii International Conference on System Sciences, 1994, 366-375.

[Larman, 2002] Larman, C., 2002. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and the Unified Process*. Prentice Hall, Englewood Cliffs, NJ

[Martin and Odell, 1995] Martin, J. and Odell, J. (1995) *Object-Oriented Methods: A Foundation*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[Martin & Odell 1998] Martin, J. and Odell, J. (1998) *Object-Oriented Methods: A Foundation, UML Edition*, Prentice Hall, Englewood Cliffs, NJ

[McClave & Benson, 1988] McClave, James T.; and Benson, P. Geroge, *Statistics for Business and Economics*, Dellen, 1988.

[Motschnig-Pitrik and Storey, 1995] Motschnig-Pitrik, R. and Storey, V. "Modelling of set membership: The notion and the issues," *Data & Knowledge Engineering* 16 (1995), 147-185.

[Mylopoulos, 1998] Mylopoulos, J. "Information Modeling in the Time of the Revolution," *Information Systems,* vol. 23, No. 3/4, pp 127-155, 1998.

[Neelameghan and Maitra, 1978] Neelameghan, A. and Maitra, R. "Non-hierarchical associative relationships among concepts: Identification and Typology," *Part A of FID/CR report no. 18*, Bangalore: FID/CR Secretariat Document Research and Training Center.

[Ocker et al., 1998] Ocker, Rosalie; Fjermestad, Jerry; Hiltz, Starr Roxanne; and Johnson, Kenneth, "Effects of four modes of group communication on the outcomes of software requirements determination," *Journal of Management Information Systems*, Summer 1998.

[Odell, 1994] Odell, J. "Six different kinds of composition," *Journal of Object-Oriented Programming,* January 1994, pp 10-15.

[Rao & Turoff, 1990] Rao, Usha, & Turoff, Murray (1990). "Hypertext Functionality: A Theoretical Framework," *International Journal of Human-Computer Interaction*, 4(2), 333-358.

[Rodriguez et al., 1999] Rodriguez, M., Egenhofer, M. and Rugg, R. "Assessing Semantic Similarities Among Geospatial Feature Class Definitions," *Interop '99*, Zurich, Switzerland, in: A. Vckovski (editor), Lecture Notes in Computer Science, New York.

[RUP 2001] *The Rational Unified Process online documentation*, Rational Corporation

[Schwabe et al. 96] Schwabe, D., Rossi, G., and Barbosa, S. "Systematic hypermedia application design with OOHDM," *Proceedings of Hypertext '96*, Washington DC, pp 116-128.

[Shaft & Vessey, 1998] Shaft, T. M. and Vessey, Iris, "The Relevance of Application Domain Knowledge: Characterizing the Computer Program Comprehension Process," *Journal of Management Information Systems*, Vol. 15 No. 1, Summer 1998, pp. 51-78.

[Siau et al., 1997] Siau, K.; Wand, Y.; and Benbasat, I., "The Relative Importance of Structural Constraints and Surface Semantics in Information Modeling," *Information Systems*, Vol. 22 No. 2/3, 1997, pp. 155-170.

[Smith and Smith, 1977] Smith, J. and Smith, D. "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems,* 2(2), June 1977, pp 105- 133.

[Straub, 1989] Straub, Detmar W., "Validating Instruments in MIS Research," *MIS Quarterly*, June 1989, pp. 147-169.

[Turoff et al., 1991] Turoff, M., U. Rao and S. R. Hiltz (1991). "Collaborative Hypertext in Computer Mediated Communications," *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, Volume IV, 357-366.

[Yoo 2000] Yoo, Joonhee. "Relationship Analysis," *Ph.D. Dissertation*, Rutgers University, 2000.

[Yoo and Bieber 2000a] Yoo, Joonhee and Michael Bieber, "Towards a Relationship Navigation Analysis," *Proceedings of the 33rd Hawaii International Conference on System Sciences,* IEEE Press, Washington, D.C., January 2000.

[Yoo and Bieber 2000b] Yoo, Joonhee and Michael Bieber, "A Relationship-based Analysis," *Hypertext 2000 Proceedings*, San Antonio,ACM Press, June 2000.