Chapter IV

Relationship Analysis:  
A Technique to Enhance Systems Analysis for Web Development

Joseph Catanio  
LaSalle University, USA

Michael Bieber  
New Jersey Institute of Technology, USA

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 by-product 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 by-product of the other analysis activities.

As further evidence, a vital aspect of hypermedia system design is identifying relationships and implementing them as links (Fielding, Whitehead, Anderson, Bolcer, Oreizy, & Taylor, 1998). Yet even in hypermedia design methodologies (Christodoulou, Styliaras, & Papatheodourou, 1998; Isakowitz, Stohr, & Balasubramanian, 1995; Koufaris, 1998; Lange, 1994; Schwabe, Rossi, & Barbosa, 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. For example, 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, such as in a library support application. (The full set would be at least half again as large [Yoo & 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.
Related Content

Impact of Internet Usage in Saudi Arabia: A Social Perspective
[www.igi-global.com/article/impact-internet-usage-saudi-arabia/2628?camid=4v1a](www.igi-global.com/article/impact-internet-usage-saudi-arabia/2628?camid=4v1a)

MCWDF: Micro Chunk Based Web Delivery Framework
[www.igi-global.com/article/mcwdf/193006?camid=4v1a](www.igi-global.com/article/mcwdf/193006?camid=4v1a)

Leadership, Collective Efficacy and Team Performance: A New Paradigm for the Next Generation
[www.igi-global.com/chapter/leadership-collective-efficacy-team-performance/68635?camid=4v1a](www.igi-global.com/chapter/leadership-collective-efficacy-team-performance/68635?camid=4v1a)

From Computer-Mediated Colonization to Culturally Aware ICT Usage and Design
[www.igi-global.com/chapter/computer-mediated-colonization-culturally-aware/4950?camid=4v1a](www.igi-global.com/chapter/computer-mediated-colonization-culturally-aware/4950?camid=4v1a)