Domain Analysis and Information Engineering: Promoting a Combined Attack on Stovepipe Systems

Roberta G. Burdick

Unisys Corporation
Government Systems Group
12010 Sunrise Valley Drive, Dept. 7670
Reston, Va. 222091
Tel: (703)620-7434
Fax: (703)620-7916
Email: rburdick@stars.reston.paramax.com

Abstract

The call for improved software quality and productivity and reduced development and maintenance costs has become a familiar battle cry, used to justify an ever-expanding number and variety of increasingly specialized disciplines, methods and tools: data administration, business process reengineering, business reengineering, information engineering, object orientation, and domain analysis, to name but a few. Although some of these disciplines did not originally focus on reuse, each is gradually coming to recognize that reuse is central to their efforts to improve software quality and productivity.

However, professionals in any of these disciplines often work with little understanding or connection to the others. Therefore we are unable to collaborate on our separate contributions to the same underlying problems. As a result, we are far from agreement on how to achieve reuse, or even what reuse means.

Ideally, to realize the promised benefits of reuse adoption, an alliance should be arranged between the relevant disciplines. This paper is proposing to start this alliance with an arranged marriage of Domain Analysis (DA) and Information Engineering (IE). It discusses the "stovepipe domain" problems that this marriage would solve. It describes the common and complementary traits of both "parties" that should lead to marital success. Finally, it explains why the religious centers of semantic modeling and/or object orientation, which represent a set of values and practices already shared to some extent by DA and IE, may offer an ideal support framework for this marriage.

Keywords: reuse, reuse planning, domain analysis, information engineering, interoperability, domain scoping, stovepipe domains, component, variant

Workshop Goals: Exchange of ideas and experiences with other researchers and practitioners; brokering the marriage proposed above.

Working Groups: Reuse, integration, and interoperability (suggestion); Reuse management, organization and economics; Domain analysis and engineering; Reuse and OO methods

1 Background

Most of my 16 year career has been focused on management information systems (MIS) integration problems. My work has consisted of identifying, analyzing, repairing, working around, and preventing information inconsistencies by addressing the problem of unplanned inter-system redundancy. For this reason, .I have chosen to investigate, and participate in the evolution and adoption of, several disciplines with an eye toward reuse. My responsibilities have included practitioner, technology transfer agent, methodology integrator, and applied researcher. I have been a data administrator (Mobil Oil), an information center coordinator (Mobil Oil, World Bank), an information engineering, object-orientation, and business reengineering consultant and methodologist (World Bank, James Martin and Company, Syrinx), and a domain analysis researcher and technology transfer agent (Unisys - STARS).

While at JM&Co., I authored a paper integrating Information Engineering and Object Orientation entitled "Information Asset Management and Reuse." As a staff engineer on the Unisys STARS Reuse team, I have co-authored our Software Engineering Environment (SEE) "Reuse Product Plan" and evaluated several candidate SEE tools for reuse support. I am currently planning and analyzing the semantic and architectural requirements for fine- grained integration of CTA's KAPTUR [1], Unisys's Reuse Library Facility (RLF), and IDE's Software through Pictures (StP) tools to improve reuse support for our STARS demonstration project. I am also participating in the formulation of our upcoming guide to Reuse Oriented Software Engineering (ROSE), an instantiation of the CFRP [2].

2 Position

2.1 Introduction

Neither Domain Analysis nor Information Engineering is itself a monolithic discipline. Each term describes families of disciplines with both commonality and variability among its members. In this sense, both DA and IE are themselves domains, worthy of more formal study to compare, contrast, and classify the thinking behind their various branches. However, the objective of this paper is to stimulate further research and discussion on the merits and methods of DA/IE integration; not to expand on the distinctions among the various DA and IE schools. To this end, the following discussion exploits simplifying generalizations.

2.2 DA and IE: Commonality

Domain Analysis and Information Engineering have much in common. Both trace their lineage to the ubiquitous quest for improving software quality and productivity, while cutting development and maintenance costs. Both point to planned and managed reuse as a central tenet. Both stress the importance of models and architectures that identify and codify relationships among multiple systems, and that systematically partition that knowledge along several dimensions: problem vs solution space, specialization vs aggregation hierarchies, as-is vs to-be models, etc. Both implement their architectures using templates, component- based and generative techniques; both use their architectures to guide the creation of reusable assets, and the composition of assets into systems.

In addition, branches of both disciplines state emphatically (especially when not in marketing situations) that the successful transition from the "one system at a time" mentality requires not only technological but cultural and organizational change [3, 4, 2]. Both promote technology transfer, and recommend small modeling teams, guided by modeling experts but staffed by professionals who collectively represent expertise and stakes in all aspects of the area under study. Finally, both disciplines emphasize process as well as product, stress the importance of planning and learning, and have developed remarkably similar frameworks for the development of a new kind of supporting infrastructure.

2.3 DA and IE: Vive la Difference

The key differences between Domain Analysis and Information Engineering are first of all differences in defining the problem scope. These scope differences reflect the origins of DA and IE in different client communities, and their consequent focus on different but related problem spaces.

The optimum scope of an Information Engineering initiative is an entire enterprise. Ideally, sponsorship of the IE initiative should reside with the enterprise's chief information officer or equivalent [5] (in practice, narrower sets of interrelated functions are often studied initially, with significant benefit). The scope of the initial planning project is broad and shallow. This results in multiple, carefully scoped candidate follow-on projects; each is increasingly narrow in scope and deep in its level of detail.

IE, which spearheaded the "enterprise modeling" discipline, was developed in response to the integration problems facing Management Information System (MIS) developers and the collaboration problems facing their end-users. Coordination, collaboration, and information exchange among separate business activities (e.g. accounting, payroll, benefits, personnel, etc.) is vital to the effectiveness of a complex business enterprise. However, the separate systems supporting each of these business activities tend to be designed and developed in isolation. Frequently, each system is designed for use by customers representing a single organization within the enterprise. Because these systems function as stand- alone vertical slices through the enterprise, often impeding rather than supporting collaboration, they are sometimes described as "stovepipe systems".

The gaps and overlaps among these "stovepipe systems" present both their maintainers and end users with enormous difficulties. Their shared information is often defined and captured in multiple systems and files. These redundant and inconsistent stores and definitions escalate system life-cycle and usage costs and increase developer and user frustration. Since many systems are designed to support organizations whose functions overlap, similar functionality is also redundantly and inconsistently defined and maintained. Structural and dynamic interfaces among these systems are frequently non-existent, or added on an ad hoc basis.

Information Engineering results in integrated systems, composed of reusable assets that are planned, architected, and designed to support enterprise-wide collaboration. IE stresses the value of creating enterprise-wide architectures, carefully partitioning shared information and functionality to create reusable "information assets," and capitalizing on combining these reusable information asset components to compose sets of interoperating systems.

The recommended scope for a Domain Analysis initiative is a group of similar systems representing both significant commonality and variability. Generally, these are functionally similar systems [1], but this is not strictly necessarily [3]. What is necessary is that they represent an opportunity for reuse that offers a clear business advantage. Ideally they should be under the control of a single development organization or a single sponsoring higher-level manager [3].

Domain Analysis originated in response to the needs of software development organizations with core competencies in one particular functional area (e.g. accounting). Their lines of business frequently consist of scores of similar systems, each "developed to spec" for a different client (or sometimes for the same client). This multiplicity of unplanned system variants results in escalating system life-cycle and usage costs and increasing stakeholder frustration. Therefore, Domain Analysis asserts that reuse is most likely to be successfully implemented and adopted by capitalizing on commonality, and understanding and controlling the variability, within sets of functionally similar systems (system families).

DA architectures may describe current system commonality and variability, and, in some cases, prescribe the range of variability to be supported in the asset base. To do this, DA models generally describe variants as "features" embodying the concerns of "stakeholders." These "stakeholders" may represent not only the user community but, for example, varying development standards, and development and operating environments. To fully understand the history of and reasoning behind system variants, several DA branches also stress the capture of system genealogies, noting the decisions, trade-offs, and rationale behind each feature.

To summarize these differences, IE focuses on reuse that supports collaboration among related functional areas (such as accounting, payroll, benefits, personnel), often ignoring the nastiness of modeling the variability among multiple systems supporting any one function (e.g. accounting). The result of IE's emphasis on commonality sometimes flies in the face of real-world requirements, and misses enormous opportunities for reuse. On the other hand, DA focuses on reuse that supports the commonality and required variability among multiple systems supporting any one function (e.g. accounting), often ignoring the nastiness of modeling the collaborations among related domains (such as accounting, payroll, benefits, personnel). The result of such isolated DA projects could very well replace "stovepipe systems" with "stovepipe domains."

2.4 DA and IE: The Courtship

Far from being incompatible, these differences between Domain Analysis and Information Engineering actually reflect orthogonal and highly complementary concerns. The reuse-based interoperability fostered by IE has an important role to play in many organizations adopting DA. The MIS area is not alone in requiring systems to communicate, share information, or support collaboration among their users. Following are several scenarios showing the need for an IE perspective in traditional DA arenas:

- Strategic and tactical DoD systems, manufacturing and process control systems, even CASE tools must frequently interoperate or support user collaboration to fulfill complex functions or missions.
- Several DA approaches recommend scoping small subsystem-level domains, acknowledging that systems will later be assembled from assets spanning several domains.

In the above scenarios, the need for the multi-domain interoperability architecture provided by IE cannot be overstated.

On the other hand, many enterprises adopting IE have considerable variation among functionally similar systems, the objects within them, or the environments in which they are developed and operate. The DA approach to control and management of this commonality and variability is potentially invaluable to such organizations. Following are several scenarios showing the need for a DA perspective in traditional IE arenas:

- Large enterprises often have decentralized operations, with significant distribution of, and consequent variability among, core functions. Frequently, they also have decentralized data processing responsibilities, with heterogeneous system and application hardware, software, development processes and standards, etc.
- Scores of similar MIS applications, and even similar enterprise models, are "developed to spec" by external companies for whom such work constitutes a line of business.
- Even less complex organizations may discover that many of the entities of interest to different functions are actually variants of the same object (e.g. Customers, employees, suppliers, contractors, affiliates, etc. are all, potentially, variants of "person"; in some cases, it may be important that they are occasionally the same person. Reuse of the more generic concept "person" offers both potential definitional and information value.)

In all of the above examples, the adoption of DA to manage the reuse of commonality and optimize the variability among the systems and environments would prove invaluable.

2.5 DA and IE: The Wedding

The good news is that organizations recognizing the need for both DA and IE do not have to do double work to adopt both disciplines. Because some of their objectives and methods overlap, and others are complementary, this interdisciplinary marriage can be accomplished relatively easily. The marriage will work on an organizational and management level because both disciplines examine similar information, pose many similar questions, and employ similar methods and processes. They both recommend the development of similar organizational frameworks and infrastructures, require similar planning, training and expertise, and require similar committed participation of many of the same stakeholders.

The marriage will also work on a technical level because both DA and IE modeling and architecture development activities employ similar structures and processes to partitioning information, albeit with different emphases. Both DA and IE systematically examine and partition information about the full life-cycles of multiple systems (or functions), decomposing and classifying, identifying commonality and necessary variability, and eliminating unnecessary redundancy.

Although other techniques are also usable, specialization analysis and decomposition analysis are two orthogonal techniques that, when combined, effectively support both the IE and DA partitioning process.

DA is primarily concerned with extending or tailoring reusable assets to form variants of similar systems. Therefore, the primary metaphor and partitioning mechanism of DA models is the generalization/specialization hierarchy. Nevertheless, decomposition is used, by some DA approaches, to model the context of operation for systems within a domain. It is also used to identify similar sub-structures within the domain (for feature comparison and component-based reuse), and to model the topology of the structural and dynamic relationships among them.

IE is primary concerned with combining reusable asset components to form interoperating systems. Therefore, the primary metaphor and partitioning mechanism of IE models is the decomposition/

aggregation hierarchy. Nevertheless, IE may occasionally employ specialization analysis to identify and partition functional and structural variants as required by stakeholders representing different user (i.e. functional) organizations and life-cycle stages.

IE and DA techniques are best combined by combining their scopes and primary partitioning mechanisms, starting with their respective planning stages. In this stage, a large problem space can be defined, consisting of multiple interoperating system families. This space can be partitioned into a high-level architecture of interrelated components. Each component known to have variants may be partitioned, in turn, into high-level specializations. Single components (or groups of several interrelated components) can be identified and prioritized as candidates for more detailed follow-on project stages. In these follow-on project stages, each partition (whether component or specialization) can be re-partitioned as required into lower-level components and specializations. IE techniques can be used to model the components, and DA techniques can be used to model the specializations. The high-level architecture can be used to coordinate the work of these projects, and to integrate their results into a growing asset base from which interoperating system variants will be composed.

2.6 DA and IE: The Church

Together with inheritance, specialization and aggregation hierarchies form the foundations of both semantic modeling and object-oriented (OO) modeling. Both DA and IE models and architectures can be fully represented using semantic networks and object-oriented modeling techniques. These scalable, versatile modeling techniques are useful for improving the precision of both DA and IE models and optimizing the reusability of their resulting asset bases. They are equally applicable to a wide spectrum of problem spaces, and maintain consistent structures and representations across the complete system life-cycle. They can be combined to represent specialization-based reusable structures in models and architectures, and to implement those same structures in code. For these reasons (as well as so many others that another paper could be written on this subject), these techniques are logical, if not ideal, candidates for DA and IE integrated modeling and asset building projects.

2.7 DA and IE: Happily Ever After?

The marriage of DA and IE seems to fulfill the quest for planned, managed, optimized reuse as neither can do alone. These two disciplines are not only orthogonal but complementary. They have common ancestry and similar backgrounds. Each seems supportive of the other's objectives. Furthermore, each seems designed to fill methodological gaps left by the other. Like any marriage, theirs will require work to truly achieve a union. However, if both parties commit to this union, they stand a good chance of marital success. Their children should turn out to be the high quality efficiently produced software we have been seeking.

3 Comparison

Several domain analysis approaches (notably KAPTUR [1] and GTE's DSSA [6]) develop rigorous IE-like topological architectures, but only within a single domain. Some domain analysis approaches stress the need for domain planning, but they do not require or even recommend rigorous modeling to scope and architect the boundaries of multiple interrelated domains [2, 3].

Conversely, as IE adopts a more object-oriented approach to modeling [7] it is starting to pay more attention to supporting functional variability. However, this rarely extends to support for environmental variability, let alone modeling of variation in systems engineering methods and system architectures themselves. Furthermore, IE does not yet include a formal approach to reusing its own models on subsequent similar projects, and does not describe how to build a library describing the taxonomic commonality and variability among such models.

In short, I am not acquainted with any literature describing truly comparable approaches, or any applications of such approaches. If anyone else has done any significant work in this area, I am most anxious to learn about it.

References

[1] S. Bailin, "Kaptur: Knowledge acquisition for preservation of tradeoffs and underlying rationale," tech. rep., CTA Inc., Rockville MD.

[2] "Stars reuse concepts volume 1 - conceptual framework for reuse processes (cfrp) version 2.0. stars-uc-05159/001/00," tech. rep., Electronic Systems Division, Air Force Systems Command, USAF, Hanscom AFB, MA., Nov. 1992.

[3] "Organizational domain modeling (odm) volume 1 - conceptual foundations, process and workproducts descriptions version 0.5. unisys stars informal technical report, stars-uc-15156/024/00," tech. rep., Electronic Systems Division, Air Force Systems Command, USAF, Hanscom AFB, MA., July 1993.

[4] J. Martin and J. Odell, *Development Coordination Handbook: James Martin & Company.* James Martin & Company (proprietary), 1991.

[5] J. Martin, *Information Engineering, A Trilogy.* Prentice Hall, 1990.

[6] "Domain specific softwaare architectures, command and control, domain model report, cdrl clin 0006," tech. rep., GTE, Chantilly VA., May 1993.

[7] J. Martin and J. Odell, *Object Oriented Analysis and Design.* Prentice Hall, 1992.

4 Biography

Robin Burdick has been a Staff Engineer on the Unisys Corporation STARS Reuse team in Reston Va. since December 1992. During her 16 year career, she has worked in all phases of the MIS lifecycle, primarily as a consultant and technology transfer agent. Her IE responsibilities have included project management, information strategic planning, business area analysis, business system design and development, information asset management, and methodology development. Other responsibilities have included testing and quality assurance, end-user support and training, information center coordination, software configuration management, and data administration. Throughout her varied career, she has specialized in discovering and improving methods to deliver efficiently developed maintainable flexible software that supports both anticipated and ad-hoc enterprise-wide collaboration. Ms. Burdick holds a BA cum laude from the University of Pennsylvania.