Enterprise Architecture Modelling Using Elastic Metaphors

Gerald R. Khoury
IBM Global Services
Level 17/821 Pacific Highway Chatswood NSW 2057 Australia
Gerald.R.Khoury@uts.edu.au

Simeon J. Simoff
Institute for Information and Communication Technologies
University of Technology Sydney
PO Box 123 Broadway NSW 2007 Australia
simeon@it.uts.edu.au

Abstract

Despite the hype surrounding enterprise architectures, they have delivered little on their promise. In this paper, we argue that enterprise architectures built using component-based frameworks are fundamentally flawed, in that they model the enterprise as a set of independent structures with discrete boundaries. Disparate concrete metaphors are used to describe each of these structures, with the result that enterprise architectures can only achieve partial success, at best, in providing a unified view of the enterprise.

This paper introduces the concept of 'elastic metaphors' as society-sourced metaphors for the conceptual modelling of information systems. By modelling the organisation using elastic metaphors sourced from naturally occurring enterprise structures, the enterprise architecture approach presented in this paper avoids the framework segmentation problem.

Keywords: Enterprise Architecture – Modelling - Metaphor

1 Introduction

As information technologies grow in complexity and scope, the need for a coherent approach to enterprise wide modelling becomes paramount. The expected benefits of an effective enterprise architecture are widely lauded; it is not the intention of this paper to argue these. Indeed, these benefits (agility, efficiency, improved opportunity analysis and so on) have acquired the flavour of motherhood statements and it seems at times that the obvious *desirability* of such benefits clouds the issue of whether these benefits are actually *delivered* by contemporary enterprise architecture approaches. This is not to suggest that enterprise architectures cannot deliver these benefits, merely to say that contemporary approaches to this problem have been largely hijacked by the consulting classes. Enterprise architectures have received surprisingly little focus from the academic community with the result that there has been scarce attention to the theoretical basis of enterprise architecture methods and frameworks.

Architecture is, by definition, borne of a metaphor based on classical architecture: the planning and construction of buildings. When Zachman established the notion of information systems architecture the analogy was very much a deliberate one, as Zachman consciously projected the levels of representation produced by classical architects onto the system development lifecycle. These representations give rise to a set of views representing the various perspectives taken by different participants in the system development process. Each of these representations are completely different, "different in content, in meaning, in motivation, in use, etc." (Zachman, 1987) However, this approach, and the ensuing developments in the conceptual modelling of enterprise architectures, created a range of issues described in the following section.

2 The Issues with Current Enterprise Modelling Approaches

The Zachman framework is probably the most recognised and popular approach to enterprise modelling. Zachman created seminal works in the area of enterprise architecture (Zachman, 1987) and (Zachman and Sowa, 1992). For this reason, this paper focuses on the Zachman framework for the purposes of comparative analysis; however, most of the observations would apply equally to many other enterprise architecture frameworks.

In essence, the Zachman framework provides a matrix that segments the enterprise into a variety of different views based on the different roles an actor can take. E.g. owner, designer, builder. Because each view is modelled using disparate techniques and methods (developed independently of the Zachman framework) each segment interface presents a discontinuity. This creates a barrier to the understanding of how structures flow from one part of the enterprise to another. Thus, by dividing the organisation into distinct views, the Zachman framework defeats its goal, which is to provide a *unified* model of the organisation.

Many new developments in the area of enterprise architecture can be viewed as attempts to complete the Zachman framework by developing techniques for specifying each of the (thirty) views precisely. There has been less focus on showing how the views inter-relate. Since this work is not yet complete, a *complete* enterprise model based on the Zachman framework is still beyond reach. In other words, after more than a decade of development, the predominant approach to enterprise architecture still does not provide a pragmatic solution to the problem of developing an enterprise wide model!

Like many enterprise architecture approaches, the Zachman framework "lacks scientific foundation" (Beznosov, 1998). While the framework provides "an observation of some natural rules for segmenting an enterprise into understandable parts" (Zachman and Sowa, 1992), there is little analysis of the laws and principles that govern these natural rules "in order not only to observe them but also to discover new rules and to be in a position to explain them."(Beznosov, 1998) As a result, the Zachman framework remains primarily a taxonomy with little efficacy for guiding the development of enterprise information systems.

What we need from an enterprise architecture is a method that is effective in:

1. Developing a single and coherent model of an enterprise, and
2. Allowing us to guide the future development of an enterprise without the creation of arbitrary internal boundaries.

3 Enterprise Systems as Models of Societal Structures

Computer systems can be viewed as digital models of real world analogues. These analogues present as either physical or organisational systems.

If the real world system is a physical one, then the use of a concrete metaphor to describe the system has the potential to provide an effective source-target mapping. (The approach presented in this paper parallels the use of metaphor in cognitive linguistics (Lakoff and Johnson, 1980), where the terms source and target[1] refer to the conceptual spaces connected by the metaphor.) Concrete metaphors are based on objects that users are familiar with from their everyday experience (L'Abbate and Hemmje, 1998). For example, if a computer system is used to replace a mechanical control system, it might make sense to model the computer interface along the lines of a mechanical interface. We might use concrete metaphors such as dials, gauges, buttons, sliders etc. In this way, we are able to conceptualize the non-physical in terms of the physical. (Lakoff and Johnson, 1980)

However, if the real world system is an organisational one, then the use of a concrete metaphor is likely to provide a poor source-target mapping which can lead us to invalid conclusions (Halasz and Moran, 1982). Indeed, some researchers have gone as far as to suggest that we should throw away the metaphor altogether, and "begin designing devices that have no metaphor, no real-world analogy." (Tristram, 2001) Yet, given that metaphor is so intrinsic to communicating and understanding new concepts (to the extent that perhaps *all* knowledge is based on metaphor) (Indurkhya, 1994) then how can we possibly hope to avoid using metaphor in a new field like information technology?

Our solution is to use, instead of a concrete metaphor, a metaphor sourced from a *conceptual* framework that is as flexible and extensible as the system we want to model. The best way to do this is to base the metaphor on the structures found within the system to be modelled. Organisational systems such as enterprises are created by societies: consequently, they reflect characteristics of societal structures. According to Gidden's Theory of Structuration (Giddens, 1984), there is an interdependency between humans (*actors* or *agents*) and societal structures (*resources* and *rules*) that is manifest through specific *actions*. Using this theory as a foundation, we can develop a wide variety of society-sourced metaphors. We term these 'elastic metaphors', because an important characteristic of these metaphors is that they are highly flexible.

Figure 1 shows a typical relationship between an elastic metaphor and a number of concrete metaphors used within the same domain with respect to "scope" and "level of description". "Scope refers to the number of concepts … that the metaphor addresses." (Hammond and Allison, 1987) (p.77) In the enterprise architecture context, scope refers to the number of separate systems to which the metaphor applies. Level of description refers to the granularity of the knowledge being conveyed. Note that an organisational system may include many physical systems and so the inclusion of concrete metaphors within an architecture framework may be warranted.

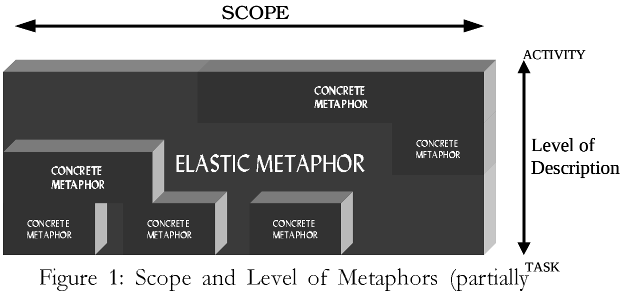

Figure 1: Scope and Level of Metaphors (partially based on (Hammond and Allison 1987))

If the metaphor is an *elastic* metaphor, it can be applied to a range of complex systems and the same metaphor can be used to model the system all the way from the highest conceptual layers all the way to the user interface. The elastic metaphor covers the entire enterprise domain whereas each concrete metaphor can map to only part of it.

At the highest level, we can use the concept of 'society' as the metaphor source. After all, "Information technology is arguably, like society itself, an abstract concept." (Marakas, Johnson et al., 2000).

[1] In the research literature the target is variously referred to as the primary system or the topic, and the source is often called the secondary system or the vehicle.

4 Structure of an Elastic Metaphor

While concrete metaphors have objects as their source, elastic metaphors have conceptual frameworks as their source. Arguably, the conceptual structure most familiar and fundamental to the human experience is the structure of society.

Gidden's Theory of Structuration (Giddens, 1984) shows us that society is shaped by the interaction between *actors* and the societal structures of *rules* and *resources* as manifest through the actors *actions*. An actor is an individual who can exert power in order to produce an effect. Resources are "structured properties of social systems, drawn upon and reproduced by knowledgeable actors in the course of interaction." Rules refer to the sanctioned modes of conduct, and an action is an activity that is performed.

Formalising this in terms of entities and relationships, we have the elastic metaphor represented by the structure shown in figure 2.

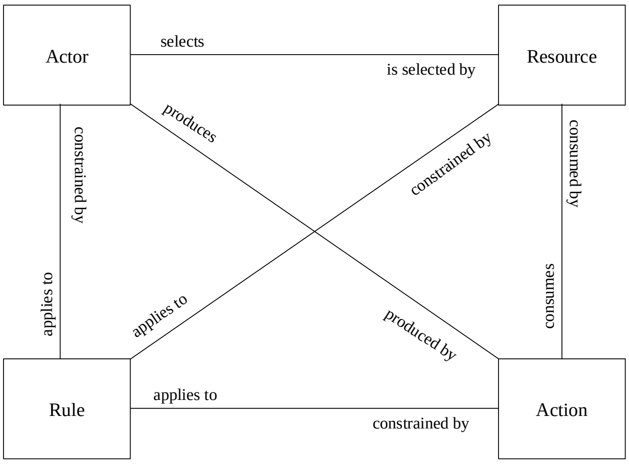

Figure 3 - Elastic Metaphor Entity-Relationship Diagram

Organisational systems can be viewed as a microcosm of society itself, and so the theory of structuration applies equally to any organisational structure. Thus, actors, actions, resources and rules serve as the cornerstones of the elastic metaphor. Because these entities are universal, we can use them to develop a myriad of metaphors that cross social, cultural and educational boundaries.

5 Applying Elastic Metaphors to the Enterprise Architecture Domain

In applying elastic metaphors to an organisational domain, we need to first identify the structures present in the organisation. These structures provide the source for the elastic metaphor.

Let us take as a simple example, a generic airline structure. In this example, as with most businesses, the organisational structure is hierarchical. Figure 3 shows an example of the high level structures of such an enterprise.

Rather than restructuring the components of this organisation to fit a generic framework, we use the naturally occurring organisational structure as an elastic metaphor for developing a unique framework.

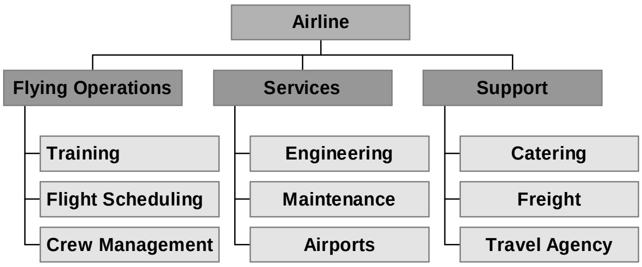

Figure 2 – Example of a Typical Airline Structure

Table 1 provides some examples of rule and resource structures that could apply to this organisation.

Business Structure	*Rules*	*Resources*
Airline	As defined by local and international trade and aviation regulatory authorities.	Aircraft. Airports. I/T.
Flying Op's	As defined by local and international aviation regulatory authorities.	Aircraft. Crew. Mtc facilities.
Training	License req's. Training facility availability.	Simulators and other training facilities. I/T.
Flt. Scheduling	Maintenance req's. Schedule req's.	Aircraft. I/T.
Crew Management.	As defined by Unions and Award Conditions	Employees. I/T.

Table 1 - Rule & Resource Structures for Airline Example

We need also to include actors in this model. Actors are also aligned to organisational units as shown in Table 2.

Business Structure	*Actors*
Airline	All employees. 3rd Parties.
Flying Op's	Crew. Trainers. Schedulers. Human resource managers. Unions rep's. Gov't rep's.
Training	Crew. Trainers.
Flt. Scheduling	Schedulers.
Crew Management.	Human resource managers. Unions rep's. Gov't rep's.

Table 2 - Actors for Airline Example

Finally, the *actions* performed by actors are both constrained, and enabled, by the organisational structures of rules and resources. However, just as any given rule, resource or actor can be present in more than one business structure, business structures are completely permeable to actors actions.

Let us then focus on the activity of Training as an example, and let us say a trainer is setting up a new course in response to a new regulation. The actions that trainer takes could include: examine regulation, develop course content, identify candidates, determine crew availability at base, determine simulator availability, schedule courses etc. The trainer is a member of the Training business unit, but these actions have required interaction with multiple business units.

Thus by building a metaphor of training (built using the structures of rules and resources and the actions taken on these by actors) we simulate the natural processes that take place within the organisation. We shift the focus of our analysis from predefined (organisation independent) technology structures towards structures based around the organisational (elastic) metaphor.

The elastic metaphor can also be viewed as a *class* of conceptual models (or metaphors). Within this class, we can form a number of different subclasses. For example, games, auctions, committees, etc. can each be viewed as elastic metaphor subclasses since they can all be modelled using characteristics of an elastic metaphor. Table 3 shows the mapping between the elastic metaphor class and a sample of the many subclasses that could created.

		ENTITIES			
Class	**Elastic Metaphor**	**Actors**	**Actions**	**Rules**	**Resources**
Subclass	Game Metaphor	*Players*	*Play to score points*	*Game Rules*	*Tokens*
	Auction Metaphor	*Buyers Sellers*	*Buy / Sell Decisions*	*Auction House Rules/ Legislation*	*Money / Communication Tools*
	Committee	*Members*	*Motion / Vote / Discuss*	*Constitution*	*Funding / Member Skills*

Table 3 - Elastic Metaphor Class-Subclass Mappings

Any of these (and other) subclasses could be used as elastic metaphors for enterprise modelling, since they all have the same basic society sourced characteristics. At the highest level it makes sense to model the enterprise on the 'enterprise' metaphor. But at lower levels other metaphors will be more natural. For instance, the game metaphor might be applied to systems that are token based and competitive. For instance, an aircrew rostering system is used by planners to reduce the cost of crewing aircraft operations. The most optimal solution for any particular month may never be known, but skilled planners can set the parameters in such a way that an optimal solution is more likely. In this case Dollars (cost) are the resources (game tokens) and the competition's goal is to score points by reducing that cost without breaking the rules.

It can be seen that the elastic metaphor modelling approach can be used to model any part of the enterprise from the largest structures down to the smallest (say, an information system interface). While this paper has focussed on the largest scale features of an enterprise, the other research shows potential for elastic metaphors to be effective in designing and redesigning human computer interfaces (Khoury and Simoff, 2003).

Elastic metaphor based models are intrinsically cohesive because the structures are tied together by concepts that run throughout every part of the model. Thus, no discontinuities are created by this model at the boundaries of framework or organisational structures.

In developing an enterprise architecture, it is the ability to portray the relationships between the different parts of the enterprise that is most essential. For the strategic planner, it is important to know what impact a change to one part of the organisation will have on another. A component based framework approach does not provide this information. It is essentially a deconstruction of an enterprise along arbitrary lines, but usually from an information systems perspective. Relationships between the various enterprise 'objects' can of course be built into the framework, but this is an afterthought that tends not to fit in naturally with the framework description, and in practice usually turns out to be extremely onerous to develop and near impossible to maintain.

Using the elastic enterprise architecture approach, the primary structures are connected throughout the entire model. Thus, the relationships between business entities are *implicitly* captured and if a business entity is changed, the relationships between it and other structures automatically accommodate the new form.

In this way, we avoid having to prescribe in detail how to model each part of a given organisational taxonomy, since we are provided instead with a set of rules that can be used as first principles upon which any part of an enterprise can be described.

The elastic metaphor approach can be used both for describing (developing) new enterprise architectures or redesigning existing ones. Khoury and Simoff illustrate the application of this idea applied to interface design. (Khoury and Simoff, 2003)

6 Assessing the Value of Elastic Metaphors

6.1 Source-Target Interaction

Using the concepts introduced by Anderson, Smyth et al (Anderson, Smyth et al., 1994) we can measure the effectiveness of the Source-Target mapping (Anderson, Smyth et al used the terms "Vehicle" and "Topic" equating to Source and Target).

The size of the intersection of the two sets representing the Source and Target features indicates the effectiveness of the particular metaphor in use (figure 4).

We previously defined four source features upon which an elastic metaphor is based: Actors, Actions, Resources and Rules. All of these features are used as part of the target system. Therefore, we can conclude that the target provides all the features supported by the source (T- S+ is zero).

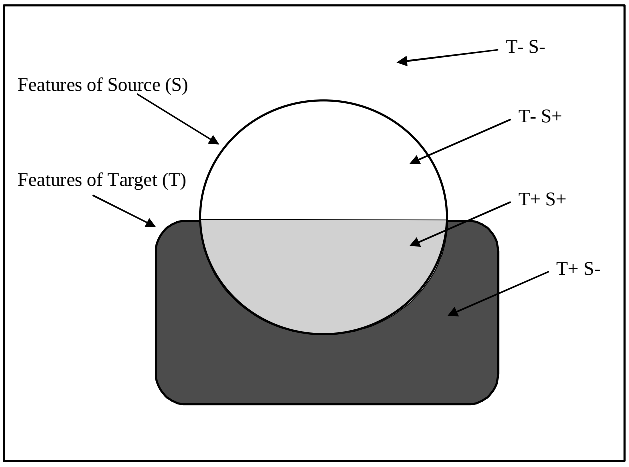

Figure 4 - Source-Target Interaction (based on Anderson, Smyth et al)

In addition, all features provided by the target systems are supported by the source, so (T+ S- is zero).

While the set S- T- is always infinite (by definition), it can be seen that an elastic metaphor will always provided a smaller T- S- than a concrete metaphor by definition: i.e. elastic metaphors are inherently expandable and always have potential to grow larger than the system currently represented. In other words, T- S- is minimised.

This leads to the conclusion that elastic metaphors can provide an *optimal* mapping from source to target domains.

7 Conclusion

In order to overcome the limitations of component based enterprise architecture approaches, a new approach is needed. The qualities of elastic metaphors make them ideally suited to enterprise modelling. The wide scope afforded by elastic metaphors means that system interoperability is improved. The depth of description means that the same metaphor is applied all the way from conceptual design to the user interface. And the increased commonality between user interfaces across functions and systems means that the useability of these systems is improved.

In today's world, the information systems *are* the enterprise. Application of elastic metaphors ensures that the information systems effectively reflect the structures and function of the enterprise they are modelling, and are not restrained by segmenting the enterprise along the lines of a generic, technology centric framework.

8 References

Anderson, B., M. Smyth, et al. (1994). Minimising Conceptual Baggage: Making Choices about Metaphor. People and Computers IX, Proceedings of HCI '94. G. Cockton, S. W. Draper and G. R. S. Weir. Glasgow, Cambridge University Press**:** 179-194.

Beznosov, K. (1998). Architecture of Information Enterprises: Problems and Perspectives. Written for the "Advanced Topics in Software Engineering" seminar given by Dr. Michael Evangelist during spring of 1998 at School of Computer Science, Florida International University.

Giddens, A. (1984). The Constitution of Society: Outline of the Theory of Structuration. Cambridge, Polity Press.

Halasz, F. and T. P. Moran (1982). Analogy considered harmful. Human Factors in Computer Systems Conference, Gaithersburg, Maryland, ACM.

Hammond, N. V. and L. J. Allison (1987). The Travel Metaphor as Design Principle and Training Aid for Navigating Around Complex Systems. Proceedings of the Third Conference of the British Computer Society, University of Exeter, Cambridge University Press.

Indurkhya, B. (1994). "The Thesis That All Knowledge Is Metaphorical and Meanings of Metaphor." Metaphor and Symbolic Activity **9**(1): 61-63.

Khoury, G. R. and S. J. Simoff (2003). Elastic Metaphors: Expanding the Philosophy of Interface Design. Proceedings of Computers and Philosophy Conference., The Australian National University, Canberra, Australia. In-print.

L'Abbate, M. and M. Hemmje (1998). "Virgillio - The metaphor definition tool." GMD Report **15**: pp.47.

Lakoff, G. and M. Johnson (1980). Metaphors we live by. Chicago, University of Chicago Press.

Marakas, G. M., R. D. Johnson, et al. (2000). "A theoretical model of differential social attributions toward computing technology: when the metaphor becomes the model." International Journal Human-Computer Studies **52**: 719-750.

Tristram, C. (2001). "The next computer interface." Technology Review **104**(10): 52-59.

Zachman, J. A. (1987). "A framework for information systems architecture." IBM Systems Journal **26**(3): 276-292.

Zachman, J. A. and J. Sowa (1992). "Extending and formalizing the framework for information systems architecture." IBM Systems Journal **31**(3): 590-616.