THE MITRE INFORMATION TECHNOLOGY ADVISOR

VOLUME 1 — ISSUE 11 — April 16, 2001

Getting Data to Applications Part 2: How We Can Do Better

ORGANIZATIONS INCREASINGLY NEED TO SHARE INFORMATION. IN THE FIRST PART OF THIS TWO PART SERIES, WE DISCUSSED SOME FALSE ASSUMPTIONS THAT HAVE CAUSED MANY PREVIOUS DATA INTEGRATION ATTEMPTS TO FAIL, AND WHY. IN THIS PART, WE PRESENT SOME SUGGESTIONS ON HOW WE CAN DO BETTER, BASED ON MORE REALISTIC ASSUMPTIONS ABOUT THE CONTEXT IN WHICH OUR SYSTEMS MUST BE CONSTRUCTED AND OPERATED.

The first step in how to do better is to adopt a more realistic objective. Instead of the perfect, universal system, where everyone can get everything from everybody, we suggest an adaptable, locally-improvable system that allows the enterprise to work with known partners, and that also has the flexibility to work with unknown future partners. Such a system will be based on high quality metadata that describes sources, services, and data requirements. This metadata needs to be active (i.e., used to run the system). Otherwise, from the start it can be characterized by error and soon become obsolete. Finally, aim to obtain short-term benefits from each investment. This allows us to get feedback if we go off course, and to keep stakeholders interested.

The second step is to partition the problem. Creation of a large system is a daunting task, requiring the building process to be partitioned, ideally so that each task is done by the most appropriate group. Specifically, the systems building process should identify the following:

- What each system needs to do individually. The main responsibility will be to provide self-description (e.g., what terms does this system use, and what services or data does it provide or use) in a somewhat structured form (i.e., using a set of defined fields). Where possible, these descriptions should use existing vocabularies.
- What each cross-system development project needs to do. These are projects that are targeted at creating specific inter-system connections. Connection-builders capture the same sorts of information that system owners might capture (if motivated), but only for the information required in the connection. We would like to see such knowledge explicitly captured, in a form that enhances the existing metadata and can later be reused in developing other connections.
- What domain coalitions must do. These groups must develop or select—and control evolution of—widely used vocabularies (e.g., common terms with agreed meanings) and make them available online. (This is something that eBusiness, health care (HL7), and other consortia are increasingly engaged in developing.)
- What technical management of the overall combination of systems must do. They must provide a technical architecture that identifies, among other things, various facets or aspects in terms of which interoperability can be defined and assessed. They must also identify what must be funded as technical infrastructure, e.g., tools, brokers, repositories, etc. Coordination efforts (e.g., to reach interoperability agreements) must also be funded and managed.

Breaking up the work in this way ensures that work is performed by those best able to perform it, and also identifies those costs which will have to be assumed by the overall system rather than by the constituent programs.

from page 1

A change of metaphor may also help. We often use the metaphor that systems should meet their data needs as if plugging into a "power grid" with a "wall plug." The idea being that each individual system and consumer should be designed to plug in to the overall system, without necessarily knowing who else will be connected. However, a wall plug is too simple an analogy when considering system interface requirements. A better analogy is the interfaces on the back of a typical computer. They have numerous pins, and require agreements about what flows through each pin. If conversion between one type of connector and another is required, what flows through each pin must be described, and a transformation devised for each flow. In other words, interfaces connecting systems must be defined in terms of multiple aspects, each one of which is important.

Hence, the metadata that must be provided should include information on multiple aspects of each resource, including:

- The semantics (meaning, e.g., in English) and representation of each data element and group of data elements (record or object).
- The signature (interface description) and semantics of each service (or object method) that can be invoked.
- The scope and completeness of the data or service provided (e.g., that the system provides information on all US fuel depots since 1970, or information on some NATO fuel depots since 1990). This is the sort of information that tends to be implicit, but needs to be made explicit when the system is made part of a larger system and hence accessible by those not familiar with its contents.
- Delivery style (information push vs. information pull, whole vs. changes)
- Quality of service, including such things as data quality, timeliness, attribution, completeness, obligation (of the service to continue to support the service), cost, etc.

In conclusion, grand vision of universal transparent data access is fine as a goal statement, but should not obscure the need to build systems that provide current value, can be enhanced incrementally, and give incentives to those who must implement them. In this article, we have described why attempts to satisfy "grand visions" generally fail, and suggested ways to keep from failing, while making realistic progress toward the future of systems that are never really "complete," but continually get better.

For more information, contact:

Arnon Rosenthal	*Arnie@mitre.org*
Frank Manola	*fmanola@mitre.org*
Len Seligman	*seligman@mitre.org*