Manufacturing Interoperability

S.R. Ray & A.T. Jones
National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

ABSTRACT: As manufacturing and commerce become ever more global in nature, companies are increasingly dependent upon the efficient and effective exchange of information with their partners, wherever they may be. Leading manufacturers rely upon computers to perform this information exchange, which must therefore be encoded for electronic transmission. Because no single company can dictate that all its partners use the same software, standards for how the information is represented become critical for cost-effective, error-free transmission of data. This paper discusses some interoperability issues related to current standards, and describes two projects underway at the National Institute of Standards and Technology in the areas of interoperability testing, and in self-integration research. We believe that tomorrow's standards will rely heavily upon the use of formal logic representations, and that these will enable automation of many integration tasks.

1 INTRODUCTION

The term "manufacturing interoperability" refers to the ability to share technical and business information seamlessly throughout an extended manufacturing enterprise (supply chain). This information, previously shared in a variety of ways including paper and telephone conversations, must now be passed electronically and error-free with suppliers and customers around the world. Disparate corporate and national cultures, a plethora of international standards, and numerous commercial products make this task all the more difficult, further underscoring the need for a clear and unambiguous interoperability infrastructure. The penalty paid by industry for the lack of such an infrastructure has been quantified in a 1999 study commissioned by NIST (NIST 1999). This study reported that the U.S. automotive sector alone expends one billion dollars per year to resolve interoperability problems. The study also reported that as much as 50% of this expenditure is attributed to dealing with data file exchange issues.

There are three principal approaches to reduce these exorbitant costs. The first is a point-to-point customized solution, which can be achieved by contracting the services of systems integrators. This approach is expensive in the long run because each pair of systems needs a dedicated solution. When there are, for example, ten partners in the chain, this would require up to 90 (10x9) interfaces. Moreover, should any system provider release a software upgrade, many of the translators would likely need modification.

A second approach, adopted in some large supply chains, is for each original equipment manufacture (OEM) to mandate that all supply chain partners conform to a particular, proprietary solution. This has been the practice, for example, in the automotive sector. While this may solve the interoperability problem for the OEM, it does not solve the interoperability problem for the partners. This happens because the first or sub-tier suppliers are forced to purchase and maintain multiple, redundant systems if they want to do business with several major OEMs.

The third approach involves neutral, open, published standards. By adopting open standards the combinatorial problems associated with interoperability becomes of order N rather than order N^2 as described above. When there are ten partners, only ten bi-directional translators are needed. Published standards also offer some stability in representation of information, an essential property for long-term data archiving. One example of a successful open standard is ISO 10303 (ISO 1994), informally known as STEP, the STandard for the Exchange of Product Model data. STEP, which is actually a family of standards, defines a neutral representation for product data over its entire life cycle. The most widely adopted component, ISO 10303-203 (ISO 1994b), (Configuration controlled design) is already conservatively estimated to be saving the transportation equipment manufacturing community over $150 million per year in mitigation and avoidance costs, with the figure expected to rise to $700 million by 2010 (NIST 2002).

But the problem is far from solved. Interoperability standards are used in layers, from the cables and connectors, to the networking standards, to the application or content standards such as STEP. All of these layers must function correctly for interoperability to be achieved. The greatest challenges remain at the top of this stack of standards.

2 IS XML THE ANSWER?

Many data interchange standards groups are adopting XML (the eXtensible Markup Language, available at http://www.w3.org/XML/) as the basis for specifying their data content standards. XML has

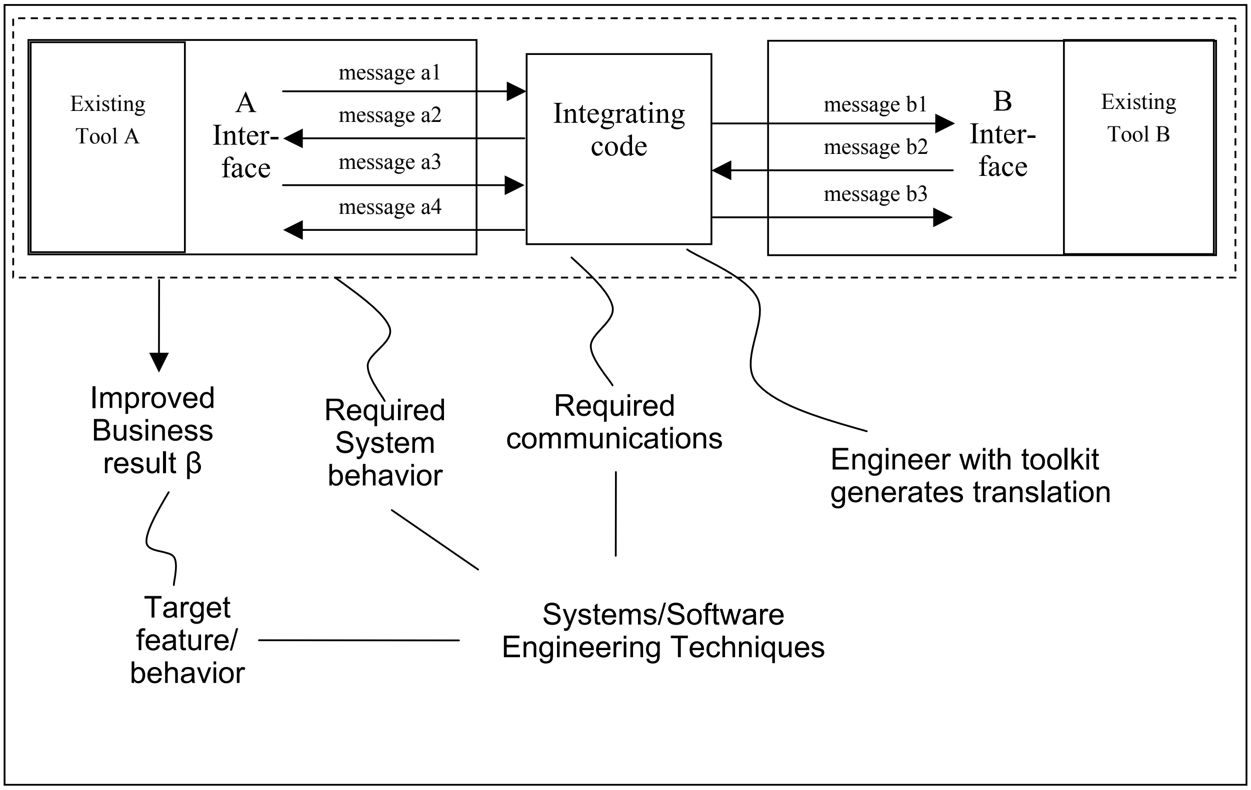

Figure 1: Typical Integration Scenario

had a tremendously positive impact on the connectivity of systems, but also has more clearly exposed what problems remain. XML is a markup language that can be used to tag collections of data with labels. As part of a standardization activity, communities can agree on the names for these labels. An interoperability problem remains, though, if different people have differing understandings of the meaning of an XML tag. Stated more succinctly, XML standardizes the syntax of data exchange, but was never designed to capture the semantics of the data. This is not necessarily an obstacle for a tightly knit community that operates within a common context. In this situation, the mental associations with a tag are shared and well understood by all. Where this limitation becomes a problem is in moving data from one context to another, for example sending data from a manufacturing context to a financial context. Without explicit, rigorous definitions of terms, misunderstanding is sure to arise.

Researchers at NIST, recognizing that we need a better way to capture definitions of terms, initiated the development of the Process Specification Language (PSL) (Gruninger & Menzel, in prep.). Upon searching for the best conceivable way to capture definitions of terms, the team adopted the answer suggested by the philosophy community (which, after all, has been pondering this question for a number of centuries). That community advocated the use of first-order logic, which brings the ability to reason over sets of definitions and to prove properties of these sets. One such property, for example, is proving consistency, which is tedious and prone to error using traditional information-modeling techniques. Using logic, it becomes straightforward to ensure the consistency of assertions for large sets of definitions by using automated theorem provers.

PSL, which is logic based, ensures rigorously defined and consistent definitions for data sets. This is a worthwhile goal in its own right, but even more exciting capabilities follow. Once a software system is equipped with a logic-based set of definitions (often called an ontology), then that system can advertise its outputs and desired inputs in a manner that can be manipulated and "understood" by other systems. This begins to move systems beyond the "screen scraping" techniques that are sometimes used today to collect data. Finally, such a rigorous foundation for data definitions can be the basis for reaching the holy grail of systems integration – self-integration.

In this paper we describe in more detail two projects that are part of our efforts to develop techniques that may lead to self-integration: the Automated Methods for Integrating Systems (AMIS) project, and the B2B (Business-to-business) Interoperability Testbed.

3 AUTOMATED METHODS FOR INTEGRATING SYSTEMS

A typical integration problem is stated as a requirement to produce some "improved business result"

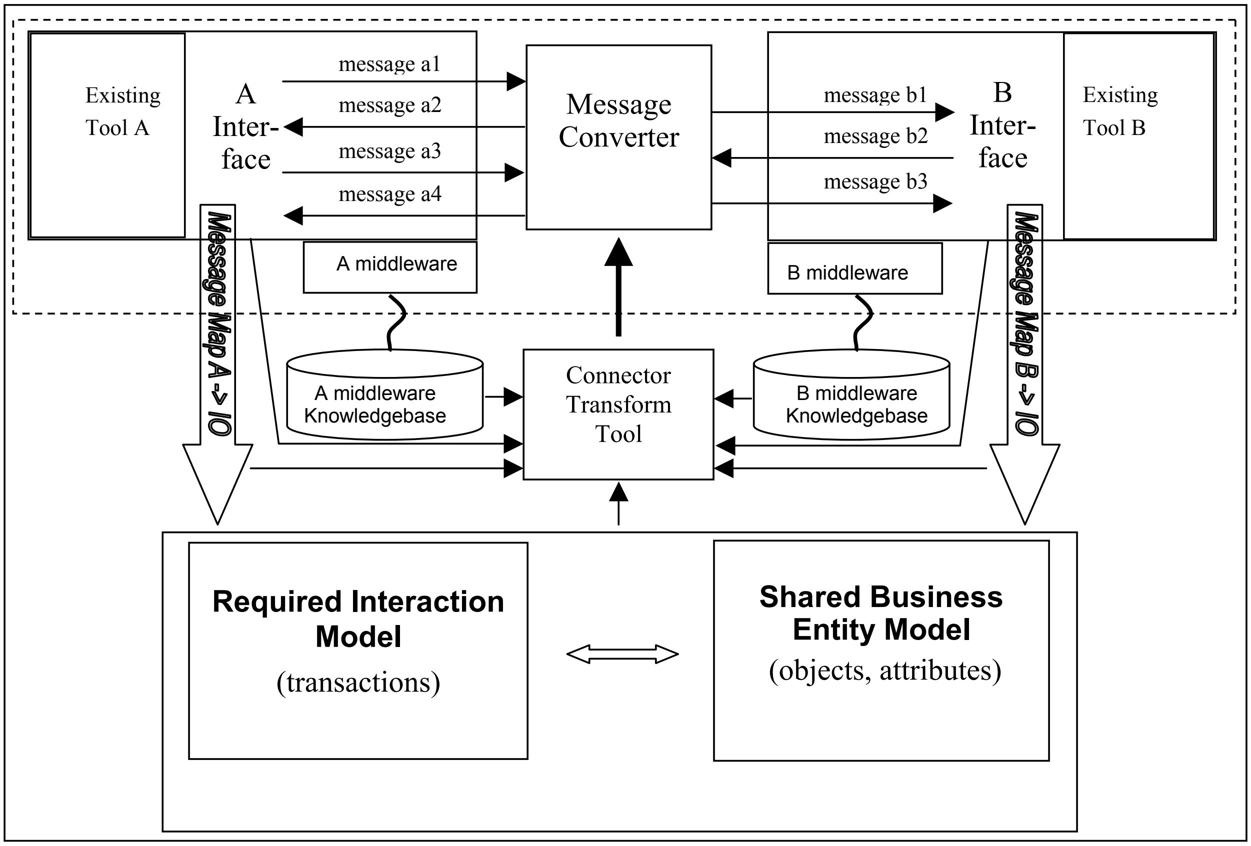

Figure 2: Visualization of the Vision

from a set of available business systems. In the case of supply chain integration, these business systems are usually software applications. Each of the available applications exposes interfaces by which it can interact with people and other applications. Integration is accomplished by building a new system that includes a set of known applications that are either augmented or modified to accomplish new functions jointly. The end result of this process is a connector between selected interfaces that leads to the stated "improved business result."

This process is largely manual, involving human-implemented and systems engineering techniques. These techniques reason from the required results and the behavior of the existing applications to the required behaviors of the modified system. From there, software engineers specify the translations between the two applications needed to produce that behavior and build a connector that uses the available interfaces to implement those translations (see Figure 1). This process is, in effect, no different than the point-to-point integration approach discussed above.

The Automated Methods for Integrating Systems (AMIS) project is addressing the following research question: can we automate (at least partially) the process of building that translation? In other words, can we build A-to-B translators that work directly from the interface specifications for A and B? This project specifically seeks to discover algorithms to automate (1) the translation of A's (standard) business transaction with B's (standard) form of the transaction, and (2) the modification of a system interface to a new requirement or technology. This does not mean integrating specific As and Bs (Asher & View, 1995; Barkmeyer et al., 2003).

There are three main areas of work in the AMIS project: interaction ontology formulation, semantic mapping, and connector transformation. Interaction ontology formation is concerned with capturing the "business" and "engineering" interaction concerns in a form suitable for reasoning. Semantic mapping pertains to building tools to create semantic maps between ontologies. Connector transformation is concerned with creating generators for dynamic message converters; this will ultimately expand to support dynamic protocol conversion. Efforts in each of these areas must come together to support automation in the integration process.

3.1 *Ontologies*

Systems engineers use a combination of top-down and bottom-up approaches to match business process objectives with component functionality. The AMIS approach is to formalize and capture the information the system engineer uses to perform this matching and then use software-based reasoning tools to support automation of the integration task. We envision two such ontologies: an interaction ontology and an implicit ontology. We also plan to use PSL, or something like it, to build these ontologies.

The top-down portion forms the basis for the interaction ontology, which is abstracted from the relevant concepts in the envisioned "business process." It contains the required interactions between the component systems with roles for each and a shared model of the business entities pertinent to those interactions (see Figure 2). For our approach, we need the interaction ontology to capture the "business" and "technical" interaction concerns in a form suitable for reasoning.

The implicit ontology contains the information each system uses to govern its dealings with the outside world it supports. It comprises a semantic or functional model and a set of explicit messages, interfaces, and data models. By "ontology" we mean a meta-model that guarantees consistent usage of its terminology. In our approach however, only those aspects of the implicit ontology relevant to the intended joint action must be made explicit.

3.2 *Mappings*

The next step of the process is to map the relevant notions in the implicit ontologies to their corresponding notions in the interaction ontology. The AMIS approach is to build tools to create the required semantic maps that define the detailed mappings for actions, messages, and information units. This is the difficult research problem – automating the creation, derivation, or extraction of semantic relationships between ontologies.

3.3 *Connectors*

Given a rigorous semantic mapping to a sufficient level of detail, it is theoretically possible to build a translation generator for the mappings. A number of enterprise application integration (EAI) tools do this for specific integration scenarios and underlying integrating infrastructures. To achieve arbitrary transformations of syntax, structure, and choreography down to the lowest levels of abstraction requires that all the information be formalized. Additionally, specific knowledge bases for the "middleware" technologies must be developed. Once generators for dynamic message converters are developed, support can be expanded to perform dynamic protocol conversion.

4 B2B INTEROPERABILITY TESTBED

Complementing the tools and methods emerging from the AMIS project, the objective of the industry-driven B2B Interoperability Testbed is an open, ongoing testing capability to provide on-demand interoperability demonstration and testing for four stakeholders: software vendors, manufacturing organizations, standards organizations, and government.

4.1 *Motivation*

With the surge in the development of B2B standards and associated technologies, the need for proof-of-concept demonstrations and interoperability testing has grown. Different standards organizations and industry consortia continue to invest in individual efforts to showcase the utility of these emerging standards and technologies. Manufacturing companies have readily supported these efforts to accelerate the adoption of B2B standards and technologies among their trading partners.

While there is support, the participants in these individual demonstrations have witnessed time and again the significant waste involved. This waste occurs because the software applications and infrastructure must be assembled from scratch each time a new demonstration is given. Procedures and rules of conduct must be reanalyzed, and organizational roadblocks must be overcome again and again for each demonstration.

The stakeholders realized that the same infrastructure could be used to perform both demonstrations and testing of different aspects of B2B interoperability – saving both time and cost. NIST and the Open Applications Group (OAG, http://www.openapplications.org.) have initiated the OAG/NIST Manufacturing B2B Interoperability Testbed (http://www.mel.nist.gov/msid/oagnisttestbed/) to provide this infrastructure. The goal is to have a distributed, living testbed that enables customer-driven A2A (application-to-application) interoperability testing over a wide range of B2B infrastructures.

4.2 *Approach*

There are three concurrent activities within the testbed: demonstrations and testing, developing the architecture, and implementing the infrastructure.

4.2.1 *Demonstrations and testing.*

The planning of specific demonstration and testing activities is driven by the identified needs of the stakeholders. Our two major focuses in the actual demonstration and testing activities are to show how (1) a web-based interoperability demonstration and testing infrastructure can satisfy the needs of customers and software vendors and (2) business message content testing can be performed to assure conformance to intended semantics. Presently, our work is focused on content testing.

4.2.2 *Architecture.*

The testbed is designed as a Web-based, distributed application. The participating nodes in the interoperability test would be of two logical types: test/monitor type and middleware/application type (see Figure 3). A testbed test/monitor node is a single logical node that may consist of multiple distributed functions running on multiple nodes. The middleware/application nodes are distributed among participating organizations such as vendors and users.

To enable interoperable behavior of these nodes, standards at different levels of the interoperability are used. To date, the testbed has focused on three layers of interoperability: messaging, business processes, and business content. The standards that are being used at the present time are XML for messaging, ebXML (http://www.ebXML.org) for business process specification and OAG for business content standards.

4.2.3 *Infrastructure.*

The testing and monitoring infrastructure used in the testbed consists of the following components:

1 Reflector and Transaction Store components support both disconnected and connected testing scenarios while allowing for the transactions to be routed to the specified end points, reflected to the originator, and stored in the permanent transaction log.

2 Business Process Monitor (or Choreography Checker) enables monitoring and conformance checking for choreographed transactions between business partners.

3 Content Checking tool enables constraint specification on syntax, structure, and semantics of the business message content.

The HTTP protocol is used to render the information exchanges in a Web browser.

4.3 *Major Deliverables.*

Figure 4 shows a timeline of recent demonstrations and activities. The testbed delivers updates of the infrastructure tools, demonstrations at conferences and meetings, and technical presentations. More details on this work can be found at www.mel.nist.gov/msid/oagnisttestbed/.

5 SUMMARY

In this paper, we discussed the emerging criticality of interoperability in the arena of Internet-based manufacturing. We argued that the traditional approach of painstakingly defining content standards for each application cannot keep pace with industrial need in the long term. We proposed a new semantic-based approach and a vision, called self- integration, that promise to dramatically reduce the costs and difficulties involved in achieving interoperability. We then briefly described two projects at the National Institute of Standards and Technology that will provide the foundation for realizing that vision.

PRODUCT DISCLAIMER

Certain commercial software products are identified

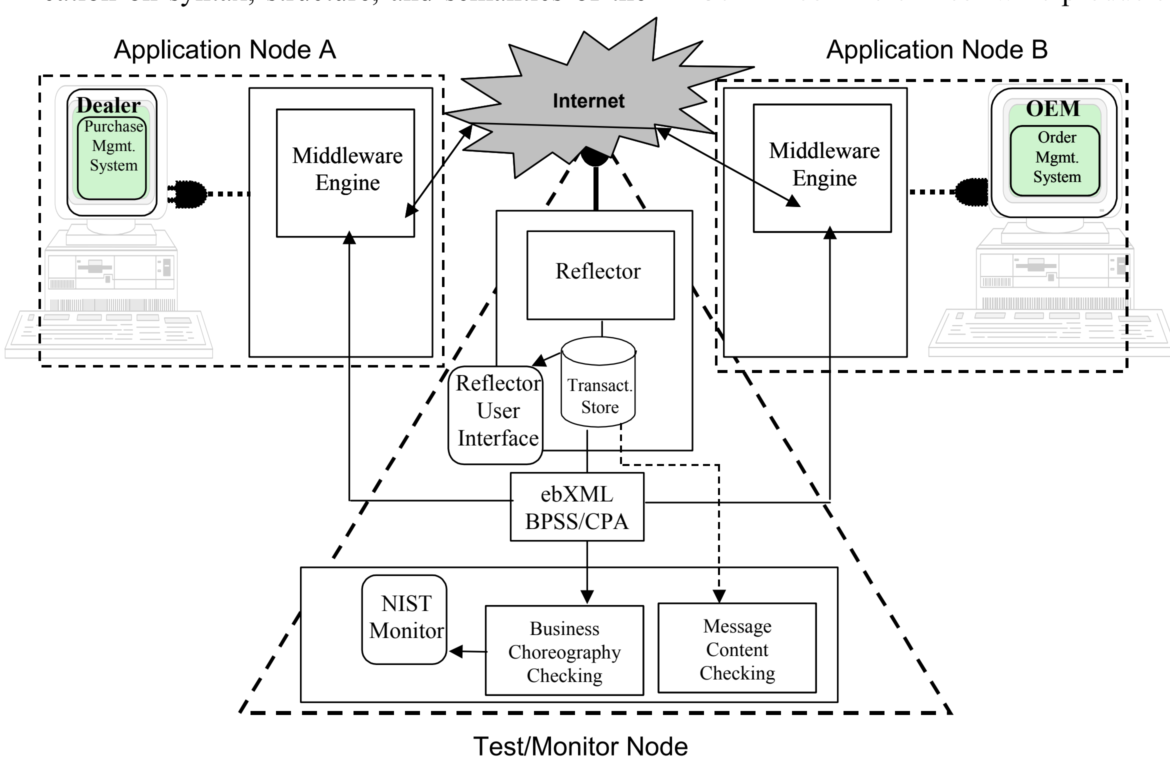

Figure 3. Testbed Architecture

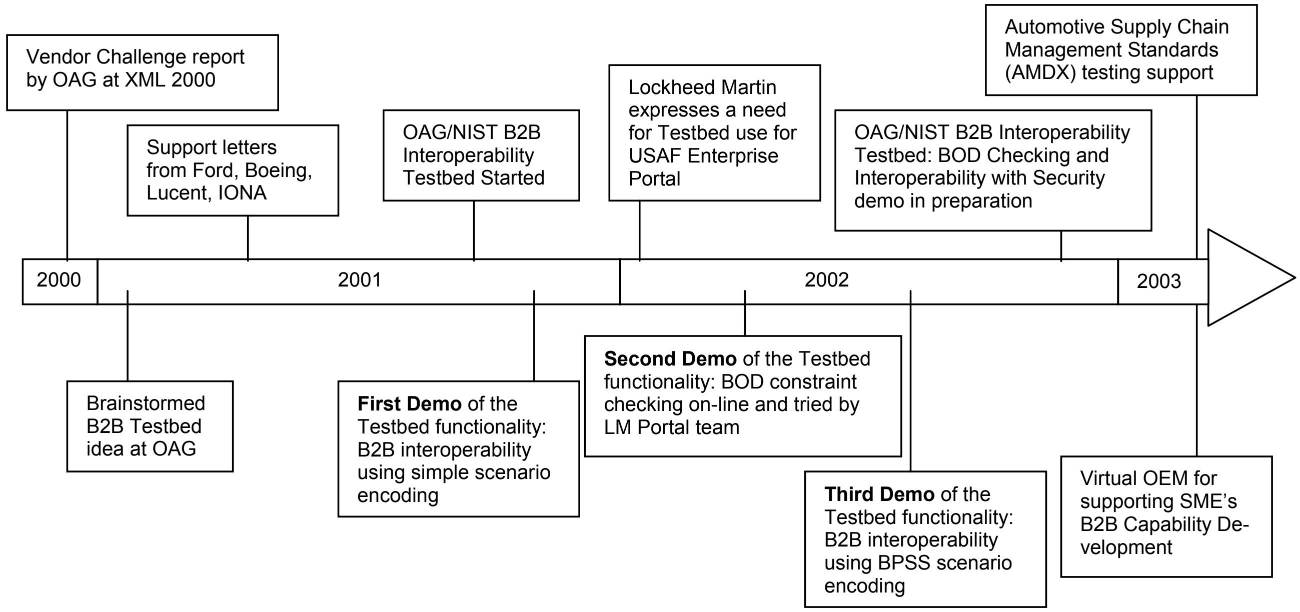

Figure 4. Timeline of Recent Activities of the B2B Testbed

in this paper. This use does not imply approval or endorsement by NIST, nor does it imply that these products are necessarily the best available for the purpose.

REFERENCES

Asher, N. & Vieu, L. 1995. Toward a geometry of common sense: a semantics and a complete axiomatization of mereotopology, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 846-852, 1995. Montreal, Canada.

Barkmeyer, E., Feeney, A., Denno, P., Flater, D., Libes, D., Steves, M., & Wallace, E. 2003. Concepts for automating systems integration, NISTIR 6928, National Institute of Standards and Technology, Gaithersburg, MD, 20899.

Gruninger, M. & Menzel, C. in prep. Process Specification Language: Principles and Applications, to appear in AI Magazine, and http://www.nist.gov/psl.

ISO 1994. ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview, International Organization for Standardization, Geneva, Switzerland.

ISO 1994b. ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 203: Application Protocol: Configuration controlled design.

NIST 1999. Interoperability Cost Analysis of the U.S. Automotive Supply Chain, (Planning Report #99-1), 1999, available at http://www.nist.gov/director/prog-ofc/report99-1.pdf.

NIST 2002. Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries, (Planning Report #02-5), 2002, available at http://www.nist.gov/director/prog-ofc/report02-5.pdf.