Fit for Purpose: Toward an Engineering Basis for Data Exchange Standards

Arnon Rosenthal, Len Seligman, M. Allen, Adriane Chapman

Fit for Purpose: Toward an Engineering Basis for Data Exchange Standards

Arnon Rosenthal[*], Len Seligman[†], M. David Allen[†], Adriane Chapman[†]

[*]The MITRE Corporation, Bedford, Massachusetts, USA
[†]The MITRE Corporation, McLean, Virginia, USA
{arnie, seligman, dmallen, achapman}@mitre.org

Abstract. Data standards are a powerful, real-world tool for enterprise interoperability, yet there exists no rigorous methodology for selecting among alternative standards approaches. This paper is a first step toward creating a detailed *engineering* basis for choosing among standards approaches. We define a specific sub-problem within a community's data sharing challenge, and focus on it in depth. We describe the major choices (*kinds* of standards) applied to that task, examining tradeoffs. We present characteristics of a data sharing community that one should consider in selecting a standards approach—such as relative power, motivation level, and technical sophistication of different participants—and illustrate with real-world examples. We then show that one can state *simple* decision rules (based on engineering experience) that system engineers without decades of data experience can apply. We also comment on the methodology used, extracting lessons (e.g., "negative rules are simpler") that can be used in similar analyses on other issues.

Keywords: Science basis for enterprise interoperability; experience reports on interoperability solutions; reference ontologies and mapping mechanisms; model-to-model transformations

1 Introduction

Many of the biggest practical successes in data integration have been due to effective use of data standards [9]. In one survey, a substantial majority of participants in data standards development indicated that high quality data standards lead to improved interoperability [4]. The reason for this is straightforward: the right standard for a given community reduces the amount of data heterogeneity that must be bridged and therefore saves time and effort when new data exchanges must be built. Unfortunately, there are also many cases of failures, where standards failed to provide positive return on investment [9].

Despite the importance of good data standards, the research community to date has offered practitioners little help in selecting a standards strategy that will work for their particular data sharing community. This is perhaps not surprising, since researchers naturally focus on well-bounded problems suited to deep computational analysis (e.g.,

schema matching [8, 2, 12], mapping generation [3, 7]). This research has spawned some powerful and useful commercial and open source integration tools. *Yet researchers are mostly mute on the important strategic questions facing data integration professionals that must pick a standardization approach appropriate to their data sharing community*. *Addressing these strategic questions is a necessary precondition to establishing a scientific and engineering basis for enterprise interoperability.*

This paper identifies a key strategic question that must be addressed by a science or engineering process of enterprise interoperability—how to select a data exchange standardization approach that is fit for its intended purpose—and makes progress addressing it. Our contributions are:

- We describe characteristics of a data sharing community that one should consider in selecting a data standards approach and illustrate with real-world examples;
- We delineate the broad categories of data standards and present the relative strengths and weaknesses of each;
- We present an initial set of decision rules that help a data integration expert pick an appropriate type of standard for a particular data sharing community; and
- We describe important topics for future research on strategic help for data integration professionals.

Instead of presenting rigorous results on a narrow question, this paper presents a novel, strategic perspective on a familiar problem (data integration) on which novel perspectives are rare. The state of the practice is "Do a tradeoff analysis and apply engineering judgment, based on your knowledge of the various technologies". We hope that this paper will inspire other researchers to address problems of strategic importance that can truly be the basis of an engineering discipline of enterprise interoperability.

1.1 Scope

For this paper, we assume that a community is interested in creating many exchanges, covering similar data, such as might be described by a message standard. For example, among a region's medical offices, hospitals, labs, pharmacies, insurance companies, and relevant government agencies, there are many reasons different sets of partners would want to exchange basic medical information about patients. The goal of a data standard in such a community is to minimize the amount of development effort required to build and maintain all the required data exchanges. This paper examines several ways that data standards are expressed and used, discusses the pragmatic tradeoffs, and provides some easy to use decision rules to guide a selection for a particular data exchange problem.

Terminology in this area is diverse and conflicting. In this paper, an *exchange* concerns all work necessary to take data between data *producer* and *consumer* organizations. A *(full) transform* is the executable function that maps data from a producer's export interface to a consumer import interface. To permit direct comparisons of al-

ternate approaches on a well-defined task, we seek specifically to minimize the effort to develop and maintain the transforms used by a set of exchanges.

A *message-based* transform is composed of two *sub-transforms*, from the producer's native interface to create a structure conforming to the *message standard*, and one onward to the consumer interface. The message standard is often described as an XML schema, supplemented by requirements captured in English or as code sets. Data typically flows point to point, being transformed en route, usually without a central server. The end points of any transform are called its *source* and *target.*

To write a transform, one needs to understand what differences between source and target need to be spanned. These differences may involve *meaning* (what is a patient, a `testDate`, a `homeState`, or a `secondaryDiagnosis`), *value domains* (what is meant by `Date`, `State`, or `Diagnosis`; what values are allowed for each?), *value format* (the coded value "54131-8" refers to `Gender` under LOINC; dates have many representations, such as "dd-mm-yyyy" or "yyyymmdd"), and *structure* (e.g., HL7 C32 message, as a tree). To make descriptions more concrete, we will assume system interfaces and the message standards use XML; the work still applies when alternatives to XML are used. A typical description is part *formal* (i.e., interpretable by tools; XML schemas are formal) and partly English. An alternative to full description is to identify *correspondences* – areas where the source conforms to the target interface and the transform need do nothing. Formal descriptions reduce ambiguity, and may drive automated tools, such as when multiple partners' data schemata are mapped to a common reference ontology.

1.2 Paper Roadmap

Section 2 describes characteristics of a data sharing community that affect the choice of an appropriate data standards strategy. Section 3 describes the major styles of data standards, along with examples. Following a discussion of pragmatics (Section 4), Section 5 presents decision rules for selecting among the major standardization strategies described. These rules consider factors described in all the previous sections. Section 6 concludes with topics for future research.

2 Characteristics of Data Sharing Communities

This section describes key characteristics of a data sharing community that affect the choice of an appropriate data standards strategy, illustrated with real-world examples. (The list of characteristics is not comprehensive, but emphasizes factors that have generally received less attention in prior research.[1])

First, *is this a community of peers with roughly equal power or is there is a dominant player* (a so-called "800 pound gorilla")? The Indiana Health Information Ex-

[1] [13] presents additional factors and offers a useful conceptual framework for standards communities, while [6] offers additional insights on the issues of motivation and technical sophistication of participants.

change[2] is an example of a community of peers, in which hospitals, rehabilitation centers, long term care facilities, laboratories, imaging centers, clinics, community health centers and other healthcare organizations exchange health information. In contrast, the Centers for Medicare and Medicaid Services (CMS) have a dominant position in the U.S. healthcare marketplace, covering almost one third of the US population[3]; providers that wish to receive payment from this major player have strong incentives to support CMS' data submission requirements.

Second, even when there is a dominant player, *communities differ in the extent to which the dominant organization uses its power dictatorially or benevolently*. Using the former approach, financial regulators could simply threaten penalties or legal action against non-compliant companies. Benevolent financial regulators might also offer tools that ease compliance as well as benefits to data submitters (e.g., by selecting a standard and providing tools to minimize submitters' costs, or by allowing participating financial institutions to see industry-wide data and compare it to their own practices).

A third issue is the *motivation level of participants*. Information consumers are usually highly motivated, since they are the ones that derive value by using the exchanged data. Whether data producers are similarly motivated depends upon their perception of benefits vs. costs and risks. Producers can be motivated by legal mandates (e.g., taxpayers that fail to provide the government required information can face penalties or even jail), financial incentives (e.g., the U.S. government is offering payments to healthcare providers that demonstrate "meaningful use" of electronic health records technology), or a perception of benefit (e.g., U.S. airlines and other stakeholders voluntarily provide safety information to the Aviation Safety Information Analysis and Sharing[4] collaboration because of a shared desire to improve aviation safety). As another example, drug companies are highly motivated to comply with Food and Drug Administration data submission requirements, since they need FDA approval to be able to sell a new drug in the U.S. market. In contrast, for an application that analyzes a company's competitors' web-accessible price lists, the data producers (other companies) have no motivation to help.

Fourth, participants often vary greatly in terms of both *technical sophistication and financial resources*. In sharing emergency response information, U.S. state and local authorities generally have far fewer resources and less technical sophistication than U.S. federal government agencies. In contrast, in the financial regulatory environment, data producers (i.e., financial institutions) have ample resources and sophisticated IT capabilities. Healthcare information exchange varies greatly; some providers (large health maintenance organizations or hospital systems) are amply resourced, while others (individual doctors' offices) have little or no IT budget or capabilities. In such cases, it is often valuable for a government entity (such as the U.S. Office of the National Coordinator for Health IT), an industry consortium, or an open source com-

[2] http://www.ihie.com/
[3] http://cms.gov/
[4] http://www.asias.aero/

munity to offer tools and/or services that lower the barriers to data sharing for under-resourced players.

These issues are all important in considering the most appropriate data standards strategy. Much prior research fails to realistically consider participants' incentives [10]; this has also led to some catastrophic failures in real-world data integration efforts [9].

3 Standards Options

This section presents several flavors of data standards and usage. For each, we ask:

- Are standard messages generated, as an intermediate result? If so, how tightly does the standard constrain them? How much of that standard is expressed formally?
- Do the formal descriptions of data producer, consumer, and (if used) message standard suffice for developing transform code? Are all these interfaces described using the same formalism? Is that formalism standard or proprietary?
- Who needs to do what work to create a transformation? How many topics must a community resolve in order to create their descriptive standard?

Our biggest contribution lies in the nature of the analysis – a well-defined task within data exchange, and description of the alternative models. We expect the set of decision rules to be extended and refined by later researchers. While there are many discussions of strengths and weaknesses, we know of no systematic comparison and usable set of decision rules. [14] begins studying the area scientifically; we seek to create an engineering solution, usable by MITRE's system engineers (who are not all data experts).

Our assessments of the strengths and weaknesses of each standardization approach are based on over 50+ years' direct experience among the co-authors, consulting on U.S. government data integration efforts, plus many discussions with other data integration experts at MITRE.

3.1 "Nailed Down" Exchange Schemas that Specify a Single Physical Data Structure

Approach 1: Create a message-based transform, using a detailed, straightforward message standard that specifies a fixed meaning and format for each element. An XML schema often describes the structure, simple constraints on content, plus part of the syntax. Remaining message-standard details are described in English, in a proprietary tool, or as standard code sets (e.g., "`countryOfOrigin` uses FIPS Country Codes"). Transform developers must be aware of these details to ensure that the delivered content meets the needs of the consumer. A natural division of labor is that each producer/consumer creates *one* sub-transform between their own interface and the standard; these are composed automatically to create producer-to-consumer transforms. For example, if a community agrees to exchange patient information via a

greenCDA[5] XML message, each participating data producer and consumer will have to develop the sub-transform from its electronic health record format to the standard.[6] Documentation for producer and consumer interfaces is typically left to each organization.

Most exchanges today take this approach. For example, the National Information Exchange Model (NIEM)[7] describes a process that employs reusable schema components to build message standards, which NIEM calls information exchange package descriptions (IEPD). Many IEPDs which are built fall into the category of "nailed down" exchange schemas. The web's Really Simple Syndication (RSS) is another important example.

Discussion: The popularity of approach 1 is based on real advantages. Only one sub-transform need be built for each producer or consumer. When development is manual, there are usually large savings over creating a separate data transform to each partner. It is often natural for each producer or consumer to build and host its own sub-transforms. Otherwise (e.g., for secondary uses), whoever benefits from the exchange can build and/or host the transforms.

There are still difficulties. Developing one sub-transform per interface is still a significant cost, and a real barrier to entry for technology-challenged organizations. Furthermore, if a community extends the standard, they face costs to create transforms that exploit the extensions. Depending on the design of the transform code, a later (extended) version of a standard may not be plug-compatible with an older version. Precision (numeric or in concept definitions) may be lost in conversion to the standard, even though the consumer could have accommodated it.

The community cost to develop a message standard can be high, depending on the complexity of the message standard and interface and the number of stakeholders with distinct points of view. Not only do they need to agree on data content, but they must also agree on a tree structure and value formats – what concepts are on the top of hierarchical structures and whose native representation to use.

3.2 Flexible Exchange Schemas that Permit Alternate Representations

Approach 2: Participants agree on a message structure but allow alternative formats for some of the individual properties and occasionally for overall message syntax. For example, they might allow different measurement units (feet vs. meters), value syntax and international codesets (National Drug Code or RxNorm to describe medications), and allow both XML and JSON to represent the tree.

With this approach, a producer chooses the most convenient supported format for each item in the standard message, and then creates a sub-transform to it. Just as in the previous section, consumers create transforms from the message, to the consum-

[5] http://wiki.hl7.org/index.php?title=GreenCDA_Project

[6] In the event that the producer or consumer organization is using a vendor's electronic health record system, it be able to acquire the data transform code as a part of the software package or as an add-on.

[7] http://niem.gov

er's format, converting as necessary. The formalism is often XML schema, with certain elements being assigned a special meaning as descriptors of other elements.

The C32 and Clinical Document Architecture (CDA) standards are examples in this area. CDA permits the use of a number of different coding systems. For example, in a Continuity of Care (CCD) document, different codes might indicate the role of a physician, or the race of a patient. But the content of the code itself is not fixed; the standard may permit several coding systems, such as SNOMED or ICD-9 to describe medical conditions. As a different example, consider "message envelope" designs that contain a standard structure for the manifest of the contents, but leave the contents open to customization depending on the user; an example for health information is Restful Health Exchange (RHEx).[8]

Discussion: This approach is friendlier to data producers. By giving them alternate submission formats, it lowers the barrier for entry. This is especially important when producers have limited resources or a low motivation to contribute data. An additional advantage is that this approach can avoid gratuitous loss of data precision caused by a "least common denominator" interchange standard, in cases where the source and target may both use the same higher precision representation. No exchange standard can remedy a situation where the consumer requires information, or precision, that the producer does not possess.

The major drawback of permitting alternate representations within the standard is that to be interoperable with all producing systems, each consumer's sub-transform becomes much more complex; it must interpret descriptors of the representation choice selected and then invoke the correct format translation. Also, while XML schema can specify the message structure (Approach 1), no popular standard captures the special usage for format descriptor elements.

To see the advantage of flexible representations, a medical data exchange standard could allow producers to furnish `prescribedMedication` using either National Drug Code or RxNorm, given that straightforward conversions exist from NDC to RxNorm. The advantages are less clear cut when such conversions cannot be automated. For example, while ICD-9, ICD-10, and SNOMED could all be used for `diagnosis`, there is often no straightforward mapping among them. If a standard allowed all of those representations, then the consumer will only see diagnoses for which there is an unambiguous mapping to his chosen coding scheme.

A final consideration with this approach is the cost of obtaining the consensus necessary to develop the standard. While initially this cost may be less than approach 1, since supporting several popular formats can avoid some arguments, subsequent costs may be similar, as participants argue about whether to accept the next one alternate representation, and the next.

[8] http://wiki.siframework.org/RHEx

3.3 "Enriched" Exchange Schema (Schema plus Formal Descriptions)

Approach 3: *This approach enhances approach 1, adding formal descriptions plus automated tools that compare descriptions and in some cases automatically generate sub-transforms to and from a standard message.*

Here, the community must choose a formalism for capturing semantic knowledge that XML Schema does not capture (e.g., that `hemogram` is a kind of `labTest` or `anesthesiologist` is a kind of `physician`); popular formalisms for capturing such a domain model include UML, the Resource Description Framework (RDF)[9] and Web Ontology Language (OWL).[10] Then, instead of just presenting the community with an exchange schema (as in approach 1), the elements of the exchange schema are described by correspondences to elements in the domain model.[11] (One could, more awkwardly and less expressively, use the XML exchange schema itself as a domain model). Whenever producers or consumers describe their systems in terms of the same domain model, data exchange is eased. First, there is the possibility of doing automated mediation using sophisticated tools that automatically generate some of the sub-transforms. Crucially, the descriptive tasks can be done by a domain analyst, or in less critical settings, a power user. When automated transform generation succeeds, there is much less need for programming. Costs are reduced, and more systems can be included in exchanges. In addition, even when fully automated mediation is not possible, the additional semantic richness of the descriptions can make the programmer's job substantially easier.

Discussion: This approach has the potential to reduce the cost and time required to create sub-transforms. Compared with approach 1, it adds formal capture of the knowledge that the developer needed for generating mappings. Major vendors' tools (e.g., IBM, Microsoft) have proprietary logic-based formalisms for expressing and exploiting correspondences to generate transforms. This represents a big advance over the current state of the practice of capturing correspondences in Excel or (even worse) Powerpoint followed by manual programming (e.g., in Java) of sub-transforms.[12] Another advantage of this approach is support for incremental adoption. Where automated transform generation is impossible or is incomplete, one can always fall back on the techniques of approach 1 to fill in the gaps.

A drawback of this approach is that while individual correspondence capture acts are simpler, the setup is not. The community (or some members) must invest in sophisticated tools, *and* the skills to apply them, when transforms cannot be generated automatically. This may be a challenge for communities in which key participants are poor in either resources or technical sophistication. In compensation, if the communi-

[9] http://www.w3.org/RDF/

[10] http://www.w3.org/OWL/

[11] There are two flavors of correspondence: a simple "is compatible with", and "is derivable by <formula>". From these fine grained correspondences, a tool can derive a transform between whole schemas. Correspondences can also include small sets of elements, e.g., M, D, Y → Date.

[12] One shortcoming of current tools is they have difficulty making very large schemas and the correspondences among them intelligible [1].

ty can come up with a strategy to address these challenges (e.g., by providing free tools or putting the mediation burden primarily on better resourced partners), the approach has the potential to substantially reduce the effort required to build new data exchanges.

We considered a hybrid this approach with approach 2 (i.e., using a "semantically enriched" schema but with alternate acceptable representations), but none of the current standards provides for elements (i.e., meta-attributes) that describe the format of other elements.

3.4 Formal Descriptions Without an Intermediate Message

Approach 4 rests on having a domain model, and formally describing all producer and consumer interfaces in terms of this model, all in the same formalism. Using these descriptions, one generates transforms for all desired exchanges, going directly from producer to consumer interface. The domain model acts as intermediary for descriptions, but not for physically creating messages.

Automation is essential because for each producer, one must generate direct mappings to all of its exchange partners, not just to a standard. Formal descriptions are essential to automation. The community must again acquire tools and skills, but will need fewer programmers. Then, the community creates its standard domain model, and some entities within it acquire the tools and skill to use them.

There are major risks with this approach. If the community is not new, organizations who have already built their sub-transforms will have little short term reason to create the needed formal descriptions. The tools are bleeding edge and not industrial strength; the techniques are unfamiliar to existing developers. And the larger number of transforms makes manual coding a less effective backup plan. Thus, its niche may be limited to new efforts, plus situations where translation to a fixed message format is not practical.

The Biomedical Informatics Research Network (BIRN) has begun to use this approach for real information exchanges [5]. Interestingly, BIRN mediates not just data transformation, but also queries. In addition, many models exist to describe portions of the health domain, including the Federal Health Information Models (FHIMS[13]), and the NIEM UML Profile[14] and the Health Level 7 Reference Information Model (HL7 RIM)[15], which is used to derive other standards like the Clinical Document Architecture (CDA). While these models by themselves do not constitute a "formal description" approach, they do provide useful building blocks for such an approach.

Discussion: This approach potentially makes interoperability, and especially extensibility, an order of magnitude cheaper, compared to standards plus hand-coded transforms (approaches 1 and 2). Also, since the tools convert values directly between producer and consumer representations without an intermediary, the standard does not cause unnecessary loss of precision. Extension is particularly easy – GUIs and wiz-

[13] http://www.fhims.org/
[14] https://www.niem.gov/news/Pages/uml-profile.aspx
[15] http://www.hl7.org/implement/standards/rim.cfm

ards relate the new standard elements to elements in source and target, and a mediator generates updated transforms. No programmer and test organization will be involved, at least once the tools mature.

Unfortunately, present transform-generation tools are unsatisfactory. Without reliable automation, it is not feasible to generate a separate transform for each exchange. For long term planners such as CIOs and architects, it is worth pointing out that without formal descriptions of systems, it will not be possible to generate transforms automatically; as such, those descriptions will eventually be necessary if systems are to move beyond hand-coded transforms.

4 Pragmatics and Metrics

In selecting a data exchange standard, there are several aspects of the sharing problem that everyone needs to examine, whether or not they have additional specialized concerns (a few were touched on in Section 2):

- **Timeline**: how much time does the community have before a capability is needed?
- **Ambition level and risk tolerance**: Will managers accept some risk, in hope of transformative improvements, long term?
- **Legacy:** Are there significant legacy message standards? One or multiple?
- **Funding and incentive structure:** Are all players willing to invest? Will a central fund or major player support community needs? Will consumers fund descriptions and sub-transforms of producers? What is the time horizon for costs?
- **Technical skills**: which technologies are developers familiar with? Do all organizations have software developers?
- **Number of producers and consumers**: how many are there, both absolute and relative? What are their incentives (desire to exchange, who hosts, who funds)?
- **The rate of change of the domain:** How often are extensions requested, and with what timeline? Is there likely to be merger with another community?

We do not have a formal metric approach. It is easy to estimate the number of transforms to be programmed, semantic correspondences to be represented, and decisions of each type in defining a standard. However, many other issues are dealt with qualitatively, because there is no general means of estimating them. How much faster is it to describe an interface, rather than code transforms? Stakeholders will agree to an approach more quickly if multiple formats (including theirs) are supported, but that decision may have costs in terms of transforms required. Analysis and programming effort is also notoriously hard to estimate correctly. Finally, it is usually hard to trade political versus technical costs.

Rather than require decision makers to estimate all of these difficult parameters, our decision rules use qualitative terms (e.g., “many more producers than consumers”) which decision makers must interpret in their program context.

5 Decision rules – Exclusions

This work sought to simplify decisions for MITRE's front line engineers, few of whom are data researchers. For that reason, we tackled one narrow problem, and provide simple decision rules. The rules are based on the authors' engineering judgment, based on decades of experience supporting U.S. government interoperability efforts, observing both successes and failures.

We initially tried to formulate positive rules (i.e., "Use approach #1 under these circumstances") for selecting a standards approach but quickly ran into a problem: most rules had a long series of required conditions, making it difficult to specify modular rules. Even worse, a positive recommendation requires comparing with all alternatives – so it is almost impossible to be simple. Instead, we found it much more natural to formulate *negative rules*. These permit data integration experts to eliminate possibilities based on what they know of their situation, and narrow down the list of options. If all alternatives are excluded, then the community must change its stated requirements or assumed resources.

Table 1 shows rules for ruling out approaches, using the following labels as abbreviations for approaches 1 – 4:

- *1: Fixed*
- *2: Flexible*
- *3: Enriched*
- *4: Formal*

6 Conclusions and Future Research

We have presented important characteristics of data sharing communities with real-world examples, several approaches to data exchange standards, and some simple decision rules for how to select a data exchange approach based on the community's situation. The current popular approach (a simple standard XML schema) is the best low-risk, moderate-immediate-cost approach today, unless there are special situations of avoiding information loss or favoring producers over consumers. Even at one sub-transform per producer or consumer, programmers are needed, and coding, testing, and maintenance costs can be high ($millions/year, for USMTF systems working with the Air Tasking Order message).

For the long term, automated mediation offers hope of order of magnitude further improvements. Avoiding those costs will eventually require automated generation, driven by formal descriptions of participant systems' element semantics and representation. Automated generation is becoming practical today with approach 3; one can supplement the automatically generated transforms with manually coded ones wherever necessary.

Table 1: Rules for Excluding Data Standardization Approaches

Situation	Excluded Approaches	Justification and Comments
Want to minimize the need for programmers in generating transforms	1: Fixed, 2: Flexible	These approaches require programmers, testers, and accreditors. They are not agile, nor suited for in-the-field improvisation.
Transform-generator tools cannot be effectively deployed *at all*. Either: • Tool purchase plus skill-center costs seem too high, and no organization will step up to support them • No transform-generator product is effective (e.g., on very large schemas) • Short term focus with little concern for long term support	4:Formal, 3: Enriched	Approach 4 depends completely on mediation, so must be ruled out entirely. Approach 3 is more amenable to incremental adoption, since less capable partners can map to the physical schema, as in Approach 1. Notes: • Open source avoids licensing, but sometimes has higher people costs • If a vendor's tool suite is already in use, skills and metadata may be available. Adding a transform-generation capability product may be less expensive.
Does the community already have many transform-based exchanges working?	4: Formal	Automated mediation involves both ends' being documented formally; systems that have satisfactory sub-transforms will not invest in formal descriptions
The burden on producers must be minimized.	1: Fixed	Approach 2 provides flexibility for producers to use different formats.
Producers will provide data in any case, and do not greatly outnumber consumers	2: Flexible	Approach 2 greatly increases consumers' costs.
Data resolution needs (numeric, concept specificity) differ; the community needs to: • Avoid "lossy" exchange; or • Avoid overhead of high resolution formats	1: Fixed	Permits certain exchange pairs to have finer qualitative categories (e.g. left ventricle valve failure, rather than heart attack) or finer granularity. Other approaches avoid the least common denominator problem of Approach 1.
A typical message has many consumers	4: Formal	Multicasting a standardized message to all consumers can save bandwidth.

This work opens several avenues for future research. First, our categories and rules are just a first-step and allow ample room for refinement. Second, there is a need for empirical research to test our engineering judgments; experimentation or surveys are needed to see how systems engineers are guided by such decision rules, and whether they are satisfied with the results. Third, there is a continuing need for progress on automated mediators that support approaches 3 and (especially) 4. One important area is how to handle situations where auto-generation of a sub-transform is only partially successful. How should tools present the remaining work that needs to be done to systems engineers to maximize their effectiveness? Will the resulting code be maintainable? How should this be handled in cases where the consumer has no development resources and is willing to accept "best effort" data, as for example in recent "pay as you go" data integration research [11]?

Additionally, one can apply a similar methodology to other problems involved in data sharing, such as populating the exchange message (we considered schema, not contents), access rights, or transport security. One can also dig deeper into how values or codesets are specified and converted. Finally, the general methodology—create a bounded problem important to practitioners and derive simple, modular decision rules—could be applied in other domains.

Acknowledgments. We thank Prof. Harry Zhu, Rob McCready, Mary Pulvermacher, and the anonymous referees for their helpful comments.

References

1. P. A. Bernstein, S. Melnik, M. Petropoulos, and C. Quix, "Industrial-Strength Schema Matching," *SIGMOD Record*, vol. 33, pp. 38–43, 2004
2. A. Doan, P. Domingos, and A. Y. Halevy, "Learning to Match the Schemas of Databases: A Multistrategy Approach," *Machine Learning*, vol. 50, pp. 279–301, 2003.
3. R. Fagin, P. Kolaitis, R. Miller, and L. Popa, "Data Exchange: Semantics and Query Answering," *Database Theory—ICDT 2003, 9th International Conference*, Siena, Italy, 2003
4. E. Folmer, P.O. Luttighuis, J. van Hillegersberg, "Do Semantic Standards Lack Quality? A survey among 34 semantic standards," *Electronic Markets*, 21(2), 2011
5. K.G. Helmer, Ambite JL, Ames J, Ananthakrishnan R, Burns G, Chervenak AL, Foster I, Liming L, Keator D, Macciardi F, Madduri R, Navarro JP, Potkin S, Rosen B, Ruffins S, Schuler R, Turner JA, Toga A, Williams C, Kesselman C., "Enabling collaborative research using the Biomedical Informatics Research Network (BIRN)," *J Am Med Inform Assoc.*, April 2011
6. M.L. Markus, C.W. Steinfield, R. T. Wigand, G. Minton, "Industry-wide information systems standardization as collective action: the case of the U.S. residential mortgage industry," *MIS Quarterly*, 30(1), August 2006
7. R. Miller, M. A. Hernández, L. M. Haas, L. Yan, C. T. H. Ho, R. Fagin, and L. Popa, "The Clio Project: Managing Heterogeneity," *SIGMOD Record*, vol. 30, pp. 78–83, 2001
8. E. Rahm and P. A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *The VDLB Journal*, vol. 10, pp. 334–350, 2001
9. A. Rosenthal, L. Seligman, and S. Renner, "From Semantic Integration to Semantics Management: Case Studies and a Way Forward," *ACM SIGMOD Record*, special issue on Semantic Integration, December 2004
10. A. Rosenthal, L. Seligman, B. Blaustein, "Beyond the Sandbox: How Integration Researchers Can Actually Help Integration," *Workshop on Information Integration*, University of Pennsylvania, Philadelphia, PA, October 26-27, 2006
11. Anish Das Sarma, Xin Dong, Alon Y. Halevy: Bootstrapping pay-as-you-go data integration systems. *SIGMOD Conference* 2008: 861-874
12. P. Shvaiko and J. Euzenat, "A Survey of Schema-Based Matching Approaches," *Journal on Data Semantics*, vol. 4, pp. 146–171, 2005
13. K. Zhao, M. Xia, M.J. Shaw, "Vertical E-Business Standards and Standards Developing Organizations: A conceptual framework," *Electronic Markets*, 15(4), p. 289-300, 2005
14. Zhu, H., Wu, H. (2011) "Quality of Data Standards: Framework and Illustration using XBRL Taxonomy and Instances", *Electronic Markets*, 21(2), 129-139, 2011