



The Case for Master Data Services

Neil Raden

Hired Brains Research

Prepared for Initiate Systems

Contents

NEW WORLD ORDER: EXTERNALIZATION OF BUSINESS	3
Planning for the Unplannable.....	4
When Good Is Good Enough	4
GETTING THE RIGHT ANSWER	5
The Never-Ending Quest for the Single Version of the Truth	6
Good Enough: Instantaneous Data with Acceptable Confidence.....	6
MASTER DATA DOMAINS.....	7
Critical Domains: Customer and Product	7
Important Domains: Location, Employee	8
MASTER DATA SERVICES.....	9
The “Service” Economy	9
Confidence.....	9
TO PERSIST OR NOT TO PERSIST.....	10
MASTER DATA REQUIREMENTS FOR EMERGING NEEDS..	11
Mashups	11
Decision Automation.....	11
CONCLUSION	12
ABOUT THE AUTHOR.....	13

EXECUTIVE SUMMARY

Organizations of any size, especially those that have grown, tolerate at least a certain level of ambiguity, duplication and even conflict. Human nature is always a contributing factor, but often the roots of the dysfunction are in sheer expedience - things are the way they are because something had to be done and the broader or longer-term consequences were of secondary concern. Because people are able to function relatively well under these circumstances and make judgments and adjustments, the problem only becomes unbearable when automation is brought into the picture. Computers are not effective, or at least, have not been effective, when dealing with uncertainty and making judgments. The remedy, typically, is to minimize these problems through standardization, consolidation and management. The current trend toward Master Data Management (MDM) is one example of this phenomenon.

One paradox of human beings is that, while we are able to cognitively deal with fuzzy, incomplete information and rapidly understand its meaning, to date we have been unable to build computer applications with these qualities. Instead, our creations tend to be too rigid, too formulaic and not ambitious enough in vision. In MDM, for example, current best practices are to identify certain key domains of data that are reused across various applications and systems such as CUSTOMER or PRODUCT and to identify the mappings between an agreed-upon name and definition to all of the individual instances in the various applications. The premise is that a central, single source of reference data, a “single version of the truth,” can be developed to serve as a sort of Rosetta Stone¹, providing translation and conformance between the authoritative reference and the various occurrences of it throughout the organization. In some cases, this is intended to be a sort of library or system of reference with no active links to the various systems. In more ambitious cases, the links are active and used to verify, for example, new input data to operational systems as it is created. But in either case, the system is full of latency because the master database is only refreshed periodically and new elements are validated by administrators or “data stewards,” a habit carried over from earlier data standardization efforts such as data warehousing. In fact, the central database of an MDM solution may be its greatest drawback.

Largely due to the Internet, which launched business data communications into hyperdrive, the volume of data that organizations handle today is something unimaginable only a decade ago. But that isn't all. The number of participants in the interactions has also exploded, and this externalization² of business has emerged as a factor whose ramifications are only now beginning to be understood. In addition, the heterogeneity of data within organizations has not diminished even with the adoption of large, enterprise systems such as ERP and CRM. The simple truth is, data is coming too fast, in such great volumes and with such disparity that building a passive translation structure as the bulwark against this onslaught is just not an adequate solution. There simply may not be time to evaluate non-conforming data and to compromise with other parties to agree on a definition. Embedded deeply in many MDM solutions is a fallacy of perfection, the notion that all of the reference data can be standardized and maintained, but weighing the cost of perfection against the opportunity cost of delay or failure needs some closer scrutiny. There are situations where perfection is not only needed, it is mandatory, such as medical information that can have an affect on someone's health. But there are other situations where a level of error is tolerable such as in marketing information or in customer service applications.

¹ The term Rosetta Stone has become idiomatic as something that is a critical key to a process of decryption or translation of a difficult problem. From Wikipedia http://en.wikipedia.org/wiki/Rosetta_stone

² For clarity, the term externalization is used here to mean the widening of the plane and the number of participants on which business is conducted, largely because of the Internet and e-Commerce. This should not be confused with the economics terms “externality” or “externalization of costs.”



The Internet has changed that completely. The volumes that are transmitted are up by orders of magnitude. The hierarchical, one-to-many arrangement of the past, with only well-heeled parties participating is history. Today, anyone can connect to anyone else. Just in the United States, there are at least 2.5 million small companies, and by 2002, the last date there is authoritative data on this, at least 60% of these small companies had an online presence³. The affect of this externalization phenomenon is a dramatic increase, not only in the amount of data, including useful, actionable data, but also in the disparity of it. Instead of predictable flows of small amounts of reasonably decipherable data, today the information comes like a torrent and follows no particular format. It is not possible to reject this data, it has to be understood. Customers, suppliers and competitors can arise at any time from anywhere and failing to take advantage of the situation puts an organization at a competitive disadvantage.

Planning for the Unplannable

People in organizations are obligated to plan, but in practice, most people spend a significant amount of their activities dealing with things that were not planned. It is normal, even fashionable, to speak about the constancy of change in business, but when it comes to conceiving of ways to deal with integration, particularly data integration, the solutions that are proposed as industry “best practices” seem to overlook the concept of change completely. At the most basic level, they strive to normalize information driven by the assumption that there is some paragon, truthful, correct interpretation of the data and that the goal should be to strive toward this platonic ideal. This premise drives the design and the process, often creating rigid, centralized reference data repositories or “canonicals” that are proposed as the “single version of the truth,” to which all other operations should conform. The physical architectures are a reflection of a deeper conceptualization, and are too rigid. They do not embrace the sheer power of contingency^{4 5} and they overlook the process of continual change that is spoken about, but not implemented. “Change” in these designs implies changing what was formerly purported to be the truth, something the custodians of these systems are loathe to admit.

The current situation with the expanding externalization of business adds urgency to the need for more agility when dealing with data. When it is impossible to know who you will transact business with, it’s useful to be armed with systems and processes that are able to not only complete the transactions effectively, but to be able to learn as much as possible from the interaction and to be able to use the knowledge in the future. Planning becomes a secondary skill to being opportunistic, but this requires relaxing some cherished notions about data integration.

When Good Is Good Enough

Suppose your organization is a manufacturer of aftermarket replacement parts for motorcycles. In the past, you operated through a catalog that detailed every part and specification, in hardcopy, and your customers could phone or fax in orders, or give them directly to your sales and manufacturers reps. Each of these methods involved finding the right code or description so that there was no ambiguity. A few years ago, you put your catalog on the Web, and allowed a much broader range of customers to do business with you and eventually, once you got the systems hooked up, allowed to actually complete their purchases online. With some difficulty and a lot of expense, this worked fairly well. But in the last few years, more and more of your business arrives, not by buyers going to your website and conducting business there directly, but rather, they come through search engines, intermediaries, aggregators and even screen-scraping bots that aggregate information about you and your competitors.

³ “The Changing Market Economy: Vertical Industry & Demographic Profile of the Small Business Market” Caners In-Stat, July 2002

⁴ Mansfield, Harvey C. 1998. *The Prince*. Niccolò Machiavelli. Translated and with an Introduction by Harvey C. Mansfield. Second Edition. Chicago and London: The University of Chicago Press.

⁵ Meaning, making use of events and situations as they occur.

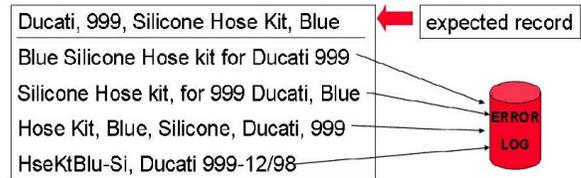


It is very unlikely that you will be able to match these parties to your persistent, scrubbed, correct in-house master data database. Moreover, by relying on your own, known descriptions and codes, it is unlikely they will present queries that you can satisfy. For example, suppose a distributor is pricing 20,000 hose replacement kits, a sale that would represent perhaps \$600,000. The RFP (request for price) arrives in this form:

Blue Silicon HoseKit for Ducati 999

But your catalog listing is:

Ducati, 999, Silicone Hose Kit, Blue



Field level matching cannot reconcile

This probably means:

Ducati is a motorcycle; 999 is a model of a Ducati;
Motorcycles use hose kits; Hoses are made from silicone;
Silicone hoses have color; Blue is a color

If your ordering system cannot deal with variations in the master data (no matter how many there are), you may reject this RFP and lose a \$600,000 sale in a few milliseconds. It may not be possible to match the information perfectly, but adopting a binary approach (accept or reject only) is a poor approach too.

One solution is to employ algorithms that can generate a probability score which becomes part of a model that decides the likelihood that a match is good enough. It may be acceptable to match a mailing address for a product promotion at, say, .85, but when assembling a patient profile from medical records, such a low probability would be inadequate. How the different probabilities are applied in practice is up to the individual application, but software that can determine the meaning of streaming data, in real time, and assign a probability to its translation is available today. In practice, where the need for perfection is high, the data requirements of the source system, that is, the system that is presenting the data, are usually high as well, so there is some symmetry to the process. This is typical, but not, unfortunately, universal.

GETTING THE RIGHT ANSWER

Computers still aren't very smart, so they must operate with explicit instructions and unambiguous data. Computers can be instructed to appear somewhat smarter than they are, though, but this is a tedious effort, especially when attempted for a single instance. Software vendors, by achieving some economies of scale, are able to provide more advanced capabilities than an individual effort can in most cases. Data integration is a particularly challenging problem for computers, which lack the native ability to understand the nuance of language. Sifting through various data sources and aligning them, MDM software addresses this problem by providing a set of functions and techniques that employ rules, matching algorithms and even standardized data for certain common elements like zip codes. But the way these tools operate can vary greatly. Some are designed to assist and facilitate the people performing data management and stewardship manually, others work in an automated fashion and still others combine both aspects.

More striking is the variety of underlying engine types – deterministic and probabilistic. The distinction between deterministic and probabilistic methods is that the former is composed of rules and some matching algorithms that lead to an accept/reject result. Probabilistic tools employ likelihood ratios and other statistical and data analysis theories to rank the likelihood that a record conforms to, or matches, another record. Probabilistic tools tend to be more scalable because, among other reasons, the algorithms are designed to use fewer machine cycles instead of exhaus-



tively firing rules and scanning lists. Deterministic rules must take a data identity condition (do records A and B match?) and fire through potentially thousands of rules to determine a binary match condition (yes, a match, or no, not a match). Probabilistic rules cast a wide net to first figure all potential matches, then to evaluate the likelihood of matches within the smaller set. Deterministic tools provide a certain peace of mind, but the productivity overhead penalties incurred through maintenance of rule sets and the “time to market” in putting the solution into production can be daunting.

The Never-Ending Quest for the Single Version of the Truth

Traditional data integration is based on an implicit understanding that once data is integrated and mapped, its mapping is correct. When issues arise, it is referred to as a “data quality” problem. This approach aims for 100% accuracy, 100% agreement and is applied uniformly, to all types and sources of data. Implicit in this understanding is:

- ▶ That substantially all interested parties and stakeholders agree on this mapping, or at least some procedural signoff has occurred
- ▶ That there is just one correct mapping, in other words, a “single version of the truth” that applies to all situations and to which variants should either conform or be formally linked
- ▶ That there is adequate time for this consensus to occur

Unfortunately, there are many situations where some or all of these assumptions are not possible. In order for all parties to agree, there is typically a watering down of the content, which leaves too many problems unsolved, or too large a percentage of the data remains unmapped, which overstates the “accuracy” of the mapped data. Within an organization, there may be more than one interpretation of the meaning of data, and in connected organizations, it is virtually guaranteed to be the case. Finally, the process of identifying, defining and socializing master data definitions is time-consuming, a luxury that cannot be justified in many cases today.

Good Enough: Instantaneous Data with Acceptable Confidence

A classic data integration technique is to define the steps and rules for integrating data from one or more sources to a single repository. In general, records that pass the tests are moved on to the repository and those that do not are shunted to an error log. In this scheme, one can say that the data in the repository is 100% correct, but in fact, its incompleteness is just another form of error. No measurement is made of the incorrectness to records not recorded. Perhaps many of the records could be correct, but the tightness of the rules eliminated them which could have the effect of reducing the value of the repository. Suppose, instead, that the records could be rated or scored and those with a score above an agreed-upon threshold could be sent into the repository.

This “good enough” approach would not be appropriate for air traffic control or a person’s vital signs during surgery because of the need for certainty. In contrast however, the ‘good enough’ approach is more than adequate to meet the needs of a marketing campaign or frequent flyer awards program. Probability thresholds can be tuned to the application. Matching algorithms can also be tuned by analysts, increasing the number of accepted records without lowering the threshold.

A typical deterministic matching application, which is based on rules developed by the analysts implementing and maintaining the project, will be much slower to react to learning and new situations. A probabilistic matching tool, because of the aforementioned economies of scale, is able to bundle



a wide range of advanced techniques out of the box. For example, creating a master product list with a few hundred thousand products (current and discontinued) from both internal and external source systems requires matching each product and mediating the ones that fail the matching algorithms. In a probabilistic approach, those that can be matched easily are handled first, and the rest as tested by the various algorithms and assigned a likelihood (0.0 – 1.0). Those elements that are “likely” matches, based on the likelihood threshold set by the organization, are captured automatically without manual intervention.

Another advantage of this approach is that it can take people out of the loop, allowing the MDM system to operate with streaming data in near real-time.

And, needless to say, with the explosion in scale and complexity of data systems, a probabilistic approach, which relies more on the computer to find the relationships in the data rather than the human, is more capable of dealing with large amounts of data, both in terms of the number of records and the complexity of the data relationships.

MASTER DATA DOMAINS

Master data is data that is used repeatedly to identify events, transactions and other instantaneous bits of information. For example, a temperature reading from a sensor in a refrigerated truck might be recorded as the actual reading (in degrees Celsius, for example) at a point in time and perhaps, with the serial number of the device. This information might be merged with routing information that appends the tractor, trailer, serial number of the refrigeration unit, driver and other information. All of these “tags” are potentially master data – but the temperature is not, it is the recorded value of the event. To the extent other applications use some or all of the tags, it makes sense to standardize the identifiers wherever possible. For example, if the refrigerated trailer is being tracked by cycle time or logistical information, not just temperature compliance, it is highly useful to identify it using the same set of identifiers.

But managing master data is very difficult because applications are developed and/or installed at different times, in different places by different groups and coordinating the master data is an extra effort and expense. Also, data is constantly streaming that contains these tags, often from outside the boundaries of an organization, and it has to be understood. There is also a lot of it. For that reason, master data management solutions tend to segment the job and implement it in phases, starting with the most useful or critical data. These different subject areas, or domains, have different priorities in different organizations, but in most cases, customer and product are always the most critical domains, though this can vary by industry. Location and employee are almost always part of an MDM solution, too, but often a lower priority. Beyond that, there are many other master data domains, such as financial, pricing, contracts, policy and others specific to certain industries, such as diagnostic code in the healthcare industry.

Critical Domains: Customer and Product

Though there are organizations that do not have customers per se, they at least have an audience that they serve, such as charitable foundations, religious organizations or even government agencies. Any organization engaged in commerce has products. For many organizations, having the ability to understand the customers and products in all of their applications, or even within certain applications, is an indispensable aid to efficiency and accuracy. When customers are people, it can be especially difficult because people’s names can be misspelled, abbreviated and even changed. People also cluster into households, affinity groups and geographical areas, but



the attachments are volatile. For every application in an organization to cleanse these listings and keep them current is an expensive process. The advantage of having a central reference process for customer data for the other systems to use is essential today.

Most organizations of any size have had a product master for a long time. The problem is, there are often more than one of them and, even more troublesome, they aren't necessarily used by different divisions or even different partners of the organizations so they provide no central, coordination facility, leading to delay, error and extra cost. Products also are grouped in many different ways. For example, to a compliance function that calculates average fuel economy for regulatory purposes, a Ford Escape Hybrid is an SUV, but for reporting national sales statistics, it may be considered a passenger car. Even when the identifiers of actual entities are the same, the taxonomic arrangement (hierarchies) may vary by application. A master data solution for product information solves this problem, not by creating one, rigid "system of record," but rather, by enabling the flexibility needed to manage a complex organization in a multi-layered environment. To do this requires a far more sophisticated and powerful set of tools that can handle all of the different contexts of the information.

Important Domains: Location, Employee

At first glance, it may seem that location should be a pretty unambiguous item, but closer examination reveals that it can be as troublesome as customer. A building may actually have multiple addresses, or have different names. An office building may be the Midwest Group Engineering Office for Subsidiary 'A' but elsewhere identified as the Missouri District Office for Flood Control. Needless to say, the hierarchical and historical attributes would be different, even for the same "location." Because of the suburbanization of metropolitan areas, the actual city name of a location can be very unclear. It may appear to be in one incorporated town or city, but actually be in another, or both! It may lie in an unincorporated area, even if the mailing address (and zip code) belongs to a city. This may not matter for mailing purposes, but it would for political purposes, such as voting or taxes or building codes. In addition, locations are not necessarily physical, they may represent areas or regions, or even stations in a manufacturing facility or the place of a traffic accident, or where a satellite is in the sky. Other types of locations that change constantly are sales territories/regions/districts/areas (even the naming of these categorizations change) – especially across companies who are acquiring others. Also, marketing regions that overlay the geopolitical delineations of countries, cities, provinces, etc. can cause real headaches when standardizing data. Lastly, there are "names" of campuses and buildings instead of addresses. Location can be a very complicated domain.

Employees are another complex entity. They are people and thus have all of the same human idiosyncrasies as customers, but the definition of an employee, and the value of the attributes associated with them, can vary widely across an organization. In some organizations, a research university for example, an employee may hold a number of jobs simultaneously, some with pay and some without.

All of these complexities may seem more like data modeling issues than data integration, and they are, but when it is necessary to blend information from more than one source, cataloging this information and creating a comprehensive cross-reference for the terms is useful, but it is not sufficient in today's world of high-speed, large-scale communications not only within your organization, but beyond it. If we employ the metaphor of a physical library, what happens when the speed of the people entering the library doubles every 18 months for 30 years and the number of visitors climbs exponentially? Those carts with returned books that have not gotten replaced on the shelves represent opportunities lost when master data can't be managed in near real-time. To do that requires more than MDM, it requires a master data service.



MASTER DATA SERVICES

By now, anyone remotely involved in technology has heard of the term service-oriented architecture (SOA) and Web Services. The concept of services as opposed to applications is not new, but it's been supercharged by the widespread adoption of a set of non-proprietary standards, loosely known as Web Services. Web services were designed to simplify and expand the use of the Web as a platform, not just a collection of HTML documents. While the W3C has moved forward with these standards for the Web, enterprise software vendors and IT departments recognized the value of a service-oriented architecture, not only for doing business on the Web, but for their entire application portfolio.

The "Service" Economy

In essence, a service advertises its function in a directory, including what it can do and how to communicate with it. In practice, services tend to be smaller pieces of functionality than an entire application so that, at least in principle, they can be combined easily to provide specific functions, and just as easily decomposed. This quality is referred to as being "loosely-coupled." The problem with the services concept is that it has the potential for both brilliant application and dreadful abuse. No controls along that axis are part of the picture.

Nevertheless, a service-oriented architecture opens the gates for some powerful applications in MDM, such as a master data service. As its name implies, a master data service is devised to be a more flexible, agile and real-time variant of a single-database MDM hub.

A successful master data service is an MDM solution delivered as a service for one kind of master data. There may be multiple instances of the master data service for different data types, but they will be orchestrated by some higher-order service. This is a point of departure from earlier MDM solutions that attempted to gather the master data and control its use from one central location. By being compatible with the various business processes that create the data, such as sales-to-fulfillment or strategic sourcing, a master data service can attend to the scalability, tenability, reliability and maintenance requirements much more effectively. Because of its architecture, operating as a cooperative Web Service, it has some additional features that are not found in most MDM solutions. Most importantly, it is designed to respond to messages in real-time. For that reason, it must be able to assimilate new information and make it available almost instantaneously. Since that precludes the review of a "data steward" to approve the additions, deletions and modifications of master data, a different mechanism is needed to decide if the new information is correct or not without human intervention (though that can happen later). A master data service cannot operate on the assumption that a single master database is the sole source for the correct data. There may not be time to update it between queries, certain information may need to be cached and, in fact, some other sources of data may be reliable, accepted sources for the correct data. This would require the master data service to be able to decide, formulate and execute federated queries to gather the data from more than one source, instantaneously.

Confidence

The purpose of any MDM solution is to provide confidence in the data to all of the stakeholders who use, manage, consume and provide it. Confidence in a well-tended subset of data is only partial solution. Delivering confidence, in a much wider set of data, provided on demand in real time, is the type of MDM solution that can capture and retain ROI. Deterministic matching solutions cannot provide the kind of response time and confidence level scoring that a probabilistic solution



can. A probabilistic solution can put a greater percentage of the data into production and can do it with less latency (real time) because it eliminates the manual checking and consensus needed in deterministic approaches.

TO PERSIST OR NOT TO PERSIST

A common approach in IT is to solve problems in pieces. The presumption is that a complicated problem can be solved by addressing its constituent parts. This is known in science as reductionism. The assumption is that understanding the entire problem can be achieved by describing the nature of the parts and the aggregation of these descriptions represents a reasonable map of the problem space. There are many cases where this is a valid approach, but there are many others where it is not. For example, even if you put the pieces of dissected frog back together with infinite dexterity and attention to detail, it still won't hop off the table. There are qualities that are greater than, and transcend the whole. This concept is called emergence. It refers to how behavior at a larger scale of the system arises from the detailed structure, behavior and relationships on a finer scale.

When designing an architecture for something as complicated as MDM, most architects tend to favor simple building blocks, or reductionism, instead of the more challenging whole-system design. By optimizing the pieces, separate databases, predictable data flows, server tuning, etc., the assumption is that the whole design is likewise optimized, but this overlooks the effects of emergence. For example, starting with a block and arrow design, it may be decided that data will flow to a permanent repository where updates and access can be tightly controlled and monitored, overlooking the potential value of federation of data, virtualization, abstraction and caching, approaches that would not be considered in a reductionist, or "one piece at a time" approach.

This optimization may seem like an insurmountable problem, one that needs to be addressed by some simplifying assumptions and rules of thumb. If every DBA in every organization had to figure it out, and keep it tuned, it would be an insurmountable problem. But there is a general case of the problem that only needs to be solved once and implemented as a software application with a set of heuristics and predictive models, freeing the individual organization from having to engage in basic science to solve the problem each time it is encountered. Software providers can leverage the work of researchers and apply mathematical algorithms from covering and partitioning, bin packing, dynamic storage allocation, graph theory and a host of other esoteric techniques. Using this body of knowledge, they can develop heuristics in a system that can quickly make sense of the current environment, produce the best answer for optimizing a single query⁶, load balancing the current streams and, most importantly, making recommendations or, even better, continuously modifying the structure as use patterns change. Consider an everyday example:

In a current hybrid vehicle, like a Toyota Prius, besides the usual dashboard controls, the driver also faces a touch-panel monitor in front of her. Graphics indicate the direction the energy is moving, pointing to icons that represent the engine, generator, motor and battery, with the arrows constantly changing direction as the internal management systems attempt to continuously balance the load and maximize fuel economy while maintaining other levels such as speed and acceleration, braking, comfort, etc., all of which is controlled by a computer not much larger than a thumbnail. This is an excellent metaphor for picturing how information systems need to be configured today – many components operating in harmony under the control of a very sophisticated management system that can assign priorities, manage workloads and make optimized, on-the-spot decisions about resources and processes.

Conventional IT thinking still centers around a physical database as a pivot point in most systems, but centralized, persistent storage may only be a good solution for a portion of MDM data. Integrating

⁶ This should not be confused with query optimizer that are part of a data management system, which are designed to find the best solution to solving an active query and are proprietary to the database software. In the case of distributed information, an optimization is needed to decide which data sources to use, which may span multiple databases, caches and even data structures without their own optimizers.



data in stream, with probabilistic matching, caching and federation of queries to multiple systems for assigning a match in real-time are desperately needed for today's emerging architectures and applications. Enterprise application architectures are gradually migrating from a premises arrangement, where all of the resources are located in one (logical if not) physical location, to a distributed, standards-based, loosely coupled architecture of services that can be configured into applications more or less dynamically. The location of data, the ownership of data and meaning or semantics of data in this scheme becomes more abstract, more fluid and more dynamic. Like the old song goes, trying to corral it is like trying to catch the wind.

Emerging architectures not only affect the nature of master data, they are having a considerable impact on how data is used and the applications that arise from these new uses.

MASTER DATA REQUIREMENTS FOR EMERGING NEEDS

Conventional thinking about master data usually draws on a few common industry examples, such as mailing list cleansing, product masters or customer information for touchpoint applications. In some ways, these familiar applications flavor the thinking about MDM approaches by predefining the requirements based on what is already known. Think ahead a little, a look at two newer applications that can be enhanced with MDM can provide a more expansive set of things to consider when evaluating solutions.

Mashups

A mashup is a term used in Web applications to describe combining separate applications (or services) to create a composite experience. A common mashup is to link address information to Google Maps so that the user can see, physically, a location that is presented as only a textual address. Many large Web applications and many smaller ones too, are already providing Mash Editors to facilitate using their services.

Without MDM, any application or service would have to define the semantics of the data elements used in an application to provide the ability to mash up the information. That would require a complete view of the data scattered about an enterprise to take advantage of the mashup, which would complicate the design of the service, adding heft and maintenance burdens. But, if you manage master data, it is always identified as such from whatever application uses it. For example, any website that displays a business name can be automatically linked to a mashed up map, or directions, or telephone number because the relationship between these values is formalized in the master data service. It doesn't require modeling, programming or maintenance. The semantics of the data elements are abstracted through the MDM, opening the door for an infinite variety of mashups, and leveraging the power of SOA.

Decision Automation

The image projected by an organization is largely the result, not of marketing campaigns or even the products and services that it provides, but of the decisions it makes. This includes not only the major strategic decisions that form the campaigns and products, but also the myriad small decisions that its many employees make on a day-to-day basis. Decision support, knowledge management, business intelligence, data warehouses and other off-line aids for manual decision making, such as spreadsheets, have served to inform these decisions, but usually only after the fact. But in the last few years, the near universal adoption of SOA, work flow and business process management (BPM) are making it possible to move the entire decisioning process into real time, whether human assisted or fully automated.



What do you need to effectively deploy sophisticated decision automation in your organization? First of all, you need access to data, and lots of it. Descriptive and predictive modeling, also known as data mining, do not operate on small sets of aggregated data, such as that in most cleaned up data marts. These tools need access to lots of detailed data, which implies a fairly robust MDM capability to understand and keep the analysis consistent, especially when the data is scattered throughout and beyond the organization.

Because people from different domains, at different levels of skill, need to participate, there is a need for software tools that can accommodate this diversity and most importantly, create a closed loop of analysis, model formation, rules, operational systems, actions and reactions and continuing analysis and refinement of the models. No handoffs, no structural latency. The benefits of a decision automation architecture are:

- ▶ **Precision** – making more profitable and targeted business decisions.
- ▶ **Consistency** – interacting with customers the same way, regardless of the means of interaction.
- ▶ **Agility** – being able to quickly adapt to changing business conditions.
- ▶ **Cost** – providing the ability to increase the scale and scope of decision management with only an incremental increase in cost.
- ▶ **Speed** – returning decisions in as near real-time as adds value for both the organization and its customers.

Only a fraction of organizations have deployed true decision automation, but a large proportion of executives surveyed felt that front-line decisions impacted bottom-line profitability⁷. Seventy percent of CIO/CTOs said that they did not get the most value they could from their data. In an environment where it is harder to eliminate costs using technology, where there is a push for in-stream decisions and straight-through processing, decision automation embedded in operational systems and processes is a solution. But it cannot work without master data services, providing data integration in-stream and built to cooperate in distributed, loosely coupled frameworks.

CONCLUSION

MDM is a must-have solution for leveraging SOA and coping with the externalized enterprise. Unfortunately, much of the discussion, best practices and tools of MDM are old school – based in traditional methodologies and thinking about data management. MDM is a challenging, on-going problem for most organizations, and MDM tools cannot solve all of the problems. However, thinking expansively, not as a reductionist, MDM solutions that are designed to be real-time, to exploit systems and resources already in place, to intelligently balance the location of the work and the data are the key for deploying the next version of business processes that exploit the technologies already available.

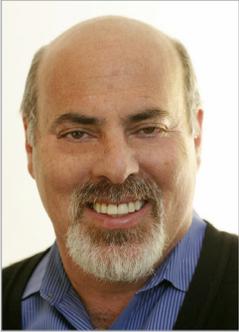
A master data service is designed to exploit the inherent advantages of SOA. However, SOA can be a blessing and curse. Like any other technology, its promise can be wasted when misapplied. Using old design techniques and methodologies, failing to make provision for the potentially chaotic interplay of services or failing to understand the longer-term maintenance implications can lead to poor return on investment or even failure. A master data service can increase the confidence in master data by being more timely, by being precise about the likelihood the data is correct and by handling greater volumes of data and requests. This latter factor is extremely important in understanding the emergent qualities of data rather than attenuating them with aggregation and summarization.

⁷ Opinion Research Center, 2007



A rigid master data solution, conceived with last generation methodology, built with premises-oriented data management tools, cannot meet the demands of distributed, cooperative architectures such as SOA and the next-generation Web. Master data services are perfectly positioned to enable the applications being conceived and built now.

ABOUT THE AUTHOR



Neil Raden, based in Santa Barbara, CA, is an active consultant, widely published author and speaker, and also the founder of Hired Brains, Inc., (<http://www.hiredbrains.com>). Hired Brains provides consulting, systems integration and implementation services in business intelligence, decision automation and business process intelligence for clients worldwide. Hired Brains Research provides consulting, market research, product marketing and advisory services to the software industry.

Neil was a contributing author to one of the first (1995) books on designing data warehouses and he is more recently the co-author with James Taylor of *Smart (Enough) Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions*, Prentice-Hall, 2007. He welcomes your comments at nraden@hiredbrains.com or at his blog at Intelligent Enterprise magazine at <http://www.intelligententerprise.com/blog/nraden.html>.

ABOUT INITIATE SYSTEMS

Initiate Systems, Inc. enables organizations to strategically leverage and share critical data assets. Its Master Data Management (MDM) software and experience as an information exchange leader provide organizations with complete, accurate and real-time views of data spread across multiple systems or databases, even outside the firewall. This allows companies to unlock the value of their data assets for competitive advantages or operational improvements. Initiate Systems operates globally through its subsidiaries, with corporate headquarters in Chicago and offices across the U.S., and Toronto, London and Sydney.

For more information, visit www.InitiateSystems.com.

