

# Governance Methodology For Information System Restructuring

## Publishing manager:

Pierre Bonnet, Sustainable IT Architecture (S-IT-A) community founder  
[info@sustainableitarchitecture.com](mailto:info@sustainableitarchitecture.com).

## Version:

V1

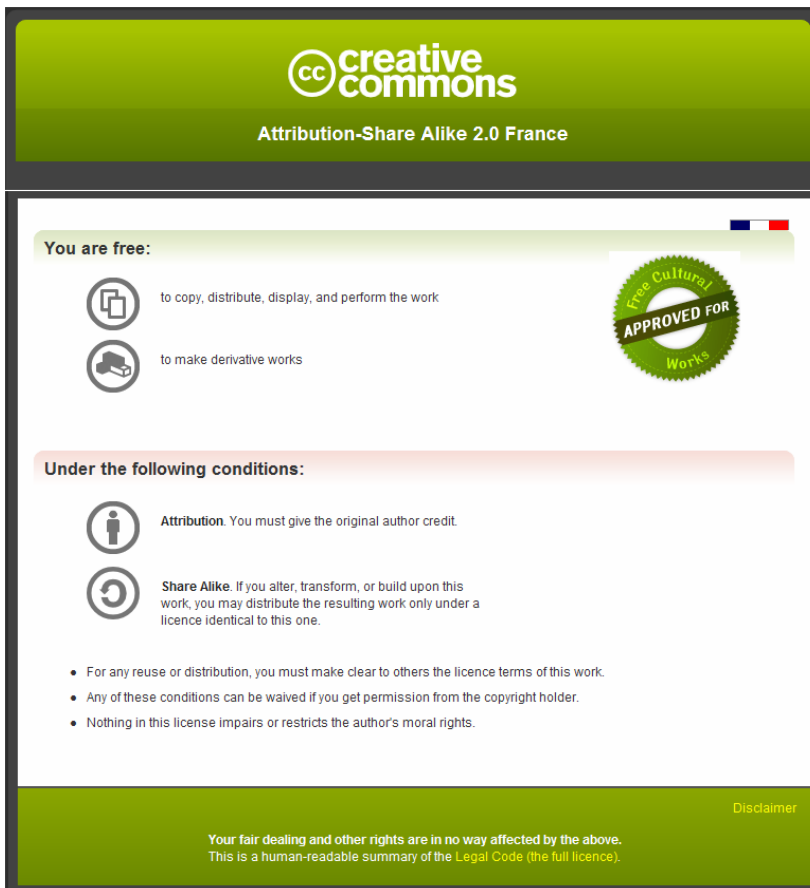
Last version available on the Sustainable IT Architecture website:  
<http://www.sustainableitarchitecture.com/home>

## Authors:

Pierre Bonnet, Orchestra Networks, [pierre.bonnet@orchestranetworks.com](mailto:pierre.bonnet@orchestranetworks.com)  
Jean-Baptiste Dezard, Ilog, [jbdezard@ilog.fr](mailto:jbdezard@ilog.fr)  
François Rivard, Logica, [francois.rivard@logica.com](mailto:francois.rivard@logica.com)  
Fabien Villard, IS expert, member of the S-IT-A community, [fabien.villard@praxeme.org](mailto:fabien.villard@praxeme.org)

## Protection:

This document is provided under a Creative Commons license with rights as follows:



The image shows a Creative Commons license banner for Attribution-Share Alike 2.0 France. It features the CC logo and the text "creative commons Attribution-Share Alike 2.0 France". Below this, it lists the freedoms granted: "You are free: to copy, distribute, display, and perform the work" and "to make derivative works". It also lists the conditions: "Under the following conditions: Attribution. You must give the original author credit." and "Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one." A "Free Cultural Works APPROVED FOR Works" seal is visible. At the bottom, there is a disclaimer: "Your fair dealing and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full licence)." and a "Disclaimer" link.

## Inside a company, today...

- CEO: We must quickly adapt our Information Systems in order for them to support change and innovation. Our business evolves, regulations increase external pressure and our customers require more and more services with increased added value. Change is everywhere and IT must not impose a barrier.
- CIO: We are good at maintaining our legacy systems up but to respond to your demands we have to undertake significant change and activities such as to rediscover the knowledge of our data currently trapped in too many databases which are often old and implemented in opaque commercial software. Its documentation is often outdated.
- CEO: Why not change the whole system by using only one modern commercial package and by outsourcing efforts to service providers specialized in this kind of business?
- CIO: Then where will be our control on our systems if software providers and outsourced teams are entrusted with them? We would accept a momentous risk of finding in the new system the exact same things we consider as obstacles today: bad control on our data, heavy maintenance, risks of dependence on service providers and loss of knowledge within our own teams...
- CEO: Are you trying to explain that IT will remain an obstacle to our change and innovation needs? It is imperative we find a quick solution to get better IT systems. More responsive and more reliable to support the business and regulation change. This is a strategic need conditioning our company performance. How do our competitors cope with the issue?
- CIO: There is a less intrusive way of achieving this goal for our legacy systems but it represents a drastically different way we change things. By creating a business rules repository and a master data repository we will be able to manage business rules and master data as business assets with much less pressure on legacy software and databases. Thanks to those repositories, opacity of legacy systems would be reduced without big-bang modifications.
- CEO: Repositories? Tell me more.
- CIO: These repositories are built with the documentation for business rules and data shared across the company. We call them reference data or master data. For example they include the description of our business products, our customers, organization entities and parameterization data for our business processes. Business teams can browse and audit the repositories and IT teams can use or execute them with legacy systems and of course with new ones. Instead of having data hard-coded in software with an IT-only readable language, data is stored in a form understandable by business teams and instantly usable by IT teams.
- CEO: Well, what about this with commercial solution such as ERP?
- CIO: It's also valid when running an external suite provided that the suite is able to connect to those repositories. IT solutions today include well known packages to accomplish this. They are called rules engines and reference data management systems and are technically known as BRMS (Business Rules Management Systems) and MDM (Master Data Management). These kinds of packages with the current level of maturity did not exist some years ago.
- CEO: Interesting. But what about our business processes in this landscape?
- CIO: A process repository is added to the apparatus beside the MDM and BRMS. Processes handle business rules and master data. This third repository needs a BPM (Business Process Management) system. Moreover we already have started to install this system. But we had not imagined at this time to extend this repository strategy to master data and business rules. They are still trapped in legacy systems. With the three repositories we would be able to increase the ability to audit our systems and manage a better transition between legacy systems and new ones by using either in-house developments or external components. Agility, which is the ability to quickly change our systems behavior according to

business needs, would also be increased thanks to the parameterization inside the repositories instead of the systematic modifications of programs.

CEO: Well build ASAP an action plan for this idea and show us how our competitors use this change management strategy.

CIO: Other companies have already started this strategy and a community of IT providers is as a matter of fact specialized in this domain. It publishes best-practices to help bootstrap the creation of described repositories, including real use cases. The complete approach is called by the community "Sustainable IT Architecture". They recommend a sustainable approach to build systems by intensive use of business repositories as just described.

## Like that is so obvious now

This CIO is right. Sustainable IT Architecture (S-IT-A) community prescribes precisely that, and others share the same ideas though often with other terms. For instance the market research company Forrester states: *"Most enterprises still embed process, rules, and reporting in applications. In other words, process flows, rules, and analytics are hard-coded into individual applications. It's hard to even find these definitions when they're mixed in with other application code, and making changes requires lengthy QA procedures."* (How The Convergence Of Business Rules, BPM, And BI Will Drive Business Optimization, Forrester Research, Inc., May 2008)"

It is obvious: why should we continue to hard-code rules, master data and processes inside program code when we can establish MDM, BRMS and BPM repositories? Maturity for these kinds of tools and methods are now enabling success, and it does not require a big-bang approach, but progressive and planned change in compliant with risk management. Coupling these three repositories is decisive. Together they form the powerful concept called ACMS - Agility Chain Management System: no agile processes without agile rules, and no agile rules without a business management for reference data. Each of these three profits from advanced governance functions including version management, rights management, traceability management and so on...

Let's consider an example. A business regulation, for instance Sarbanes Oxley or Solvency II, requires financial reports to be auditable. Which means that the company must be able to display data and computation rules used to build reports and the proof the data and rules are the right ones. Commonly in this context the company may at best show auditors documentation on paper which obviously cannot be executed by IT systems. At worst documentation for IT tools is obsolete and only specifications or even worse program codes are available to be analyzed by auditors. And because auditors are not IT technicians they won't be able to use these materials. This could be accepted in some cases but what if auditors insist? What if independent IT rating agencies appear to rate Information Systems by request of shareholders? The same auditors by accessing business rules and master data repositories could easily audit financial elements by scrutinizing rules and data usages. This would be possible because the repositories systems include business oriented UIs and not technically-only oriented ones.

To be fully convinced that this is so obvious and real, we have to understand the following points. First, tools and methods are available to build rules and master data repositories and have reached a maturity level never seen before. This maturity level authorizes a progressive deployment of the repository strategy across the whole Information System and provides a maximum benefit. After the Y2K crisis some leaders from the information industry started to specialize in technologies aimed at increasing agility and regaining business knowledge trapped in legacy systems. Legacy systems are often too complex and badly documented, even when we speak about off-the-shelf commercial software. This has been the primary driver leading to the birth of BRMS and MDM tools. To support these tools in early projects methods have been used to discover business rules and to model master data. Since 2002 some innovating companies have started to deploy BRMS and MDM systems on a large scale. They have since been able to stabilize technologies and

methods and prove that the technologies are suitable in production environments and on big projects, while keeping response times in control both for real time and batch usage. Barriers to the exploitation of such systems have been broken and good business adoption has been clearly shown. These real customers use cases are presented in the book published in the Sustainable IT Architecture community framework<sup>1</sup>.

Then, we must also understand that deploying repositories is usually a progressive task. An enterprise starts modifications on legacy systems by moving out some business rules and by adding an MDM repository in front of legacy databases. Legacy databases are not touched in this phase of the transformation process. In less than a year benefits may be obtained from rules and master data repositories by the increase in traceability and agility. Business gets at last, access to information system's assets with a business point of view which cancels the common opacity lying between business and IT people. Commonly, and as an important target of the process, enterprises should also see an increase in the business knowledge both on business rules and reference data, thanks to the recognized modeling efforts and the documentation obtained from repositories. It is important to keep in mind that this documentation is directly executable by IT teams and systems: this is a guarantee for good alignment between business and programs and an increased auditability of systems, including answers to regulation requests.

After a first period using methods and tools to control repositories and their adoption for integration with legacy systems, the next step is commonly a progressive rebuild of all systems to get even more agility and to abandon obsolete platforms if useful.

We are now going to give more details on principles used to modify legacy systems, to overhaul the IS and then to establish repositories when we are in a context using commercial software.

## Act on legacy systems

The objective is to encourage legacy assets handling without the technological constraints imposed by their implementation. Two threads must be started: the first one about reference data, the second one about business rules.

First of all, an information model must be designed to bring a detailed business view on data structures handled in legacy systems. This model must be based on a global organizational data scheme to encourage weak coupling between data aggregates which group data sharing a strong semantic cohesion. For example, models for real estate assets must not be coupled with models for addresses because addresses are reused to describe actors and other real world objects. In the same idea clients must not be modeled separately: it is better to model persons and consider a client as a person's role toward another object like a contract. Some parts of the common model form the basis to build the repository for reference data, i.e. data shared between systems which are often the primary target for business auditors. To avoid the blank page syndrome we may choose an off-the-shelf data architecture model like the one published by MDM Alliance Group (MAG) which is focused on processes for reference data modeling. This model defines the most generic levels of the data design scheme, the most universal concepts which have a great chance to be suitable to whatever activity domain in the enterprise. We call them "Business Object Domains" and "Data Categories". For example, there is a "Real Estate" Data Category inside Business Object Domain called "Reality" which gathers objects from real world. Designing business objects used to describe real estates may be drastically different if the company is an insurance company managing real estate assets for leasing or an army managing barracks. In all cases the Data Category and the Business Object Domain are the same. There is another important point in this example: address description according to the global data architecture is located in a separate Data Category called "Geography" and this avoids duplicating address model

---

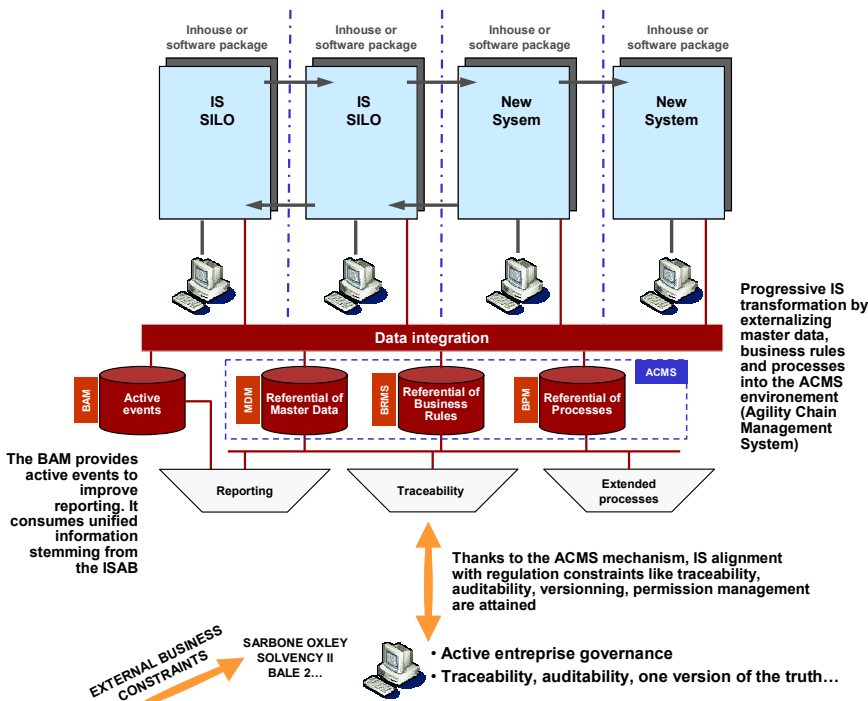
<sup>1</sup> See reference at the end of this document

management.

To install the MDM concept, a Common Information Model (or CIM) must be built. This model is also used to handle data integration flows between the MDM system and legacy systems databases. Legacy systems are left running as usual but their reference data are governed by the MDM system. This repository must be able to host all data kinds without limitations: product descriptions, clients, organizations, charts of accounts, real estate or transferable assets, functional and technical parametrization of systems, etc... To achieve such results a generic MDM system must be used<sup>2</sup> which must not enforce type or complexity limits in models they manage (associations between data, hierarchies, data life cycle...). Without this generic MDM system, enterprise would be forced to build many repositories which hence may be seen as reference data silos when the target is to rationalize legacy systems. On the business rules side, some algorithmic parts must be extracted from existing code to be rewritten as rules in the BRMS. To replace the diverted code, a call to the rules engine is added. Business data routed by the call to the rules engine are formatted with respect to the Common Information Model introduced above. The BRMS and MDM system share the same data models which are independent from physical representations often heterogeneous and living in current systems. Rules discovery is not an easy task. It requires collaboration between business and IT teams. Together they analyze IT systems and discover or re-discover rules hidden inside. It's a long term job which must be conducted with proper tools: automated code scanners can help paving the way even if they are not able to automatically find the rules. For example when business people decide to make an inventory of all rules acting on the data "Credit Rate" we just have to pass all the legacy code in current scope through the code scanner to discover processes handling this data either by modifying or viewing it. Process cleaning, formatting according to business rules grammar and legacy code refactoring lead to a preservation of existing programs while including a rules engine invocation in place of the old code. In the same process a regression test strategy is positioned on rules and will be used

during initial acceptance process and successive maintenance tasks.

A natural coupling appears between the two repositories. Some reference data validation rules are instantiated in the BRMS and on the other hand business rules handle reference data which benefit from being managed in the MDM system. A combined governance for those two repositories (MDM system and BRMS) completed with the standard BPM system is then established to determine business and IT commitments. On one hand a new charter for maintenance is used to establish use cases



for MDM system and BRMS according to maintenance types: feature release or bug-fix. Metrics are included in this charter to rate the usage level of the MDM system and BRMS and avoid

2 Also named 'Model Driven MDM'

atrophied or excessive use.

### Testing transformation to be convinced

Starting with the data scheme proposed by the MAG community<sup>3</sup> a first design area in the Common Information Model is selected including two Data Categories with a hundred data. Modeling these data may be done in a few days. From this model an XML representation is produced to be absorbed by the MDM tool. The master data repository is then ready-to-use. All governance functions like version management, viewing and data input screens, API services and so on, are generated automatically. The MDM tool must give this flexibility: this is a model driven MDM. Using an integration data layer like EAI/ESB/ETL MDM system and application databases that consume the data must be synchronized. A first set of business rules is identified in the reference data validation context. They are capitalized in the BRMS which shares the same data model found in the MDM system. The BRMS tool must also accept an XML representation of meta-data. At this stage, that may be done within few weeks, a proving system is obtained showing the benefits from these two repositories in a way that business people are also able to rate.

To complete the proving system some business rules may be extracted from legacy systems and inserted in the BRMS. A technical architecture must be designed to authorize those legacy systems to invoke the BRMS. An XML invocation will often be preferred but it is not mandatory. During familiarization with the technical design, at least with the proving system, benefits from code scanner tools that help to discover rules and to re-factor code may be measured: rules discovering and code re-factoring are needed to position correctly BRMS invocations in legacy systems. In the end rules located in the BRMS repository may gain benefit by using reference data already present in the MDM repository.

On the other side the Common Information Model in the MDM system and some business rules are completed with indicators used by a BAM supervisor system (Business Activity Monitoring) to follow behaviors of assets included in the two repositories and also the more traditional BPM. For example the reference data "Product Type for Family X" may be accompanied by an indicator watching the variations of this data enumeration country by country. Each country has a threshold for product type modifications that throws an alert when exceeded. Key assets implemented in the BRMS and MDM systems are then under active governance, i.e. a dynamic watch which becomes easy to install. On the other hand it would be complex and not reliable if indicators were built from legacy systems where business rules and reference data are diluted in functional and technical silos. In other words deploying a BAM system should follow MDM and BRMS systems deployments, not the opposite.

## Specific development of new systems

In case of progressively overhauling a legacy system or building new IS domains, an enterprise will get considerable know-how momentum if it has already done things according to the "act on legacy" principle described above. In fact new developments will naturally build on master data and business rules repositories. The Common Information Model will be seen as the unified language for specific developments to describe information. It is important to remind that this model is not built by technically sharing the legacy physical formats found in databases; it results from a business modeling process built on the sustainable data scheme which guaranties its validity when used for new developments. Of course this model is not immutable; it will evolve according to new needs with respect to the data scheme established during the first step of modeling. The data scheme will help to increase isolation from physical data layers.

3 <http://www.mdmalliancegroup.com>

Rules repository will also be reusable and will be enriched with new rules discovered when overhauling or designing new IS areas. In case of an overhauling strategy preserving identical functions, aimed for example at clearing off an obsolete technical environment, capitalized test cases bundled with business rules will be a determining plus to detect regressions and to reduce risks.

If the enterprise does not act on legacy systems, it will nevertheless have to design the Common Information Model not only in the point of view of new developments needs but also in the sustainable information scheme spirit. We strongly advise to build on a universal and reusable framework rather than start from a blank page<sup>4</sup>.

In order to host new developments a methodological add-on will help define the components structure which will probably take advantage of a service oriented approach. Processes reorganization will take a bigger part because components re-factoring will authorize a potential review of orchestration embedded in business processes which won't be limited by artificial bounds created by functional and technical silos. Processes implemented in a BPM tool will be based upon business rules repository to drive the orchestration of steps and rules will extract their reference data from the MDM system. This re-factoring will also bring opportunities to enrich management indicators and their benefits we have already discussed.

## Commercial software deployment case

In recent years a wide experience in commercial software deployment has been brought by waves of ERP and CRM solutions deployments. With a little step back many enterprises face the same issues with their commercial solutions than those having chosen internal developments. An increasing weakness is observed in business knowledge management: knowledge is lost while deploying the commercial package. From an economic point of view, commercial products prove themselves as costly as specific developments because license models for product updates established by editors reflect their well anchored position in enterprise IT systems. And finally, commercial products editors operated in the same way as for specific development by embedding business rules and reference data in IT languages often completed with parameterization principles. This partly shows the MDM idea but it does not bring the Common Information Model and a sufficiently business oriented tool.

Despite these facts not being in favor of commercial products, some enterprises find less risk in a commercial package orientation to overhaul their legacy systems rather than starting time-consuming specific projects that may fail due to a lack of competencies in modeling and architecture. This is a well-grounded point of view that makes sense for some enterprises. On the other hand and according to what we described above, can we accept to completely avoid the MDM and BRMS repositories when following a strategy based on commercial products? In other words what is the durability of a new system built with commercial products if at the end business rules and reference data are trapped in a technical language or a technical parameterization inside a commercial solution? Will this situation be more acceptable than legacy systems we try to discard?

Choosing a commercial product is still reasonable provided that the solution can live with the enterprise level MDM and BRMS repositories which means that the repositories are not trapped in a proprietary architecture inside the boundaries of the commercial product. The existence of the repositories in the commercial solution may be seen as a real strategic advantage but it is not sufficient. They also need to be independent from the technical and functional framework provided by the product to be fully reusable outside of the scope of the product if necessary. This is a reversible strategy that gives opportunities to deploy an applicative framework different from the

---

4 See the MAG community

one provided by the commercial product initially chosen. This is the only way to avoid known problems with commercial products already deployed in companies and to get the key advantages of deploying repositories. Though keep in mind that this implies that the effort to design a Common Information Model cannot be avoided because it is mandatory to handle data management in the MDM system. On the business rules side, hard-coding rules in the commercial product is no longer a solution. Instead, a technical platform is required to be able to invoke the BRMS from the commercial product. A product without this ability may be candidate to rejection. The maturity of commercial process repository tools (BPM) is globally good because they offer an externalization in enterprise workflows. It belongs to customers to require from their providers the same openness toward MDM and BRMS systems.

It has to be noticed that the same approach is also recommended for an already deployed commercial product in order to regain control and increase its ability to be aligned with business: externalize business rules in the BRMS and position a Common Information Model to handle reference data management in the MDM system.

## Success factors

Success in gaining control on IT systems needs willpower, a vision and help from specialized skills for governance, methodology and tools supporting repositories. It is better to avoid reinventing the wheel and take benefits from experiences capitalized upon since 2002.

Main principles driving governance mandatory for our IT systems metamorphosis unfold from the description of the MDM and BRMS repositories usage stated above. A wide experience in functional and technical architecture change management is necessary and must be associated with an adapted methodology. Logica, an international consulting company is an historical member of the Sustainable IT Community and of the MAG (MDM Alliance Group). Their many real experiences conducted in IT systems overhauling enrich the capacity to establish the vision described in this paper, in particular when speaking about IT transformation governance and methodology around BRMS and MDM system.

### What SOA brings to support the IS transformation

Service Oriented Architecture is used all along the IS transformation process. Ever since the first step "Act on legacy systems" SOA style is mandatory to urbanize flows between repositories (MDM, BRMS) and other components of the information system. This is the first maturity level called "Cosmetic SOA" with ACMS because it is not yet intrusive for legacy systems.

Then during the overhaul, SOA is used to design a service oriented logical architecture able to help build services reusable in multiple business processes. This is called "Overhaul SOA" with ACMS. Reusability of the services will be increased by the flexibility brought by BRMS and MDM repositories. In fact, services will exploit reference data and other parameters (MDM) and business rules (BRMS). The opacity flaw found in hard-coded services is fixed and replaced by the increased readability of MDM and BRMS repositories.

Orchestra Networks and ILOG companies bring enterprise level software solutions respectively for model driven MDM and BRMS which are able to manage all kind of enterprise data and rules in both specific developments and commercial software deployments contexts. Business governance function are user-friendly oriented: versions and rights management, auditability, UI management: these features enforce business teams involvement and are the guarantee that the fundamental IS assets, business rules and reference data, won't be trapped inside IT languages only understandable by IT people.

## References

Sustainable IT Architecture (S-IT-A) <http://www.sustainableitarchitecture.com/home>

MDM Alliance Community (MAG) <http://www.mdmalliancegroup.com/>

S-IT-A community's book: "Sustainable IT Architecture" by Pierre Bonnet, Jean-Michel Detavernier, Dominique Vauquier - WILEY publisher.