Connect, Navigate, Transform: Unlocking Defence Architecture Data

Yann LEBEAUPIN – SODIUS – Chief Technical Officer
SODIUS, 14 Boulevard du Maréchal A. Juin – 44100 NANTES
FRANCE

ylebeaupin@sodius.com

ABSTRACT

Organizations are building ever more complex systems that generate more and more data with ever increasing complexity. Sharing and collaborating effectively on this data between stakeholders (among country, project, team and individual) is not easily accomplished due to the lack of communication between tools. In practice, a single toolset is not possible. However, organizations need to share or exchange data with partners on a joint project even if teams have heterogeneous authoring tool choices (historical, national, or business). Consolidated information in a federated common view is required to enhance visibility and unlock engineering data from their native authoring formats.

This paper describes some of the results and lesson learned of our own R&D efforts and experimental and research activities done with the DGA, implemented as a flexible Defence Architecture (NAF) collaborative portal.

1.0 INTRODUCTION

1.1 Problem Statement

Usually data authored and managed during the lifecycle of systems projects is not open or accessible outside their native spaces. Tool selection, driven by historical, business and national reasons, has resulted in data only being available to users of the original authoring tool. This situation profoundly limits the accessibility of information to the other stakeholders of the system. This is a problem especially in the domain of system of systems and for trans-national applications: closed environments do not allow analysis relying on common concepts and relationships because they are managed by different tools.

These gaps are evident - organizations need to share or “exchange” data with partners for review or consolidation activities. Typical responses involve many manual steps, intermediate document production, and other generation activities that are non-value adding. While a single repository to manage all data would alleviate many issues, it is unrealistic as it implies a single vendor adoption which is impractical. The proposed solution is a collaborative portal to enable transformation, linking and interoperability using a federated approach for multi tool environments. The sharing and exchange of data is achieved by models, which are adapted to common concepts using dedicated interfaces.

This paper presents some of the results of building a collaborative portal, applied to NAF architecture exchange and review, enabling the sharing of Defence Architecture (NAF*)* engineering data. We are experimenting it with the DGA, the French Government Defence procurement agency.

1.2 Challenge

In many systems or systems of systems projects, models become the masters, more than documents. Models have become the critical focus in handling increased complexity. These models have many uses including: capturing behaviours, structures or interfaces, as well as supporting tracking and verification of requirements. In recent years, the inclusion of model-based assets in the design process has become standard practice rather than the exception. While we can say that models are everywhere, we can, and need, to be more precise. In many projects, we are dealing with various model-based environments and "design" cultures: NAF or UPDM2 are used for operational scenarios and high-level descriptions of programs and services, SysML or UML are used on the lower levels. We include requirements as in scope since they have the same nature as models, design elements and relationships, however the operational practice is still vendor and document based. While standards such as RIF/ReqIF exist, the dominant formats still remain IBM Rational DOORS® and Microsoft Word®.

The real challenge becomes not only to answer "how do you connect data" but "how do you enable collaboration through various multiple and non-homogenous model-based systems engineering environments?" As an experimental development project, our goal was to build a flexible and pragmatic solution that could help to answer this question. To tackle the problem of tooling integrations, we have designed a dedicated solution called MDWorkbench, in production use for over a decade, to rapidly create and deploy integrations.

With this platform, one of the goals is to get a concrete tooling to replace locked document outputs to flexible model-based exchange – including semantic and diagrams.

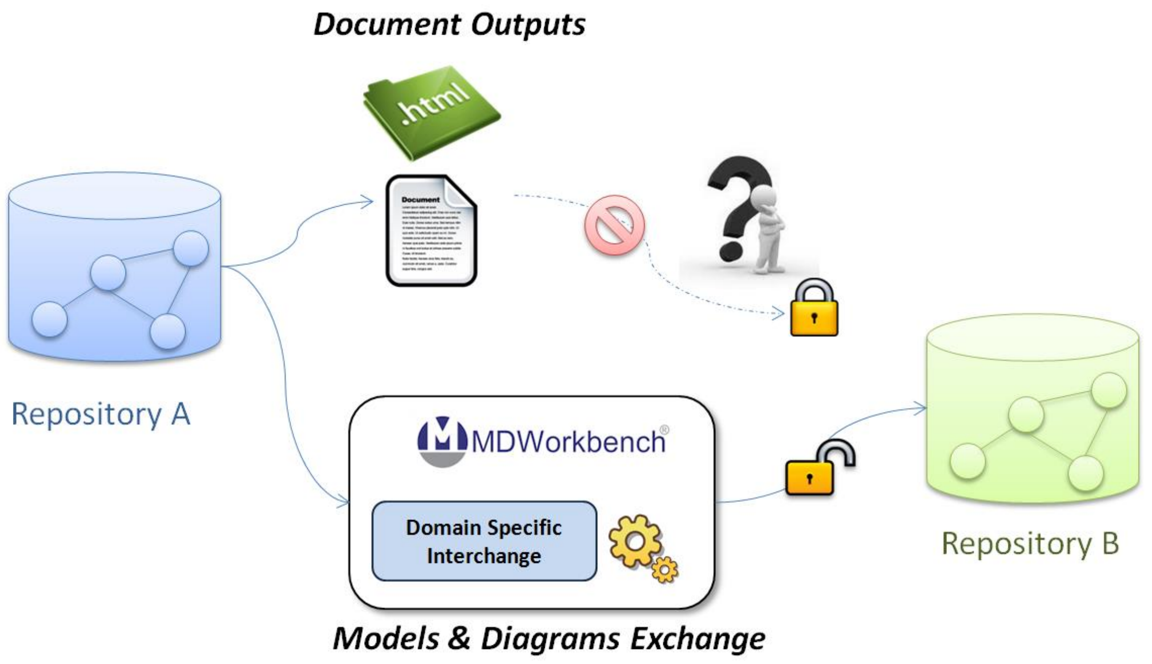

Figure 1: From documents locked outputs to NAF model exchange

Prefer a "component-based" architecture to a monolithic solution to guarantee flexibility, scalability and agility for our future projects,

- Adopt a model-based approach and standards-based notation i.e. NAF or UPDM2 to navigate/request/link/trace data across disparate architectures using a common vocabulary,
- Provide a User Interface to make a federated architecture accessible to end users,
- Minimize impacts on infrastructure by "plugging" our components into existing repositories instead of reinventing a new data management system.

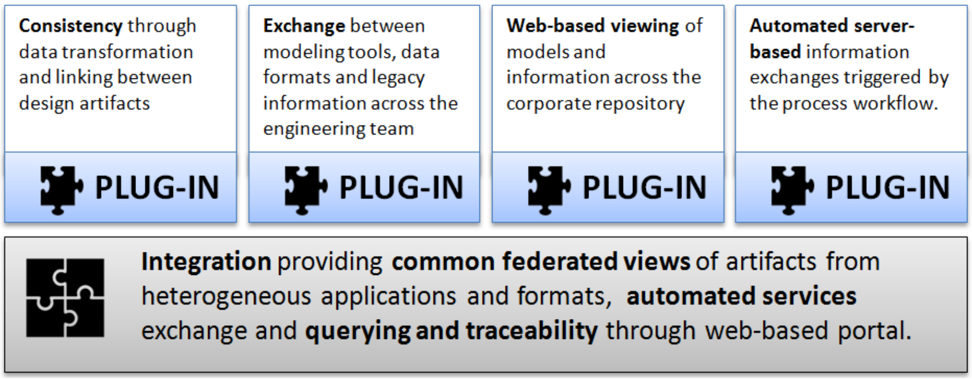

Figure 2: Core Components of the multiple-tool federated portal

The core of the prototype is to improve the integration of tooling to enable user collaboration on technical information. To achieve this organizational level of interoperability, we identified three critical capabilities. The ability to *connect* to all data sources, the ability to *navigate* and explore the contents of the data sources, and the ability to *transform* one data source to new formats and repositories. We further refine these objectives as follows:

Connect Capability

- Give visibility across many sources of data, including both semantics, links AND diagrams
- Support linked data to remote repositories (Requirement Management and Change Management), based on efforts done through the OSLC approach and Jazz-based infrastructure.

Capability

- Organize them in a common understandable way – a simple Engineering Configuration Structure (Workspace, Model Asset, Views, Elements and Relationships) – and support tool-agnostic common aspect – NAF 3.1 for the DGA prototype.
- Use data connected to the portal to support analysis functions – collaborative annotation threads for reviews and querying of models.

Transform Capability

- Provide automated server-hosted functions with a large range of conversion services: import/export/transform many data formats and generate Word/PDF/HTML reports on the linked data.

With the evolution of the landscape of tools, tool architectures, and tool repositories we are also driven to explore the benefits of deployment options. While traditional methods have deployed these tools in an intranet with very exclusive user bases, the alternative is cloud based solutions that are more inclusive. Since collaboration is a key motivator, the ability to deploy and provide services via a cloud hosting solution demands critical exploration.

2.0 PATH TO A FEDERATED NAF PORTAL

Of course, the path to get a federated vision and exchange capability over multiple engineering environments has no unique answer. The topics presented in this paper are not definitive but they provide some of the key points we had to deal with while developing the solution: support proprietary formats or APIs, interpret them in a common vocabulary, and consolidate interpreted data to reach the federated architecture visibility target.

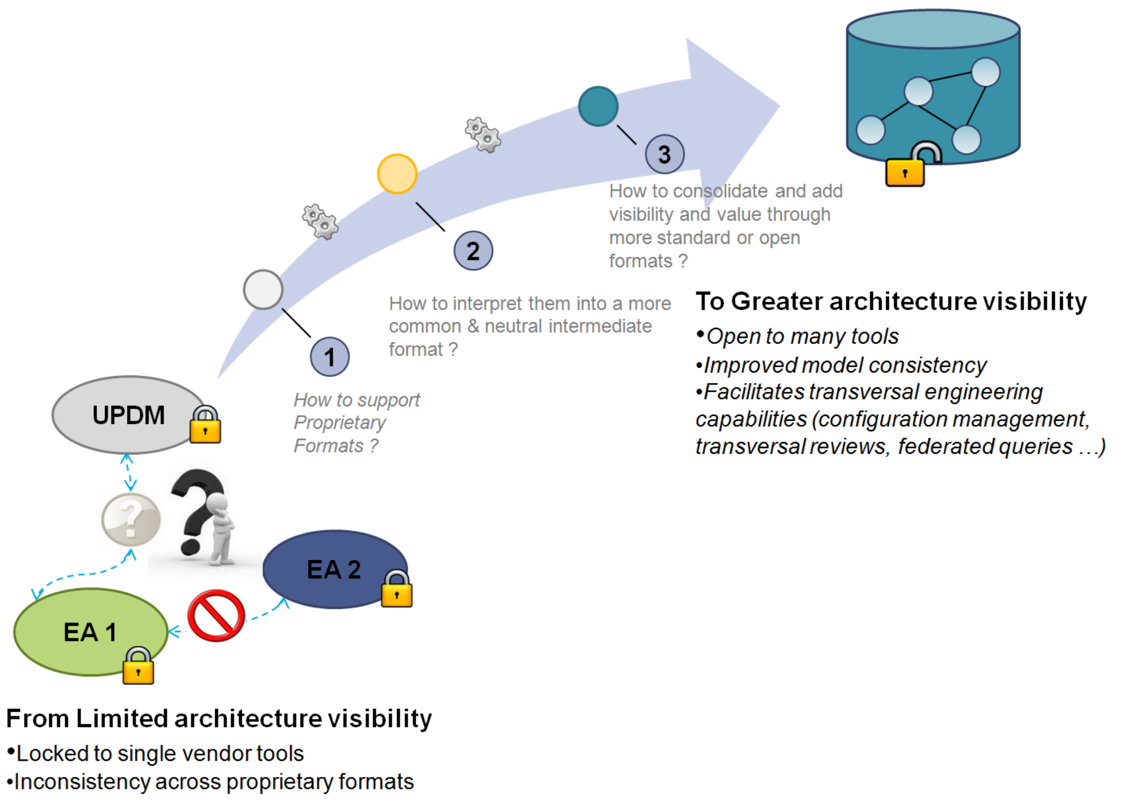

Figure 3: The path to a greater architecture visibility on the data side

2.1 Data Connectivity

The data connectivity layer is based on different components, delivered as a technological platform called MDWorkbench, with accessors providing complete read/write interfaces (including diagrams for modeling tools) to authoring tools and automated rules to mediate the different concepts between tools. Note that all our components can be united together, providing n:m mapping and exchange capabilities.

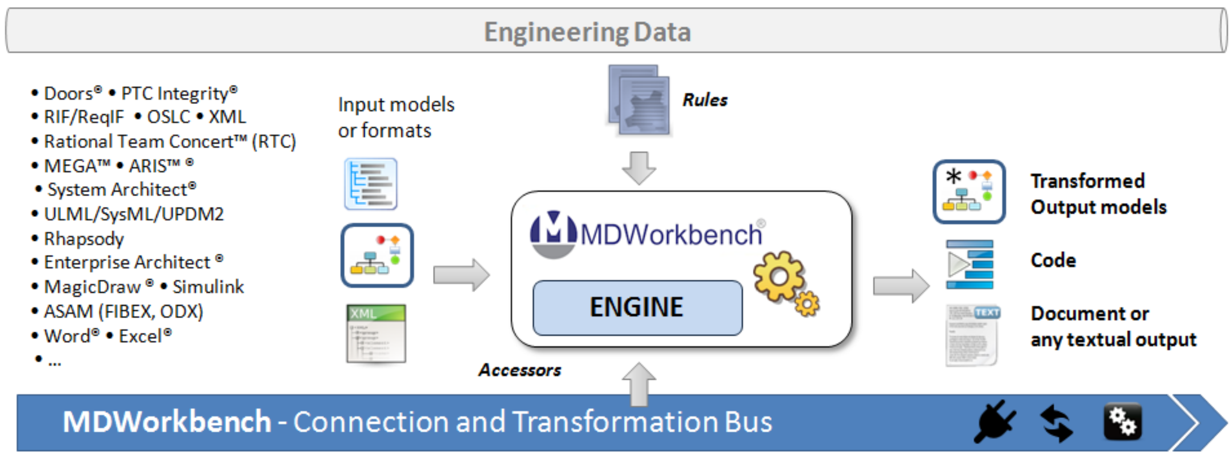

Figure 4: The MDWorkbench platform providing data connectivity and transformation capability

How do we connect data?

MDWorkbench enables the creation of new tooling connectors and encapsulation complexity of direct connection to authored data, realizing the vision of an adaptable model hub architecture. Our approach is based on MOF meta-models (using the EMF layer of Eclipse) and dedicated accessors supporting read/write operations. This architecture allows us to expose a "semantic" view of the connected data instead of dealing with the raw data syntax. As a software vendor, we handle the maintenance of those connectors, managing the changes between versions and minimizing the impact on their integration. Users of our connectors build integrations based on the semantic meta-model rather than the underlying syntax, enabling greater longevity to their integrations. For many years these capabilities have been strictly available as client-side applications from authoring tools, however this is now evolving. As engineering tools have become more server/repository centric, the MDWorkbench connectivity layer has evolved to "server" side assets, and RESTful services to execute them. For example, the evolving OSLC standard is a common connector for the technology enabling advanced linking capabilities.

How do we achieve transformation?
The SODIUS team has a long-dated experience with semantic transformations. We have chosen an imperative approach (based on Java layers, powerful connectors and utility frameworks) preferred to other declarative approaches. Having implemented such transformation for very large sets of data (several hundred thousands of objects, thousands of diagrams), we have clearly understood it was difficult to transform huge graph models using only declarative means. We have also identified most declarative tools have included "imperative" capabilities and at the end rely on them when you want to transform between concrete complex domains and not only map "my class" to "my other entity" topics. Encapsulation of the transformation complexity is done through imperative rules organized as exchange services.

What is the level of data exchanged?
As explained previously, we have built a specific set of connectors to the engineering tools to be able to manipulate 100% of internal data. It means we have the capability to transfer the whole scope of data AND diagrams for most of tools (depending on the availability of APIs or formats). Once this first step of data accessibility is done, the next step in the process implies mapping the different concepts between tools using transformation rules and at the end to make them "compatible".

How do we map data between tools?
On the standardization side, if the adoption of universal metadata has been partially unsuccessful those last years, it was partially caused by conflicting priorities between vendors rather than because of irresolvable issues. We have built the solution to practically achieve the vision delivering bridges that meets semantic intent over the pure syntactic intent that the individual vendors have delivered. Additionally we have developed patterns (including pivot approach) to make these quickly and more valuable to n:m type bridges. We have worked to reuse the outputs of modelling standardization efforts by adding to them the missing level of flexibility and consistency required to support data heterogeneity (per tool vendor, per country, stakeholder, project, team or individual). Depending on the level of reusability expected and the number of tools involved on a bridge, we can mix direct exchange between meta-models (point to point mapping) or a pivot approach.

How to deal with the common language (capture, define, impose)?
For the NAF bridge, we have built a tool chain to extract concepts from specifications and developed dedicated sets of rules to map each connected tool to the common vocabulary. In our federated approach, we do not impose constraints on the connected data. However, only reconcilable concepts (where a "sameAs" relation exists with the common vocabulary) can be considered as interfaced at the end.

2.2 Standard Evolution Management

Another requirement of the solution is the capability to "survive" the evolution of standards. Today, there are many architectural frameworks. For formalisms, we manage various data coming from NAF (NATO) DoDAF (U.S.), MODAF (U.K), UPDM (OMG) and many others. There are initiatives like IDEAS on the foundation side and MODEM that could clarify interoperability of data in the future (to see how to leverage those efforts at the model or architectures side).We are already aware that the current exchange based on NAF 3.1 will have to change in the future. So we have decided to get a flexible structure that could be adapted to many domains and their distinct evolutions (currently defence, but can be applied to any domain as automotive, aeronautic or others). We can summarize our initial requirements as the following:

- **Integrate variability of framework** implementation between Enterprise Architecture/UML Modeling tools targeting first NAF3 but **extendable to UPDM2** with adaptations
- Manage **Data AND Diagrams**
- Use a **pivot approach (Common Data Model)** to avoid point-to-point connections between authoring tools **(tool agnostic format)**
- Build an **extensible solution** to be able to support custom extensions, new tools and future version of frameworks

The goal is not to deliver a solution for the next year but the visibility of the solution in time to be able to maintain over years, including changes of tools and standards. In parallel to the application design itself, we worked to create a productivity tool chain to derive and produce most of the implementation directly from the standards specification as UML profiles or other model-based assets.

On the linking path and **o**n the industrial point of view, we developed components as our Conversion Server that are deployable and can interact with OSLC (Open Service Lifecycle Collaboration) layers. This project provides useful functions for both industrial and national defence stakeholders. We already have supported other features such as internationalization, connected/disconnected mode to allow HTTPS or file-based transfers, different roles and access rights.

Internal data representation has been realized in the context of this collaborative portal using W3C Web Semantic technologies such as Resource Description Framework (RDF) or SPARQL (RDF query language,) for queries.

2.2.1 Example of data exchange

Using this approach, we developed several solutions dealing with the exchange of NAF and UPDM2 data and diagrams, with the complete set of standard views. On the NAF side, we have created a complete bridge for System Architect and MEGA including diagrams. This exchange service is used to exchange system designs between teams. Additionally, we have enabled partial but representative exchange services to UPDM2 (Enterprise Architect for example). One of the benefits of the pivot approach is that any new incoming format, once plugged into the core common environment, will be available for all existing integrations.

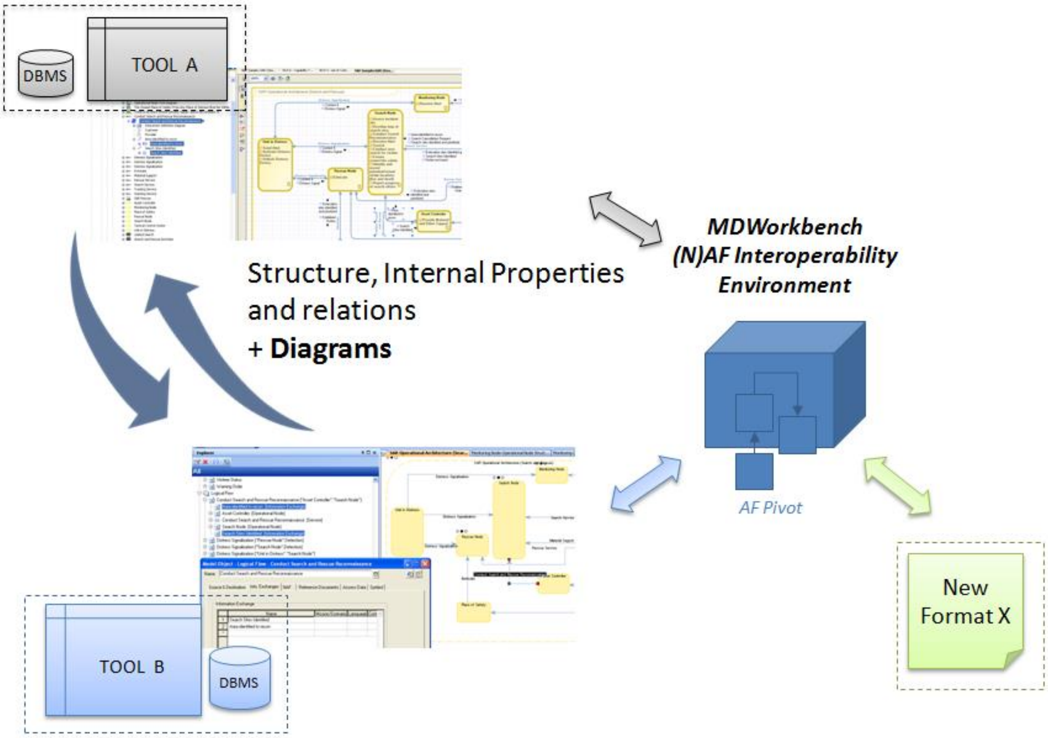

Figure 5: Example of NOV-2 Exchange between System Architect and MEGA and Extensibility

2.3 Flexible Model Hub Vision

While the data is federated there needs to be a hub that is the entry point. It enables the accessibility of the information semantically correct but in the form the stakeholders need both for review and further work.

2.3.1 Flexible Data Management Structure

To integrate and enable a more flexible interaction with our portal, we have defined a structure flexible to connect and organize the raw data in our NAF portal prototype: a collaborative workspace composed of connected models. Typically, a model is composed of Elements, Relationships and Diagrams.

We are accessing and processing this raw data in 2 ways:

- one is on its population into the hub, for a design review for example, because we are reviewing a stabilized version of assets
- the second is the processing of raw data on demand. The use cases are on demand indexing and categorization, linking using OSLC layers, or generating output for a user's needed tool format.

This is the first level of flexibility, synchronous or asynchronous access to data.

Data is exported from authoring tools using the built-in file exports or custom add-on applications we have developed to facilitate connection to the model hub server.

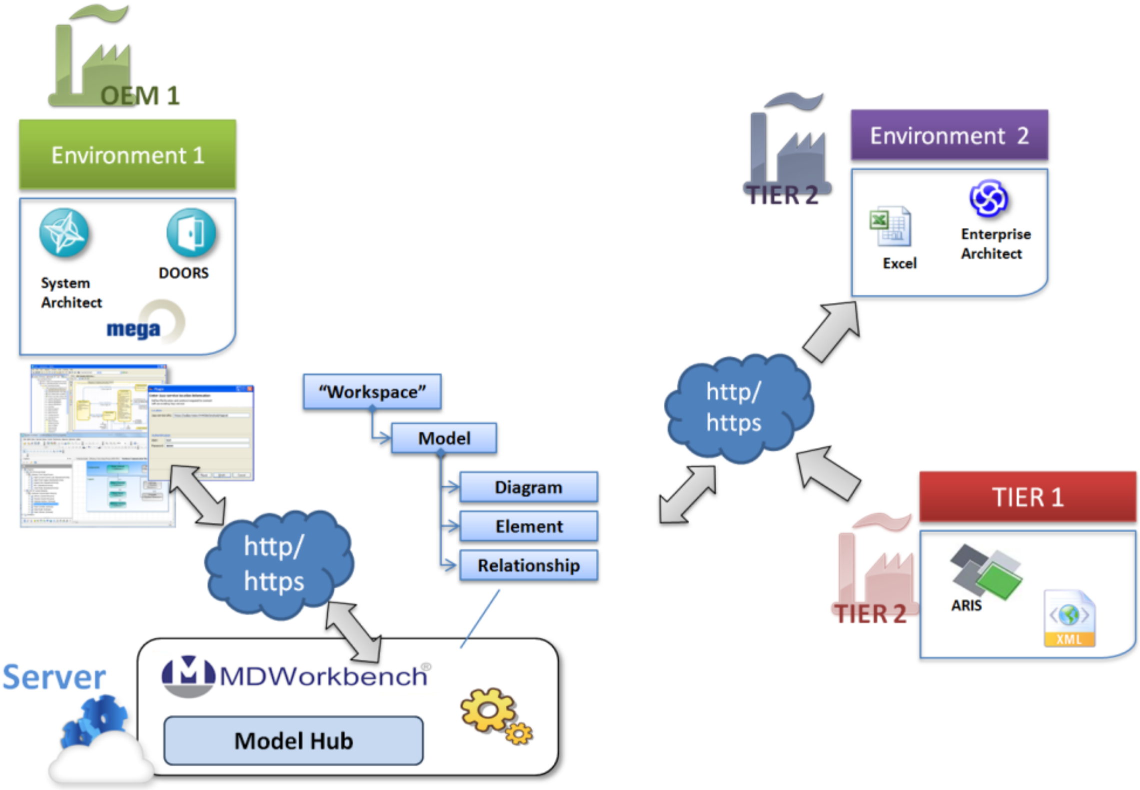

Figure 6: Exchange Flows Overview between different stakeholders

Additionally to those "raw" inputs, we add the desired domain specific view and categorization by plugging a classification system defining mapping of types and domain queries. We provide the presentation of common classifications through the hub with specific adaptor rules. Conversion of project data from multiple different tools into a common presentation eases common collaboration.

To get a second point of flexibility, our classification is not a specific indexing activity but on-demand. We have distinguished semantic adaptors that allow different classifications for a same model and conversion services that deal with the complete transformation of data between two tools. It avoids introducing a conversion on the connectivity or storage layers that would only make available the "minimal" common vocabulary.

The third point of flexibility deals with the technical dependencies of the solution. We have tried to implement "target platform" agnostic components and frameworks. The benefits of being as independent of one and only one specific data management solution is that we can deploy our technical components on standard web containers (as Tomcat) or more advanced data management systems as ALM platforms (as Jazz) or PLM platforms (as Windchill) existing in the stakeholders infrastructure without introducing a completely new (and redundant) persistence system. We focus on our added-value: fluidity of data exchange and multi-tool environments mediation.

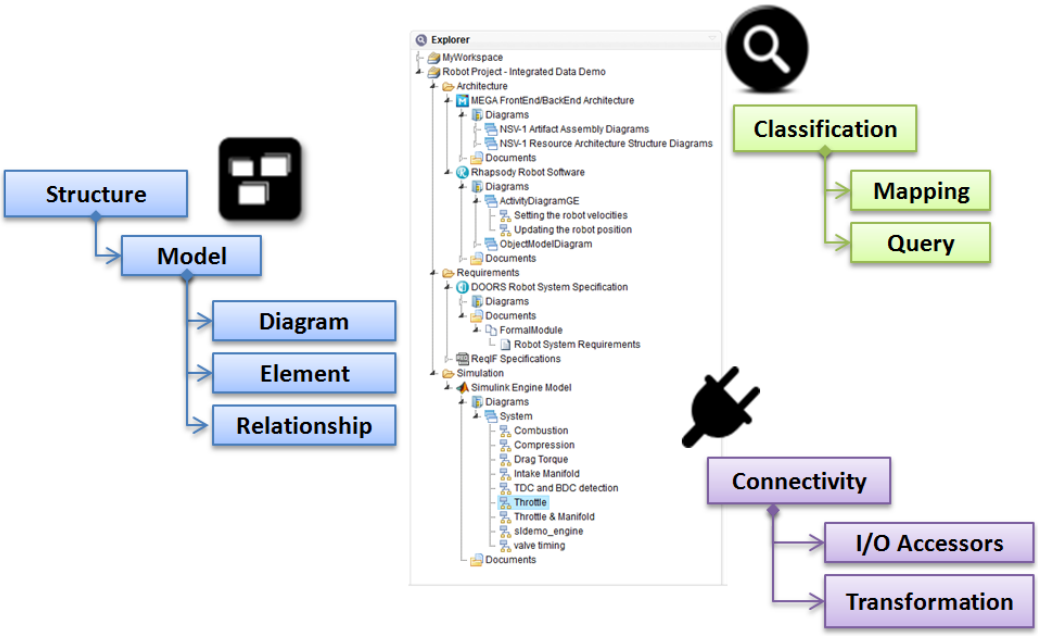

Figure 7: Flexible layers of model structure, classification and connectivity

2.3.2 Federate repositories instead of centralized

Our "federated" approach should not replace the initial constraint problem of the "unique" engineering toolset by another one at the repository or infrastructure level forcing adoption of a unique portal/storage solution for example. For this reason, our model hub solution is mainly based on a set of individual components that we can aggregate or not into a final portal. As explained before, we can deploy them on various target platforms.

Using a delegation principle for authentication and storage, we aim to address various existing configurations between the stakeholders having to collaborate. To ensure we remained non-dogmatic in our approach, we identified three critical pillars for success in addressing critical usage scenarios:

- First, focus our efforts on our added-value on advanced connectivity and transformation capabilities, delegating engineering configuration topics to existing solution of tools or data management platforms.
- Second, complete engineering data scope access and transformation capability to be able to publish, transform and restore engineering data to a common "transient" publication place,
- Third, be able to link data using Web semantic standards and the scalability of the Web architecture keeping the data in their own systems.

Those pillars have allowed us to create a federated sharing workspace, offering not only linkage capability but advanced exchange and conversion features. The concrete results consist in a NAF portal that is able to manage the publication from different Enterprise Architecture tools (MEGA®, System Architect®, ARIS®), UML design tools (Rhapsody®, UPDM Enterprise Architect® and MagicDraw®) and providing specific exchange capabilities (MEGA® to System Architect®, MEGA® to EA® UPDM).

The portal workspace is extensible beyond architectural models. This is demonstrated with the support of requirements management data linked with NAF data. Similar to the support of multiple sources and targets of NAF data, the Requirements data supports common representations including OSLC RM, DOORS®, PTC Integrity®, and ReqIF.

3.0 RESULTS

Our model-based solutions have allowed us to prototype a collaborative portal supporting collaboration use cases under the various infrastructures and domains. To validate the approach, we are experimenting with several deployment scenarios in 2013: our complete integrated NAF solution with DGA under the Jazz platform interacting with RTC, model web viewer components in interaction with Windchill PLM and exchange services with custom PHP and Tomcat-based portals in the automotive industry.

The next diagram represents one of our target logical architecture.

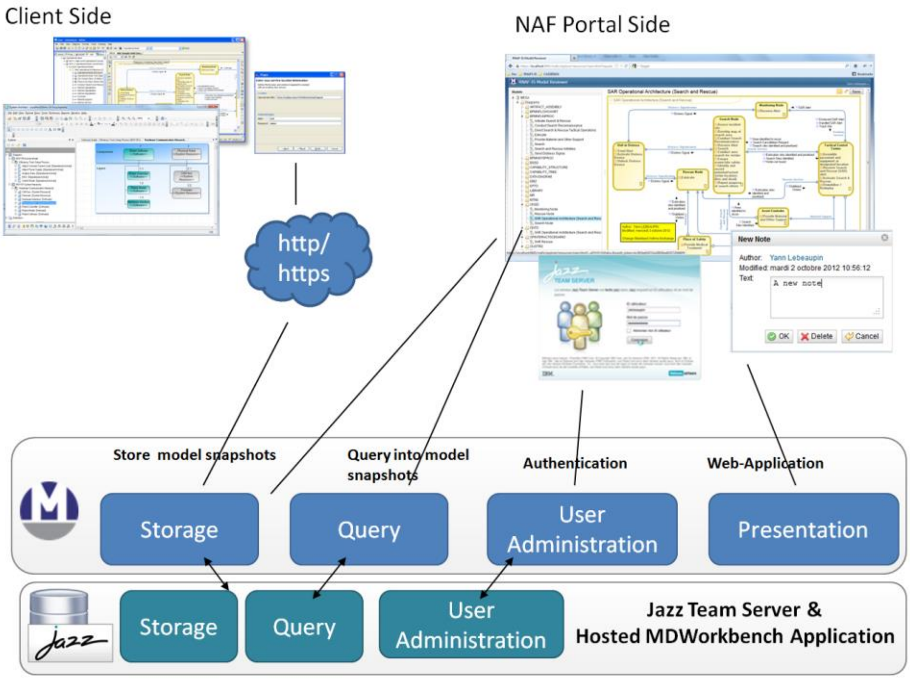

Figure 8: Example of Jazz based deployment

It supports critical collaborative services including:

- **Model Hub: Create a central multi-tool connectivity space**
  - Create a multi-tool workspace and establish a unified system of connecting engineering tools to ensure accuracy, consistency and accessibility of all your data and discussions around model items.
  - Flexible architecture to be able to connect with different Enterprise repositories solutions for persistency
- **M2M Services:** Exchange and synchronization of engineering data through model-based solutions
  - Deploy exchange services to enable advanced import/export/update scenarios between stakeholders' workbenches
  - Allow multi-tool document generation from the shared workspace

- **Workspace-Model Explorer**
  - The workspace tool explorer gives you the capability to navigate and display the connected artifacts and relationships.
- **Classification**
  - Our layer is based on the “decorator” aspects enabling the activation of multiple point of views on the connected data
  - This feature enables extensibility to various aspects (NAF/DoDAF, SysML, etc)
  - Extensibility criteria to be able to deploy it as non-defence domains
- **Search and Query Extensibility**
  - Extensibility is required to build further step-by-step standards integration (as MODEM) and manage unanticipated semantic queries for example
- **Multi-Tool Dashboards: Assemble data and requests to provide multi-tool metrics and portal activity feedback**
  - Make it easy to assemble model-level views of design activity and data using widgets within the dashboard.
  - Trend Commitment Report: This particular dashboard allows to quickly follow publication, review and synchronization events for users
  - Get feedback with everyone using multi-tool Reviews: using the review capability, a member of the project team can send a specific set of comments out to the team and stakeholders to review and provide feedback in-line
- **Traceability: Understand the ripple effect of a change.**
  - Track the relationships between every published artifact from heterogeneous models
- **Internationalization**
  - To support trans-national mode
- **Off-Line Mode**
  - This mode allows “off-line” reviewing of some previously imported data.

4.0 CONCLUSION AND FUTURE WORK

We have demonstrated through this prototype the technical feasibility to provide connection, navigation, and transformation capabilities between heterogeneous design repositories minimizing impacts on existing infrastructure.

In the Defence domain, we have demonstrated the operational exchange capability of Enterprise Architecture or UML-based models and diagrams, enabling common NAF point of views. We expect that our technical components and systems engineering expertise could help many organizations to move from their silo syndrome, in which barriers limiting sharing and visibility are in place, to help improve decision-making capability. Of course, this has to be to be done in parallel with adoption of new processes and introduction of a collaborative culture.

As we have gone in detail in the connectivity and transformation side, we know that other aspects critical to long term success include configuration and asset management. As an example of the extended challenges, it is understood that management of transversal traceability links at a very fine grain level is not a simple topic. The next step will be to explore engineering configuration controls to provide first level interfaces to handle set of objects for each of the connected tools.

As the prototype was created for Defence application, we have started other works in 2013 to give other industrial “flavours” introducing ReqIF, SysML and Automotive standards to further validate the results. We are considering other research and development: one will be to standardize the management of intermediate and federated meta-models or ontologies. Support of RDF is now complete therefore the next steps will focus on ontology tooling to prepare the compatibility with potential IDEAS activities.

As an output of the experimentation phase, we have built our own test models inputs, gathering data from many authoring tools to identify and resolve interchange issues associated with specifications. But it should exist official test models for the future defence architecture framework perspective (as SysML efforts for example) to share really "unified" testing material. These kinds of common "validation" assets (and programmatic means of checking results between tools) should be provided or built to enable defence AF format standardization.

On the product deployment side, we intend moreover to go further to push our solutions to cloud environments to provide more flexibility in the delivery and accessibility to a model hub. This is especially important in mixed organizational activities where the connection to multiple systems is required. We see this as critical capability growth to collaboration even if we do understand that there are additional challenges in such environments, including items such as authentication, role based rights, and data visibility.