D1.1 Goal and metrics document

v1.0 Draft1

	NAME	PARTNER
WRITTEN BY	Y.BERNARD	Airbus Operations SAS
REVIEWED BY	F.RACCARU	AKKA Technologies

RECORD OF REVISIONS

ISSUE	DATE	EFFECT ON PAGE	EFFECT ON PARA	REASONS FOR REVISION
01	31/05/2010			Document creation

TABLE OF CONTENTS

TABLE OF APPLICABLE DOCUMENTS

N°	TITLE	REFERENCE	ISSUE	DATE	SOURCE	
					SIGLUM	NAME
A1						
A2						
A3						
A4						

TABLE OF REFERENCED DOCUMENTS

N°	TITLE	REFERENCE	ISSUE
R1	Galaxy glossary	D.1.2.2	
R2	F2-4 : Analyse des exigences pour le passage à l'échelle	Lambda F2-4	v1.3
R3			
R4			

ACRONYMS AND DEFINITIONS

Except if explicitly stated otherwise the definition of all terms and acronyms provided in [R1] is applicable in this document. If any, additional and/or specific definitions applicable only in this document are listed in the two tables below.

Acronyms

ACRONYM	DESCRIPTION

Definitions

TERMS	DESCRIPTION
Extended enterprise	a loosely coupled, self-organizing network of firms that combine their economic output to provide products and services offerings to the market. Firms in the extended enterprise may operate independently, for example, through market mechanisms, or cooperatively through agreements and contracts. (Wikipedia)

1. INTRODUCTION

The goal of the Galaxy project is to work on the technical hard points related to the fragmentation and to the distributiveness of huge models, and to their synchronization in regards of the communication means classically used by development teams. As a starting point, the challenges linked to the industrialization of the MDE technologies have been studied in the frame of the Lambda FUI project and the results are available in [R2].

The work that will be done will be driven by use cases provided by an industrial (Airbus) that will already have characterized and described scalability issues face during systems developments. This document is produced in the context of the work package 1 that aims to provide the definition of quantified objectives for the technologies developed in the frame of Galaxy.

1.1 GOAL OF THIS DOCUMENT

This document refines the scope of the Galaxy project by specifying both its qualitative and quantitative objectives. These objectives are defined based on two sources of information:

- The description of the scalability challenges and of the metrics provided by the Lambda project in [R2],
- Airbus realistic projections about what will be the requirements of the next generation of avionics systems

1.2 DOCUMENT ORGANIZATION

Chapter 2 specifies industrial objectives for the MDE technologies, so that they will be able to support the development of avionics systems we will have to develop in the next twenty years. The requirements on these technologies are driven by two main axes: complexity of the specifications and complexity of the industrial organization.

Chapter 3 analyses the factors identified by [R2] as potential source of scalability issues against the frame defined by chapter 2 to decide whether and why they are relevant or not, to specify which one Galaxy will work on and with what quantitative objectives (where relevant).

Chapter 4 gives the conclusion.

2. INDUSTRIAL CONTEXT AND CHALLENGES

2.1 IMPACT OF SYSTEM COMPLEXITY

The more obvious, no way around consequence of the increase of system complexity is to raise the number and size of elements to manage. The first factor of this growth is the progress made by the integration technologies allowing putting always more components in the same unit of volume. Over this design aspect, the great interest of models is their ability to be used by test activities for early validation. Test activities typically provide results under the form of execution traces. By formalizing those trace into models, we get all power of model engineering technologies to exploit them but then we face the problem of potential infinite model management.

Complex systems involve model artifacts from various natures and kinds, often expressed using several distinct formalisms. This results in numerous relationships between those artifacts, which need to be reified and becomes themselves additional model artifacts, some at the crossroad of two paradigms and/or tools using distinct technologies. This is a two level issue.

The first one is semantic, linked to the nature of the item to the laws that bind their properties together. Thus, it has to do with the concepts of “view & viewpoint” and then with the synchronization of those views. The development of an avionics product involves several disciplines from several domains. Each of them uses its own set of views. All those views conform to viewpoints that can be specific to a domain or a discipline but they are related since they have the same product for subject.

The second one is syntactic and linked to the technology used to implement those views or even miscellaneous part of a view. Assuming than the first level is resolved, the problem it raised is more practical than theoretical and resides in the amount of data manipulation required for the synchronization.

In this domain, one of the challenges Airbus is very interesting is the ability to integrate in a single well-formed and consistent model an architecture description using an asynchronous language (e.g. SysML) and the specification of control/command laws using a synchronous formalism like SCADE. In addition to the design of its software and electronics components, the development of an embedded computer may also require analyses from other viewpoints like: dependability, Electromagnetic Compatibility (EMC), thermal energy, etc…

With the hardening of the commercial competition there is a high pressure on the industry to make always more effective systems for an always tighter cost. It implies to improve the productivity as much as possible. That is the purpose of lean engineering processes. One of the main principle of the lean engineering is to do only what is necessary and thus, to avoid doing the same thing twice. That is applicable to modeling also: capitalization and reuse of modeling works are key points. The ability to manage efficiently and to share model libraries between projects is on the critical path to success.

The models that those libraries will provide must be seen as share components that will have to evolve over the time for miscellaneous reasons, either intrinsic (e.g. specification changes, corrections, …) or extrinsic (e.g. modeling language or supporting tool evolution). Because those evolutions may impact the way shared components are integrated, it is mandatory to manage their versions and to be able to configure each model of product to refer to given and registered versions of those shared components.

2.2 IMPACT OF INDUSTRIAL ORGANIZATIONS

No industrial product is developed by only one person on a single workstation. Model driven development implies the collaboration of numerous people, possibly located in distant geographical sites, sharing models and parts of thereof. Most of the management methods applied on industrial development projects required that a certain degree of desynchronization (i.e. inconsistency) can be supported during a given time between parts of a model or between copies of one those parts. The point is to be able to build on demand a full consistent model for all those parts. Mechanisms and tools able to provide that kind of services classically rely on both a repository and a configuration management tool. The repository provides a shared space to store and provide the official version of the miscellaneous parts. The configuration management system registers the version and generates consistent builds.

More and more companies are organized according to the “Extended enterprise” pattern. Those organizations are networks of loosely coupled companies that group and work together to provide products and/or services. If each member of those networks must agree with other member on what data and/or material they exchange, the tools and the technologies all of them use internally to provide their added value in the production process remain their own choices. Nevertheless this can work only if the technology that supports the exchanges: (1) is robust to this diversity and (2) is scalable to deal with the volumes involved in an acceptable timeframe.

3. GALAXY SCALABILITY OBJECTIVES

3.1 ABOUT "SCALABILITY"

According to its common acceptation, a system is said "scalable" if the amount of power it requires remains proportional to the amount of work it has to achieve when this amount growths.

Transposed to the MDE domain, we can say that a technology will be *scalable* if it can keep linear the relationship between:

- each factor depending on complexity of the developed product, on the one hand
- and the amount of resources this technology requires to work, on the other hand.

Most of current MDE technologies use software application running on digital computers to manage model and model transformations. Those applications consume the three kinds of resources a digital computer provide in a limited manner, depending of what is commonly called its "power":

- storage (memory)
- computing (processor)
- communication bandwidth (bus)

Analyzing how the complexity may impact the scalability of a MDE technology means understanding how the miscellaneous dimensions of the complexity drive the amount required for each of those resources.

Human processing shall also be considered. While some manual processing are acceptable for small projects, they can become too much error prone and/or time consuming when the size of the project growths.

The industrial objective of Galaxy is to provide solution so that the MDE technologies we plan to use will remain scalable in the usage domain corresponding to the development of the embedded system we forecast for the next 20 years.

3.2 AMOUNT OF DATA

3.2.1 Number of artifacts

The number of artifacts to manage is the most obvious factor linked to the complexity. Since it is directly related to the consumption of the three kinds of resource, it is also one of the main ones for the scalability.

Two kinds of artifacts must be considered: "entity kind" artifacts and "link kind" artifacts, which describe the relationships between the entities. The complexity may impact the number of artifacts on two axes: the number of entities and the number of links per entity. That impact can be either linear or non linear for each axe but if both axes are impacted the relationship with the global number of artifact will surely be non-linear. If so, scalability would be far more difficult to achieve.

Based on the trend computed thanks to the Airbus' programs history and Airbus expert forecasts, we can get a reasonable estimate about the amount of data that will be necessary to model the next generation of avionics computers, which are among the most complex software intensive embedded systems. The analysis has to consider both the hardware and the software domain.

Regarding the hardware part, Figure 1 points out several interesting points. First we can see that, in the considered period the total number of electronic element evolves according to a slightly "S" shaped curve. Its inflexion point is the consequence of the technological jump realized by the A380 program. We can reasonably consider it is linear.

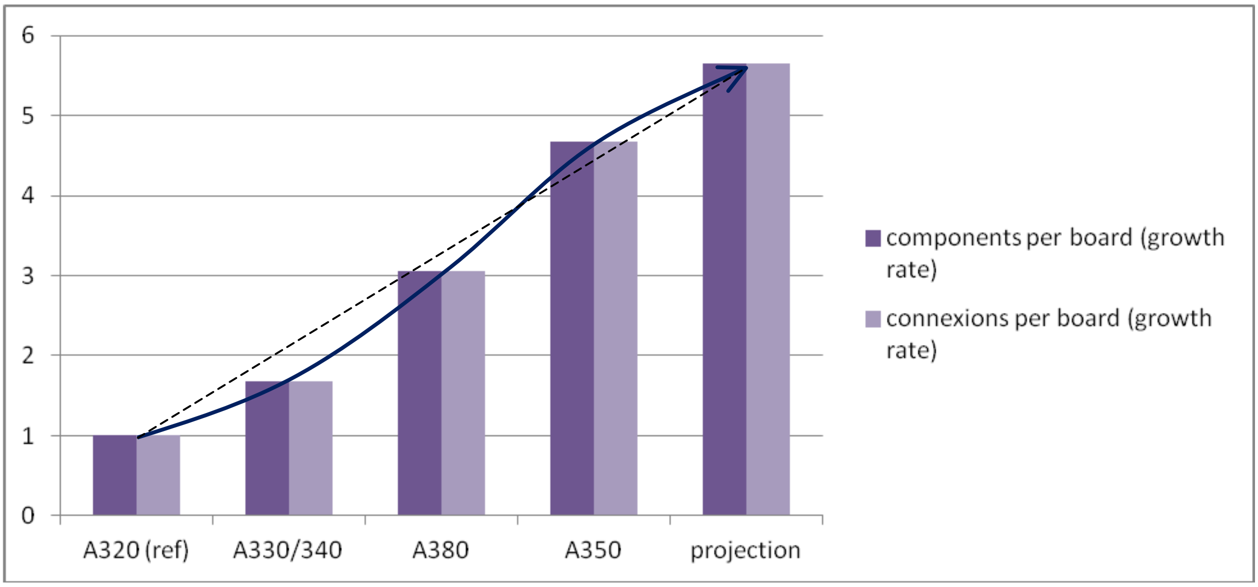

Figure 1: Growth rates of the number of artifacts per electronic board of avionics computer vs Airbus aircraft programs

Another interesting point is that, as the number of components (entities) growths, the number of connections (links) per component remains very stable and greatly participates to keep global the growth curve in the linear domain.

Airbus projections for the next generation of avionics computers are:

- 1400 components per board
- 5500 connections per board
- Up to 3 complex component per board described by 100000 lines of HDL code

The inflexion of the "S" curves but more pronounced for the software part, as shown in Figure 2. Two kind of avionics computer are considered. Code embedded in Flight Control Computers (FCC) seems to have reached a plateau around 1.4 millions of lines including both manual and automatically generated code. For Flight Warning Computers (FWC) the plateau is not reach yet and the volume might reach 3 millions of lines in the next generation.

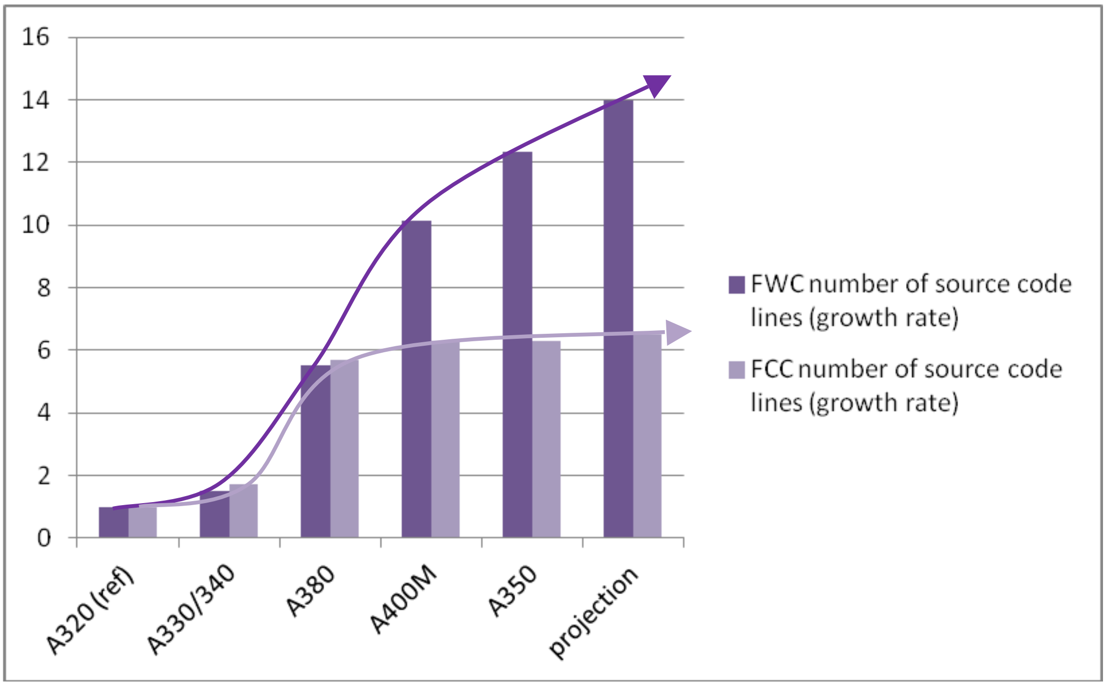

Figure 2: Growth rates of the number of embedded source code line per avionics computer vs Airbus aircraft programs

3.2.2 Size of individual artifacts

The size of individual artifacts is a direct consequence of the previous point. We have to consider that if the complexity raises the number of component per product it potentially also raises the number of elements per component. Nevertheless, this is a problem that can typically be solved using a reductionist approach to recursively break down a big part into a network of smaller subparts. Thus, we will consider this point as a methodological issue and we will keep it out of the Galaxy's scope.

3.2.1 Unbounded models

Unbounded model are typically linked to the representation of execution traces and to the specification of stream processing. The most obvious issue is related to the storage resources. Some modeling formalisms provide facilities for such models. In UML, for instance, the activity graphs can have "expansion nodes" to specify how streams are processed assuming that all items from the stream are managed according to the same algorithm, what is typically the case.

Dealing with (potentially infinite) execution traces is another affair. However, Galaxy focuses on the collaborative development that is mainly about the design part of the development process. Management of execution traces is rather related to verification activities. Galaxy is dimensioned to work neither on the verification aspect of the development process nor on the very specific technologies that would be able to manage true infinite models. This point will then remain out of the scope.

3.3 CAPITALIZATION & REUSE

The ability to reuse modeled artifacts is a key to productivity and there is no way around in practice for a dissemination of MDE technologies in the industrial world. However, capitalization & reuse of model artifacts is a subject by itself. It includes: the universal identification of model element, the material infrastructure to store and to distribute and, on the top of that the organization that manages the lifecycle of those reusable artifacts.

The objective of Galaxy is not to provide a solution for all of these points. Nevertheless some of the technologies that will be studied could be of great interest as technical building block for designing solutions to share model libraries.

3.4 INTERDEPENDENCIES

3.4.1 Artifacts traceability

Traceability is first of all a methodological issue. It can use dedicated relationships created between model artifacts or it can exploit already existing and non-specific ones, which are intrinsically holder of traceability semantics. The purpose of Galaxy is not work directly on the traceability approaches but to provide implementation technologies so that traceability analyses using those approaches can be efficiently made on industrial size projects.

3.4.2 Management of dependent processing

With a full MDE approach, engineering data are spread in multiples interrelated storage units, just like data of a software project are spread in multiples files. Once one tries to build something from the model, for instance to extract a specific view or generate a simulation, we will face a list of recurrent questions: what are all the model parts we need? Where are they located? How must they be processed and in what order?

All those questions are painful and very time consuming to solve if not automated. In software engineering this is the purpose of the “make files” that are automatically generated in any modern software development environment. But where software engineering uses basically only compiling and linking transformations, many more are required for model driven engineering. Galaxy will study a technology that could be used to compute such “make files” for model transformation.

3.5 INFORMATION & DATA SCATTERING

3.5.1 Management of collaborative aspects

As evocated in §2.2, the development of an industrial product is rarely a single man work. Development teams involve classically several tenths of people, who have to share the engineering data provided by the models. “Share” means that those people have to work in parallel on at least one part of those models either for consultation or for modification. Galaxy shall provide mechanisms so that:

- A developer can decide when a version he has modified is made available to the whole team
- Modifications made by two team members will not conflict, once committed.
- All the developers share the same committed version of the model.

We already know that modularity is the key to deal with those issues. The remaining questions are: (a) how to define the perimeter of each part to optimize the collaboration? (b) What is the more efficient technology to manage those parts?

According to an industrial organization such as Airbus, it seems then reasonable to set to 150 the size of the developer team that Galaxy will have to consider for the evaluation of the technologies used for those mechanisms.

3.5.2 Management of the geographic distribution

Development teams cannot always be collocated. Extended enterprise networks may involve people from all around the world in the same development project and even if we consider less international organizations, development teams are often spread in subgroups hosted in distant sites. In such cases, data interchanges shall have to use Wide Area Networks (WAN) – classically Internet - for data interchanges. This may results in bandwidth limitations and security issues that we have to consider during Galaxy's works.

3.6 CHANGE MANAGEMENT

3.6.1 Metamodels evolution

In [R2] it is stated that over the changes that directly impact the specification of a system, the metamodel used to build the model of the product may change during the development phase or later. When the model reaches a certain size, it is no more possible to perform such maintenance operations by hand and automated transformation must be used.

Galaxy shall provide a scalable technology to manage the evolution of metamodels on "industrial size" models, as defined in §3.2.

3.6.2 Model configuration management

As evocated in §3.5.1 and in §3.3, an industrial management of model implies they are broken down into parts which, when reused or shared between several products, will have their own live cycles. More, we may have to manage several versions of the same part even for the development one unique product. The rational might be a change in the specification happening during the design phase or the study of several solutions options.
This kind of issue is well known by software engineers who use tools like CVS or SVN which, by registering individual version for each file code, allow to create and to merge of "branches" aside from the "trunk" of the main development flow. The product is then managed as an aggregate of items, each of them with a given version number.

We need the same kinds of facilities for model parts, and most of current modeling tools serialize models into file what allows using of the configuration management tools initially created for software engineering to manage the models. As a matter of fact and as underlined in [R2], most of current modeling editors offer more or less integrated interface with such tools. The point is that this integration assumes that model parts are stored into files - and if possible text files to be fully efficient – and that this kind of storage might be pointed out as of the main obstacle to the scalability of MDE.

3.7 HETEROGENEITY MANAGEMENT

The development of complex embedded systems require the involvement of several disciplines like: design engineering, dependability, thermal, electromagnetic compatibility (EMC), etc... Each of these disciplines defines a viewpoint or a set of viewpoints that specifies the concepts used and then the model artifacts which represent them. Independently of the viewpoint they belong to, all those model artifacts represent one or more aspect of the same product. In that sense they are all related together through more or less complex relationships.

Applying the MDE technologies for such projects in a reliable way leads to reify as far as possible these relationships that become integral parts of the model. Although the way they must be reified is a methodological problem out of the Galaxy scope, their management so that the consistency of the global model is preserved has consequences at implementation level that are directly in our field of investigations.

First, the synchronization of those multiples viewpoints may require the exchange of significant volumes of data with direct impact on the bandwidth resources. More the transformation of the information between concepts may consume large amount of computing resources.

Second, all those disciplines use their own tools, often based on specific technologies. Thus in addition to the semantic transformations evocated above, some computing resources are necessary to perform the syntactic transformation required for the tools to interoperate.

Galaxy shall provide communication and transformation technologies to minimize the resources required for those syntactic and semantic synchronization.

4. CONCLUSION

The technologies studied in the frame of the Galaxy project will focus on aspects related to the communication level and to the processing level, that is: more than 80% of potential problems pointed out in [R2]. This Lambda document describes also issues related to very specific domains that cannot be worked out within the current budget and partnership of the project. They are:

- Ergonomics issues
- Legacy software management issues
- Infinite models issues

Those issues will remain out of the scope and will require one or more additional research projects.

The technology related to the Communication bottleneck will be studied in the work package 2: “Conceptual model for MDCD”, and will be built on the concept of *collaboration unit*. Scalability aspects related to computing power requirements will be worked out in the work package 3: “Transformation with large and/or numerous models”