Experience of an efficient and actual MDE process : design and verification of ATC onboard systems

E. Bonnafous, E.Saves, E. Gilbert, J.Honoré

CS-SI - Parc de La Plaine - BP5872 - 31506 Toulouse cedex 5
{eric.bonnafous, eric.saves, eric.gilbert, julien.honore} @c-s.fr

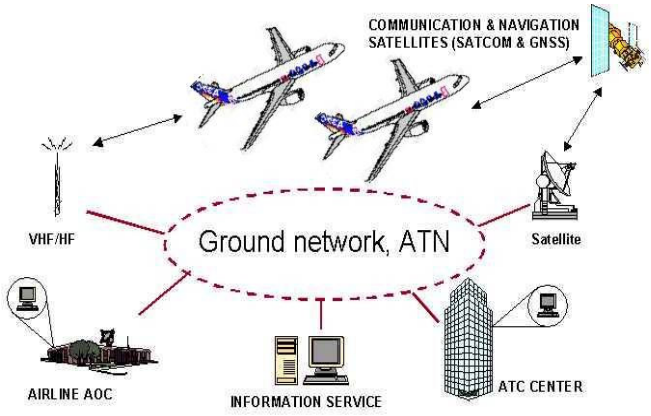
Abstract: Since 2002, CS is involved in the conception and the specification of ATC systems for A320, A340, A380, A400M and A350 aircrafts. The ATC module, a ground/onboard communication system, is designed with SDL, a modelling language normalized by the ITU-T and which is described as a UML2-profile, too.
The SDL well defined semantic allows to have homogeneous code generators and model simulators: these two technics are used in the process of the ATC projects, giving to the process a very efficient productivity.
The automatic code generator generates the C code of the application. The code generator is qualified in accordance to the DO178B requirements (C level). This very strict qualification (development tool qualification) allows to highly reduce the tests effort of the ATC application.
Verifications based on tests are realized on the SDL models, through execution simulation. CS uses RTDS, a SDL Z.100 simulator developed by Pragmadev. As well as providing edition and syntax/semantic checking of SDL models, RTDS provides debugging facilities such as breakpoints and step-by-step execution at the model level and a powerful scenarios language which enables to call some directives of the simulator such as MSC generation, internal signals sending, variables printing and so on…
In order to improve the quality and effectiveness of the ATC model verification, CS has developed a set of applications that are plugged with the SDL simulator through TCP/IP, in order to interact with the system to be tested. These applications are onboard systems (FM, cockpit HMIs simulations: CDS or MCDU/DCDU) and ATC ground simulations: CPDLC, ADS, A623…). For the parts of the complete application manually coded in C code, the simulator is able to call such services using the XmlRpc protocol.
With this complete set of high level and very representative simulation, the process is able to detect and reproduce quite all problems detected very late in traditional processes.
In parallel to this actual process, some research actions are leaded in the ATC project area by CS and ENSIETA. To enhance the safety and the maturity of the ATC product, CS and ENSIETA study and develop a way to perform exhaustive verification of ATC models requirements. The technics used is model checking; studies are focused on a way to systematically reduce the combinatory explosion, and to improve the formalism of properties to be verified. The goal is to make this technic an industrial, cost-effective, and certifiable way of verification.

Keywords: Model driven engineering, critical software, simulation

1. Introduction

The process described below is of interest because it has led to certification in accordance with DO-178B recommendations at criticality level C for applications used in commercial flight since the late 1990s.

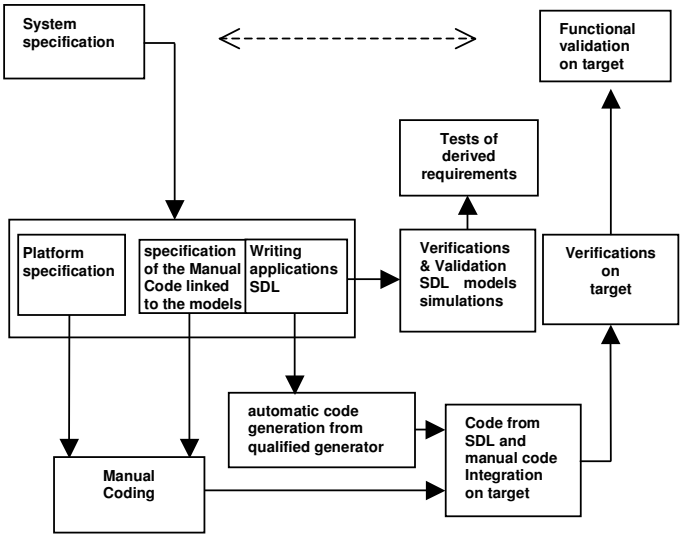
These air traffic control applications can be of the utmost importance as they form part of the CNS/ATM (Communication Navigation Surveillance/ Air Traffic Management). Initially used in low density areas such as oceanic areas, they are being developed more towards surveillance in higher density areas.
Communication is one of the enlarging domain in the aeronautical context : it follows growing circulation in the air space.
ATC (for Air Traffic Control) data-link applications are dealing with communications between an Air Traffic Control Center and an aircraft. It allows manual (involving the pilot) and automatic information exchanges such as alerts, position, speed, weather, etc. ; it is used for certain route and departure clearances too. These applications can operate in ACARS (Aircraft Communication Addressing and Reporting System) or in ATN (Aeronautical Telecommunication Network) networks.



These data were obviously an influence on the main choices that define the process of development. The choice of formalism SDL (Specification and Design Language) is one of them. Indeed, the power of formalism is to specify and design applications unambiguous and driving through automatic code generation to a program true to its specifications. This aspect is crucial to any critical software development where the concept of traceability is central.

2. Process overview

2.1 Development cycle overview



2.2 Methodology overview

The methodology is a classical « V Cycle » methodology. At the earliest system development phases, requirements are classified. Two main sources are identified : requirements for platform definition and requirements dedicated to ATC applications. The platform is hardware dependant and support applications running and datas exchanges between those applications. Platform software and various components linked to models of the ATC applications are developed following a standard DO-178B process and conforms to level C recommendations.

The ATC applications:

- AFN (ATC Facilities Notification): sort of logon for ATC applications,
- CPDLC (Controller Pilot Data Link Communications) : automation than assists with workload management for the flight crew and the controller,
- L'ADS Automatic Dependent Surveillance : position and speed report to the ground
- L'ARINC 623 : Essentially weather information application,

are designed using SDL formalism.

SDL models obtained are identified as detailed specifications of those applications. The activity of traceability between the system and the models leads to the identification of derivated requirements. Those requirements are clearly marked out and are specifically verified. Those verifications bring credit for system certification in addition to system requirements verifications realized on target. The evidence of verification of these requirements are provided by the results of SDL model simulations.

3. Development of ATC applications

3.1 MDA - SDL approach

Airbus ATC applications are based on a model driven approach, MDA. The architecture and detailed specifications are defined using the ITU-T's SDL formalism. This language is perfectly suitable as it can be used to model asynchronous objects that communicate by exchanging signals. Three types of diagrams are used in SDL:

- architecture diagrams representing the breakdown of the application into blocks and processes along with the signals exchanged between those entities;
- state-transition diagrams representing each process's internal behavior;
- MSC diagrams generated by the SDL simulator SDL, allowing all messages exchanged between the various processes to be verified.

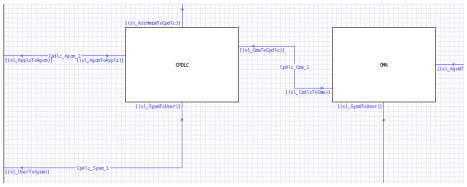


Figure 1: SDL architecture diagram

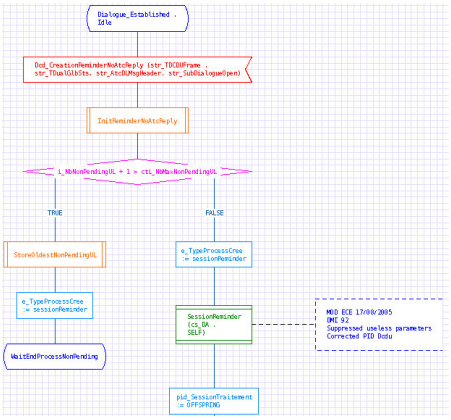


Figure 2: SDL state-transition diagram

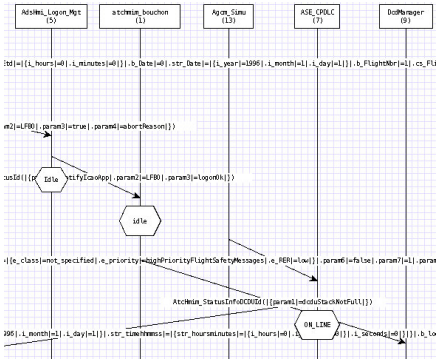


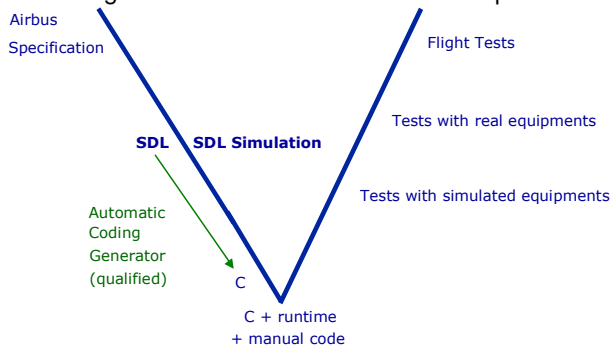
Figure 3: MSC diagram

The RTDS software from Pragmadev is used for the edition and simulation of SDL models.

Using this high-level formalism avoids problems related to hardware services, process scheduling and memory allocation, etc., facilitating **faster development** and **a very high degree of responsiveness**.

2.3 Automatic and qualified generation of code

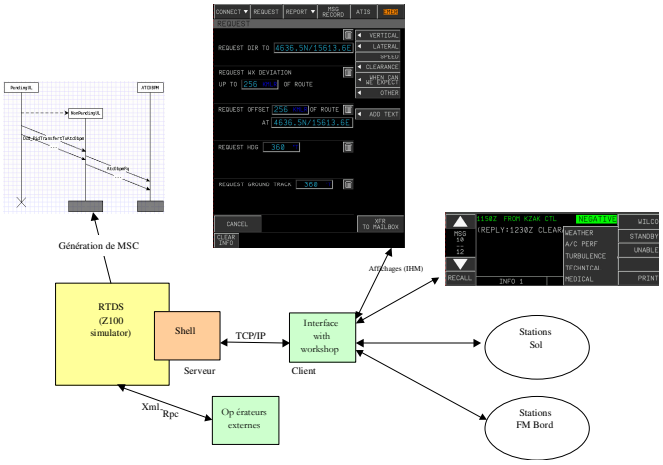
The Automatic and qualified generation of code allows substantial cost savings in addition to an enhanced traceability. Indeed, the verification of automaticaly generated code is greatly reduced. There is no need of reviews of code and additional verifications of the Low Level Requirements. The specific qualified code generator has been specifically design so that hazardous mechanisms should be avoid. Special verification tools prevents from these mechanisms and brings some additional evidences useful for certification credit such as semantic controls according to special SDL design standard.
The SDL generator tool is DO-178B level C qualified.

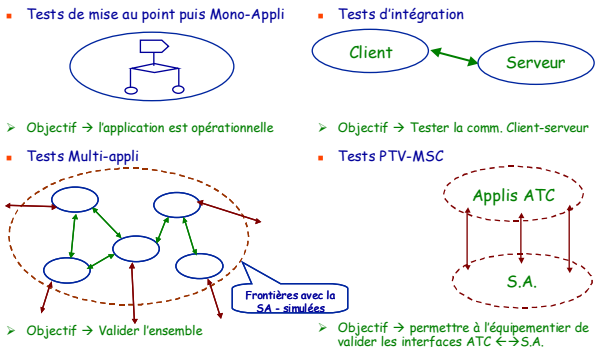


3.3 Verification

As the semantics of SDL (abstract data types, action language) are unambiguous, ATC applications can be checked by simulation. These verifications are performed on PC well before tests are conducted on the actual computer using real equipment. This is much **less expensive** and allows the functional part of the application to be verified **very early** in the development cycle.

This simulation is made possible by the use of equipment simulating the external behavior of ATC applications: HMIs in the aircraft cockpit (CDS, DCDU, MCDU), CPDLC stations for the transmission and reception of messages between ground and aircraft, etc..
This equipment is able to communicate with the SDL simulator. The model is simulated by sending signals and reacting to signals output from the simulator (HMI updates, sending of automatic replies, etc.).
The simulator can also call configurable external operators that simulate hardware services, enabling the carrying out of certain degraded mode tests that are difficult to implement on the end product.
In addition, by close co-operation with Pragmadev, the supplier of the SDL RTDS simulator, developments have been integrated in the package in order to come closer to the actual behavior observed on the target computer. For example, the simulator now includes an option allowing internal signals to be consumed before external signals (i.e. signals from the environment).
Using this complete simulation workstation (customized simulator, HMIs, external stations and operators), extremely good **representativeness** can be achieved in the simulations performed.





3. History of the process improvement

3.1 Starting point

In 2002, CS started producing A380 ATC systems on the basis of work already carried out on site in the AIRBUS design office. Regarding the reply to the call for tender, one of AIRBUS's main requirements for potential suppliers was to improve the process. This mainly concerned test phases and activity traceability.

3.2 Improving the verification process

In general, the first phase for CS consisted in integrating the specific field and assessing the existing processes. The firm's experience in flat-fee management services was one of its most important assets in order to successfully optimize this process.

The two main improvements were made in traceability and test methodologies.

As regards traceability, CS implemented a software solution based on Reqtify, increasing the level of detail of traceability. As a result, the certification audit demonstrated a satisfactory level.

An overall strategy was adopted for the tests, as shown in the following diagram.

In terms of architecture, ATC systems are made up of six high-level blocks, each of which represents a specific function. Four of these are ATC-specific functions and two are servers.
The tests fall into the following categories:

- Single-application tests: The function is tested independently. The boundaries are blocked off around the function's perimeter.
- When the applications have been validated, integration tests are implemented with the aim of validating pairs of communicating functions in client-server mode.
- Then, "multi-application" tests covering all functions are performed.

CS then also defined a methodology for the performance and passing of certification tests.
CS automated these tests, reducing the number of certification test cycles.
The process was then further improved, the single-application tests being close to certification tests.
The process is still developing. The formalization of single-application and certification tests has been enhanced. Like debugging tests, simulation tests are no longer formalized.

Multi-application tests have been completed from the functional point of view and provide the end customer with a basis for customer acceptance tests.

5. Certification process

CRI-F22 tests are among the tests required for the **certification** of ATC applications and servers.
Some of these tests are performed on the **CS simulation** workshop in single-application mode (application/server isolated from other applications/servers)
and cover:
- all the requirements specified in DFD (Derived Functions Document),
- functional requirements of specifications that AIRBUS laboratory test facilities cannot cover.

The representativeness of the simulation facilities is thus one of the key criteria for this A380 ATC certification phase.
For this purpose, CS integrated a representativeness improvement process into its simulation workshop.
Each fault detected on an AIRBUS aircraft or simulation bench is played back on the CS simulation bench.
The test results are compared and, in case of any discrepancy, the modifications required to increase the workshop's representativeness are implemented.

By this iteration process to improve the representativeness of the simulation of ATC detailed specifications, such a high **level of maturity** has now been achieved that **behavior is the same on the CS simulation bench** as on aircraft or the AIRBUS bench for virtually 100% of faults detected.
This environment now enables CS to achieve excellent quality in the integration of modifications and fault corrections.

The current use of a simulation workshop for both the CRI certification of derived functions and those requirements that cannot be tested on AIRBUS benches is paving the way to the wider use of **virtualization tools for certification.**

In keeping with its aim of improving the quality of ATC systems, CS has anticipated the traceability requirements for certification. This has enabled CS to provide for a traceability granularity that goes well beyond the initial prerequisite. This anticipation was crucial for certification, since this new granularity has proved to be necessary.

Lastly, a study of **formal proofs** on the basis of detailed technical specifications is being carried out at CS in order to provide for the future of certification tests, and this study is expected to lead to positive results.

6. Conclusion and further work

In conclusion, the process of developing ATC applications by means of SDL formalism has proved itself in terms of reliability since this new system's introduction on the SA and LR AIRBUS. This history along with rigorous formalism and the process's constant improvement has instilled confidence in the certification authorities, end users and airlines. This example of the successful utilization of models in the context of highly asynchronous, critical applications provides a wealth of information. It contributes support for the MDE approach and holds out promising prospects for the utilization of simulation results as proof supporting the certification of critical aeronautical systems.

As modern avionics incorporate a growing number of increasingly complex systems and as the current trend is to meet increasingly strict reliability requirements, the manufacturing process will have to continue to evolve towards greater exhaustiveness and formalization.
With this in mind, CS has since 2004 taken the initiative of conducting studies on the applicability of formal proof to ATC models, as part of its R&D. Between 2007 and 2010, this work will involve the writing of a thesis aimed at incorporating exhaustive verification by means of proof techniques using ModelChecking into the current process. CS has great hopes that this approach will equip it to tackle the challenges to come.