Open Source and Embedded Software Development for Avionics

Presented by

Pierre GAUFILLET
Software Engineering Specialist
Avionics and Simulation Products
AIRBUS
Summary

• Open Source strategy for avionics software development

• Status of Open Source solutions for software development in AIRBUS
Summary

• Open Source strategy for avionics software development

• Status of Open Source solutions for software development in AIRBUS
The avionics software beginnings

- First significant use of software for avionics: the 80’s
The avionics software beginnings

• At that time, and for some years, « software » seemed nearly synonymous with « bug »... :
System/software engineering tools : a short story

• From the very beginning, we have needed tools to develop the avionics software at the requested dependability/productivity levels

• As there was nearly nothing available on the market place, major airframers / equipment suppliers in Europe developed their own solutions (make) :
  ‣ Automatic Code Generation, Process Management, Test automation, etc.

• Progressive reorientation of our companies on our “specific skills” led to the move from “make” to “buy” :
  ‣ The internal tools are transferred to editors who industrialize them (SCADE, RTRT, etc.) which leads to
    – … improving them through massive investment
    – … trying to sell them on a “larger” scale … that today doesn’t actually exist

• Leading to the current situation of the software tools market for dependable embedded systems…
Long term availability…

One example: AIRBUS A300

• The program began in 1972 and production stopped in 2007
  \[2007 - 1972 = 35 \text{ years}\]

• The support will last until 2050
  \[2050 - 1972 = 78 \text{ years}!!!\]

On board software development for very long lifecycle products
... and volatility

Industry / research centres

- AUTAN
  - AIRBUS

- SCADE
  - AIRBUS/Schneider Electric/VERILOG

- Object/GEODE
  - VERILOG

- CAVEAT
  - CEA

Commercial world

- ATTOL
  - MARBEN

- RTRT
  - RATIONAL

- SCADE
  - VERILOG

- Object/GEODE
  - VERILOG

- TELELOGIC

=> mastering, continuity, long term availability not always guaranteed...
Distribution difficulties

• Most of our software are developed with partners and sub-contractors in an extended enterprise mode:
  ‣ Availability of development tools is not problem-free in this context:
    – deployment cost,
    – number of licenses, etc.
  ‣ It may become impossible to manage when some tools integrated in our development framework are not distributed anymore:
    – Code controller commercial tool: we discovered that distribution had been stopped when a new sub-contractor asked for new licenses…

• And sometimes, we face unacceptable practices:
  ‣ Example: tool move to a new machine, without any order change

=> Value for money ???
Many innovative tools are not distributed

• Too many tools developed by Airbus equipment suppliers, or innovative tools available in research labs as prototypes, are not largely available:
  ▶ They are essential for the developments but not easily endorsed by classical means due to the investment required
  ▶ They are most of the time de facto “proprietary” and thus not easily available for deployment on a large scale
  ▶ They are too specialized and/or too costly for a profitable business in a very small market => no stable vendor => very few users => vendors crisis,…

The result:

stagnation (and even decline) of the offer of innovating tools

leading to a

stagnation of the overall productivity

of embedded system developments
And the contradiction increases …

Increase in Size and Complexity, and Decrease in Price: an Illusion. The fact that there will be more embedded software systems and more complexity within them in the future is a tendency which is well recognized by the Business Units. I agree completely with that view.

Now, the sometimes mentioned conclusion by the Business Units that the development price of such embedded software systems should be limited and that the deadline to achieve their construction has to be shortened is, in my mind, an illusion.

EADS minimise l'impact d'un dollar faible

AMSTERDAM (Reuters) - EADS prédit que l'impact négatif d'un dollar faible sur ses comptes s'accroîtra de près d'un cinquième cette année mais il a assuré aux actionnaires qu'il pouvait y faire face par le biais d'opérations de couverture et une réduction des coûts.

Toutefois la récente glissade du dollar face à l'euro rend d'autant plus obligatoire pour le groupe de défense européen de réaliser son programme de réduction des coûts chez Airbus, connu sous le nom de Route 06.

- Embedded systems complexity increases but with strong request to decrease costs

=> we need more modelling, more virtualization and more automation, so more tools but less expensive ones
Open-Source: a solution?
Open-Source …

• contributes to **standardization**, a need for this industry as for others…
  
  ‣ “Gerstner reportedly observed to key IBM insiders: "This is the only industry where competitors don't regularly agree on standards to enable greater value for customers." ” — An Open Secret.

  *The economist (Oct.22, 2005), quoted in ACM communications October 2005*

• **avoids unavailability of** the knowledge (essential for certification purpose)

• contributes to **cost reduction**

• restores the hope of **tools sustainability** (federating the market of tools for ERTS development tools, gathering a significant community of users)
Open-Source …

• does not ease the access of our competitors to the technology, no more than proprietary SW (1st cost item for a SW vendor is marketing effort)

• may improve the common use of the tools amongst the stakeholders of a same domain (e.g. Airbus & its suppliers)

• stimulates creativity and innovation by easier access of contributions for the whole community (industry, research, education)

• enhances meritocracy: the best IP (Intellectual Property) solution becomes the Open-Source standard

• eases access to the tools for the education community
… for all these reasons, Airbus commits itself…

…with other industrial partners in the ISAURE federative program (Aerospace Valley Competitiveness cluster)

- To an Open-Source approach for system/software engineering: TOPCASED, GENE-AUTO, etc.

- Non exclusive:
  Others projects in ISAURE

- Open:
  - OPEES with SYSTEM@TIC and MINALOGIC
  - EICOSE (with Safetrans, SYSTEM@TIC etc.)

- In an extended industrial partnership:
  - Trans domains synergy (Aeronautics, Space, Automotive, etc.)
  - Software service companies
  - Tool vendors

- In partnership with the academic community (research & education)
... with Boeing soon?

Boeing’s code contribution to OSEE

The Eclipse Open System Engineering Environment (OSEE) project is one I have been interested in watching. Of particular interest is the fact it is being led by a group of engineers inside Boeing. In my opinion, I think the next wave of open source projects will involve non-traditional ISV’s, companies like Boeing, building industry specific frameworks. Who better to led a system engineering framework project than a group that designs software for aircraft manufacturing?

Therefore, it is great to see that OSEE now has code available in the repository. Also, check out the extensive set of screencasts to help people learn about OSEE. Seems like they are off to a great start.

This entry was posted on December 18, 2007 at 4:24 pm and is filed under Uncategorized. You can follow any responses to this entry through the RSS 2.0 feed. You can leave a response, or trackback from your own site.

4 Responses to “Boeing’s code contribution to OSEE”

Alex Bilewitz Says:
December 18, 2007 at 5:21 pm

The link to OSEE is one character off: your E is before the a, not after it.

ianskerrett Says:
December 18, 2007 at 6:42 pm

Alex.

Bored.

beredt Says:
December 19, 2007 at 9:50 am

I think this is similar to the TOPCASED project, which is being supported by Airbus, isn’t it?

ianskerrett Says:
December 19, 2007 at 1:36 pm

Beredt,

I am not an expert on either TOPCASED or OSEE but my understanding is that they are complementary in many areas.
Opening « more than source code »?

Specialized for given certification standards

Workflows

Document Templates

Qualifiable Tools

Certifiable Components

Education Materials

PSAC SDP, SVP, SCMP...
Standards SAS...

OS runtimes IP stack middleware...

Toy certifiable projects Specialized Examples (e.g. for do-178c annexes)

OSEE Couverture

Gene-auto Topcased
Why an open initiative for the DO-178 world?

- A support to the DO-178C effort
  - Educational materials for clarifying intent
  - Experimental test-bed for annexes

- Avionics industrial community
  - Provides a shared infrastructure
    - For long term investment
    - For long term cost reduction
  - Allows some level of cooperation with competitors
  - Lower training costs (especially for subcontractors)

- Certification authorities
  - Lower training costs for DERs
  - Vehicle for clarifying specific issues
  - Help sharing or practices between authorities

- Tool providers
  - Offers an ideal showcase for their open technologies
  - Tool sharing makes it easier to provide a complete supported solution
  - Creates an ecosystem where everyone can meet potential customers and partners

Open-do.org
To conclude…

• Neither « make », nor « buy » is the right answer today to our needs for ERTS development tools

• We support a new software edition landscape, with three props : Open-Source, CSS, Commercial products

• This new landscape won’t exist without support from industrial end users and public incentives. It must be :
  ‣ Realistic = not “make” again, and loose illusion that “buy” may rise spontaneously in our domain …
  ‣ Funded = to start the virtuous circle
  ‣ Organized = need for an international eco-system to :
    – Define a consistent architecture to better integrate the contributions,
    – Federate contributors scattered all over the world, with editors who develop their tools on top of the generic components, and service providers
    – with a light structure (marketing/com.) to develop the community.
Summary

• Open Source strategy for avionics software development

• Status of Open Source solutions for software development in AIRBUS (extended enterprise)
In innovation projects

• Projects to support Model Driven Engineering process by providing tool chain:

  ‣ TOPCASED
  ‣ GENE-AUTO
In innovation projects

TOPCASED (www.topcased.org)

• Develop an Open Source system development environment, implementing an integrated model-based dev. process, from the system specification down to the final product
• Covering modelling (edition, transformation, simulation), code generation, verification
• Offering framework common services:
  ‣ Tools interoperability (backbone, standard interfaces)
  ‣ Generic services (configuration, change, traceability,… management)
In innovation projects

**GENE-AUTO**

- Develop an Embedded software code generator tool from Matlab/Simulink
  - based on the common user requirements specifications,
  - able to produce optimized & certifiable embedded software code,
  - supporting systems and software engineering approach
  - Open source
Final project reviews with ITEA and DGE:

"The work done on qualification was impressive and a positive point was that experienced partners were willing to share their knowledge with others"

"Successful project with a great potential for the future."

already disseminated:

- AdaCore selects GeneAuto for their Simulink auto code generator

- Rockwell Collins is currently trying geneauto for Display units.

- Gene-auto integrated in the Edona project (System@tic) for test generation
In innovation projects

- Projects to support the use of System on Chip in critical embedded systems:
  - SOCKET
    - Define a seamless design flow including certification constraints from the system level down to the SoC
    - prototype an Integrated Development Environment:
      - implementing the SoCKET process
      - based on open standards of the market and using open source components,
      - providing: High level synthesis, heterogeneous simulation techniques (SystemC/TLM), Formal verification by model-checking, Test cases automatic generation
      - Allowing IP encapsulation & interoperability
In innovation projects

• Projects to support verification:
  CAT / U3CAT

  ‣ Develop formal verification techniques for C code verification
  ‣ Based on open source framework (Frama-C) and open source plug-in
In Aircraft programmes

- **SAM** (Structured Analysis Modelling) editor (TOPCASED)
  - Software specification model
  - maintenance and flight warning system A350
In Aircraft programmes

• Use of **Eclipse IDE** for software development of maintenance system (A380, A400M)

• Use of **GNU compiler** (C, ADA) for software development (all A/C)
In Aircraft programmes

• Remember examples in the first slides:

Distribution difficulties

• Most of our software are developed with partners and sub-contractors in an extended enterprise mode:
  ‣ It may become impossible to manage when some tools integrated in our development framework are not distributed anymore:
    – Code controller commercial tool: we discovered that distribution had been stopped when a new sub-contractor asked for new licenses...

• Now solved thanks to Open-Source development based on Frama-C:
  ‣ Small investment
  ‣ Better coverage
  ‣ Long term availability
  ‣ Convergence
In Aircraft programmes

- Virtualized test platform based on an adaptation of Open Source BOCHS (PC simulator) for
  - A320 family (Flight Warning System)
  - A330/A340 family (FWS + Maintenance computer)
In Aircraft programmes

- Virtual I/O board based on **UNISIM** (CEA Open-Source product) for the Audio card of Flight Warning System (A330/A340 family)
In Aircraft programmes

- **GPM** Generic Process Modelling (TOPCASED) used in A350
  - instantiation -> configuration management, change management
In Aircraft programmes

- SVN (Subversion) Version control system, used for large tools development since 2008
Open Source (G.Ladier)

Open Source Approach presented was approved by all participants. Decisions on recommendations:
In conclusion

• Open source engineering tools are already available and deployed in aircraft programmes

• Current running research project should complete the tool chains

• The next step is to ensure long term maintenance and evolution of the tools through adequate infrastructure and organization

→ This is the purpose of OPEES initiative
OPEES project

Consortium for the definition, elaboration & deployment of an

Open Platform for the Engineering of Embedded Systems
Ensuring long-term availability of critical / embedded systems engineering technologies to secure industry competitiveness and development

OPEES
OPEES general overview

• Coverage of the whole lifecycle
  ‣ Analysis & Design
  ‣ Verification & Simulation
  ‣ Execution infrastructure
  ‣ Exploitation

• Working priorities
  ‣ Interoperability, Open Standards & Open Source
  ‣ Consistency of solutions for integrability
  ‣ Maturity: quality & support

Vision
Choice
Diffusion / actors

Critical and/or embedded systems

Aeronautics
Automobile
Energy
Telecommunication
Health
...
OPEES main goals (1/2)

• A technical repository to allow sharing of open-source tools & components in the domain of embedded systems/software engineering and to support collaborative works.

• A first set of core components & tools that will be matured and assessed through the proposed methodology and will be available through OPEES as the baseline for further R&D activities.

• A set of professional services to support the users.
OPEES main goals (2/2)

• An open and visible **organization**, implementing the business models / business plan and dissemination plan defined in the project allowing the emergence of a European service industry

• A **process of maturation** for the candidate tools & components validated through experimentations on real use cases to ensure adequacy with industrial needs and constraints

• A **qualification process kit** for selected components to answer the needs of safety critical embedded systems
OPEES lifeline

R&D Project X

R&D Project Y

R&D Project Z

Mature components & solutions

Experimental & mature components

OPEES ITEA Project

OPEES Organisation

End-User A

End-User B

End-User C
OPEES Actors and Processes

- **Govenance**
- **Vision & Roadmap**
- **Technical Repository**
- **Maturation Process**
- **System Engineering Process**
- **Service Providers**
- **SME SSII Tool vendors**
- **Users**
- **Industrials SME**
- **R&D projects**
- **Labs SME industrials**

**OPEES members**
(Industrials, SSII, SME, Labs, Clusters)
OPEES organization

- **Governance**
  - steering role
  - sets the orientations to ensure the mission

- **Operations**
  - provides common infrastructure maintenance
  - provides a no-charge access to OPEES Components/Solutions under open-source licenses
  - provides maintenance and support to OPEES Components/Solutions contained in the repository with a commitment to continuity of service in the long run
  - manages maturation processes for experimental components
  - provides a small number of integrated, polished and maintained solutions
  - delivers support, training and customization service with commitment to continuity of service on the long run.
OPEES workflow

- **Solutions**
  - Integration
  - Added value / reuse

- **Mature Components**
  - Maturation
  - Contributions / hosting

- **Experimental Components**
  - Opees Forge (FLOSS)
  - External Artefacts
OPEES WBS

WP1: Organization & Management

WP2: Technical Roadmap

WP3: System Engineering for Durability

WP4: Implementation & Validation

WP5: Dissemination

Requirements

Feedback
Business Models

• Expected business models or specific strategies drive the choice of open-source licences

• Expected services
  ‣ Support
  ‣ Corrective & evolutive maintenance
  ‣ New developments
  ‣ Integration / tailoring
  ‣ Maturation
  ‣ Training
  ‣ Configuration-controlled solutions for specific programmes
  ‣ Qualification / certification
OPEES Status

• French KOM 12/05/2009 :
  ‣ 35+ attendees :
    – Industry : Airbus, Astrium, Schneider Electric, CS, ATOS, Anyware, Dassault, MBDA, CNES
    – Labs : CEA, ONERA, INRIA, IRIT
  ‣ Strong motivations :
    – Availability of advanced tools (Astrium ST) in a « certification » context (Dassault, Airbus)
    – Durability (all…)
    – Business models
  ‣ An already significant set of available tools :
    – Modeling tools (e.g. : Papyrus>UML, Polichrony, AADL,…)
    – Simulation (e.g. : UNISIM)
    – Verification tools (e.g. : FRAMA-C, AGATA)
    – Automatic Generation (e.g. : Gene-auto)
Conclusion

• OPEES approach
  ‣ Critical / embedded systems development
  ‣ Covering the whole lifecycle (including exploitation)
  ‣ Durability and maintainability on the long term
  ‣ Settling a European service industry
  ‣ Keeping competence locally
Thank you for your attention!

Gérard LADIER  
gerard.ladier@airbus.com

Pierre GAUFILLET  
pierre.gaufillet@airbus.com
© AIRBUS S.A.S. All rights reserved. Confidential and proprietary document.

This document and all information contained herein is the sole property of AIRBUS S.A.S.. No intellectual property rights are granted by the delivery of this document or the disclosure of its content. This document shall not be reproduced or disclosed to a third party without the express written consent of AIRBUS S.A.S. This document and its content shall not be used for any purpose other than that for which it is supplied.

The statements made herein do not constitute an offer. They are based on the mentioned assumptions and are expressed in good faith. Where the supporting grounds for these statements are not shown, AIRBUS S.A.S. will be pleased to explain the basis thereof.