

Middleware for a distributed and hot-redundant software

Use of Ada 2012 in a railway application

Ada-Europe 2016
Vincent MONFORT

Pisa, 16 June

Systemerel Group

- Systemerel Group
- Objectives
- Introduction
- Principles
- Ada features
- Feedback
- Conclusions
- Contact

Systemerel Group
Turnover of € 8 million
Over 100 engineers



Critical systems engineering

- Safety studies
- Safety critical software development
- Formal methods



Electronic & Embedded equipment

- Digital electronics
- Signal acquisition and conditioning
- Signal processing



Objectives

- To describe the middleware characteristics and main principles
- Use of Ada₂₀₁₂ features for developing an industrial product
- Feedback on Ada₂₀₁₂ and environment tools

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

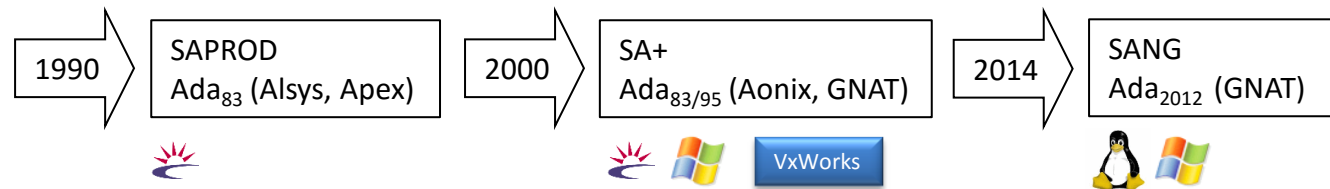
Conclusions

Contact

Introduction - 1/2

Alstom Transport developed and has used since 20 years an Ada_{83/95} middleware for its ATS (Automatic Train Supervision) and FEP (Front End Processor) railway equipment.

After a study and software model made by Systemel, Alstom Transport entrusted Systemel with the complete overhaul of SA middleware.



Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

SANG (SA+ Next Generation) is a full Ada₂₀₁₂ middleware:

- Provides a generic and high level interface able to host a supervision software
- Hides mechanisms of communication, distribution (not using Annex E) and hot-redundancy
- Not dependent on Operating System
- Integrated in a SIL2 (EN 50128: Safety Integrity Level) process
- Guarantees performances and high availability for application software



~ 10 000 LOC full Ada₂₀₁₂,
~70 procedures and functions API

Systemel Group

Objectives

Introduction

Principles

Ada features

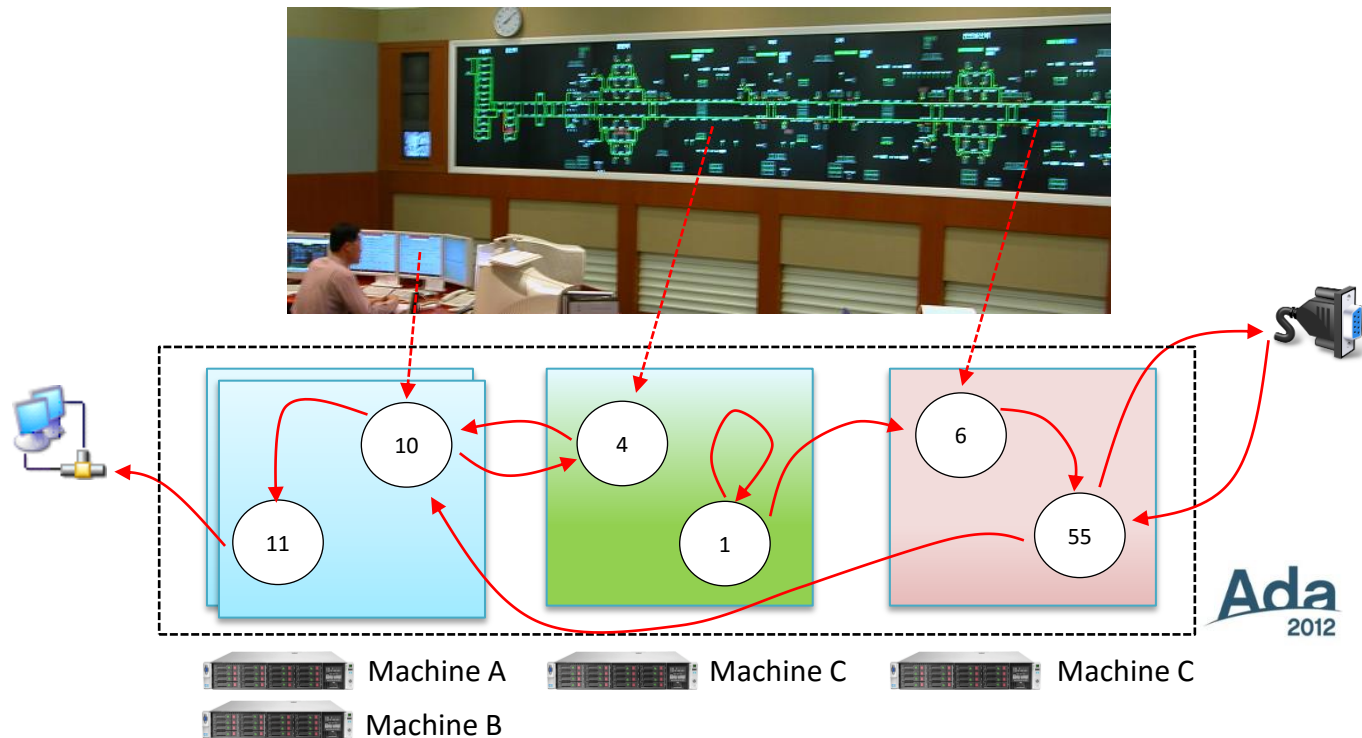
Feedback

Conclusions

Contact

Middleware principles - 1/5

SANG hosts application functions, for specific applicative treatments, in a distributed and redundant architecture.



Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

Middleware principles - 2/5

An application function is able to receive or send messages:

```
type Message_T (From : Fid_T;  
                To    : Fid_T) is abstract tagged private;
```

A source function does not know which machine hosts the target function.

Guarantees:

- Message will be delivered to target function
- Redundant function data and unprocessed messages will be restored in case of failure

Systemel Group

Objectives

Introduction

Principles

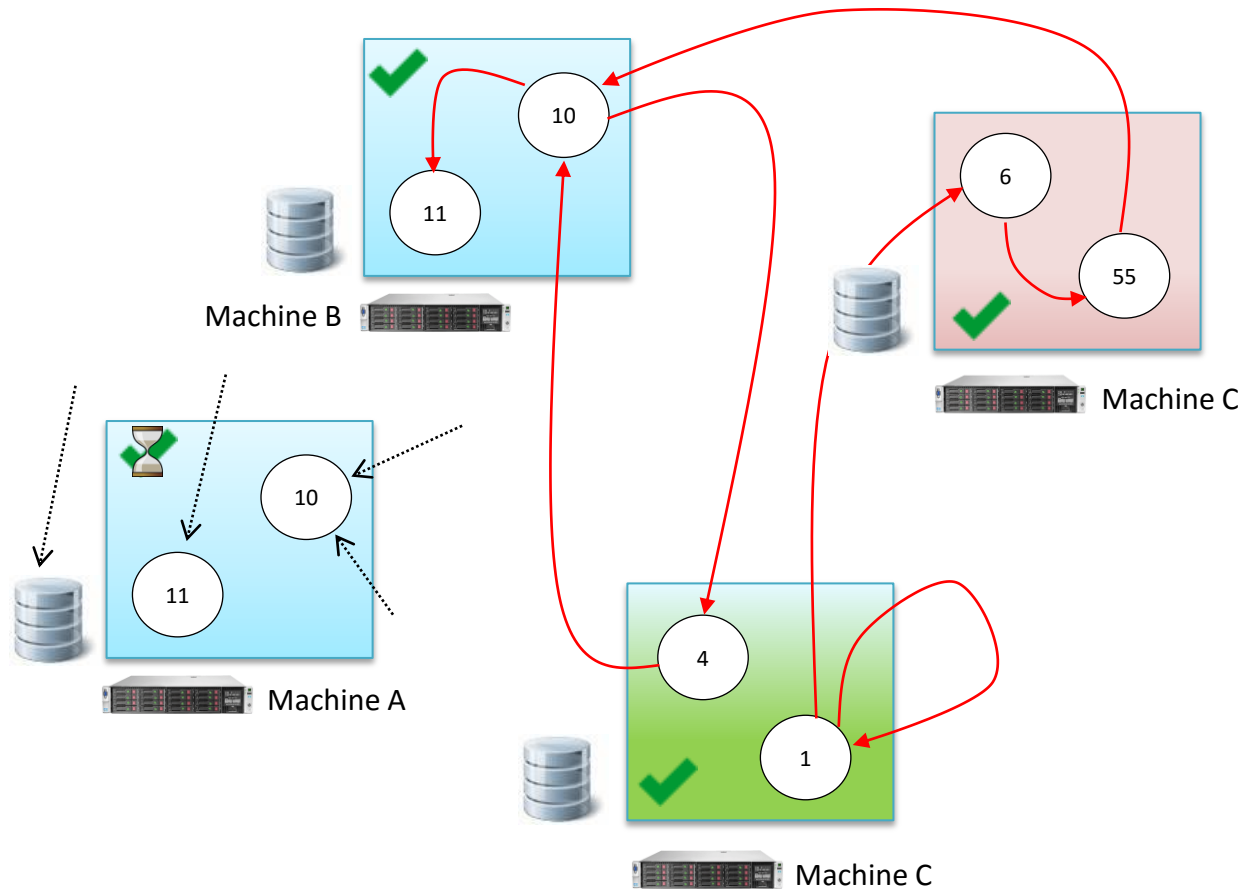
Ada features

Feedback

Conclusions

Contact

Middleware principles - 3/5



Systemel Group

Objectives

Introduction

Principles

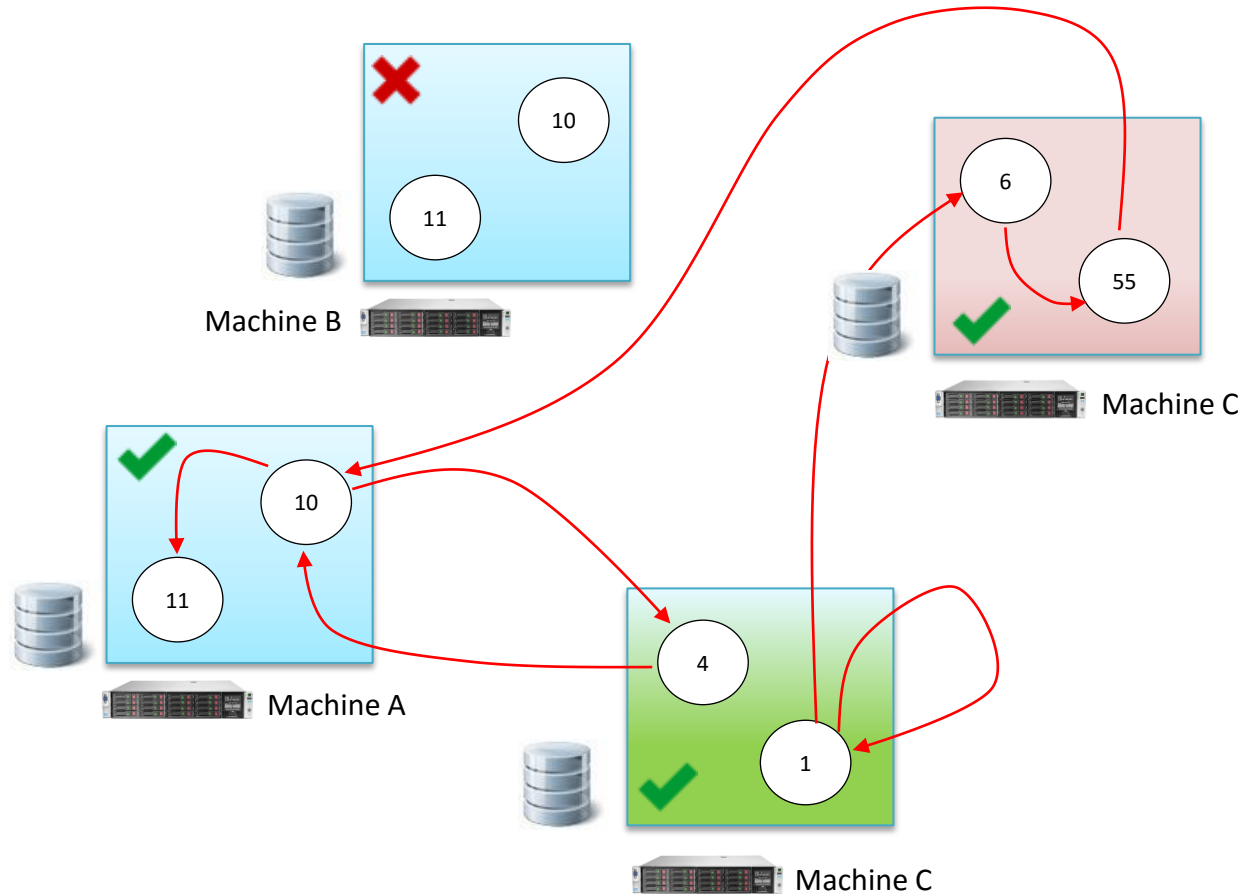
Ada features

Feedback

Conclusions

Contact

Middleware principles - 4/5



Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

Middleware principles - 5/5

Function code is called on message reception:

```
procedure Process_Message (Message : in Message_T'Class) is  
begin → 1) reads received message  
        2) updates internal data →  
        3) sends messages
```



A function is a separated task with a CPU in CPU_Range associated.

Sending a message is very simple:

```
type Msg_Function_4_To_10_T is new Message_T (From => 4,  
                                                To   => 10)
```

```
with record
```

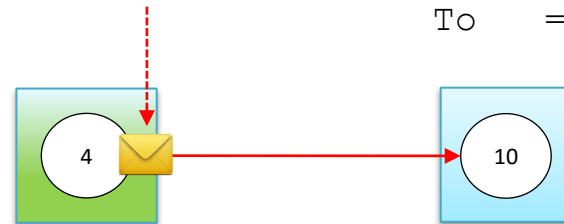
```
    My_Data : ...
```

```
end record;
```

```
...
```

```
My_Message : Msg_Function_4_To_10_T;
```

```
My_Message.Send;
```



Redundant data structures are based on Ada containers

Use of Ada features - 1/2

The new middleware gains from use of Ada₂₀₁₂ features.

One of the benefits is a code size reduction of ~80%!

Main Ada₂₀₁₂ features used are:

✓	Preconditions and postconditions	Development, expectations and obligations of API
✓	Conditional / Quantified expressions - Expression functions	Expressiveness / Readability
✓	In-out function parameters	Messages sent / received: Function task → Client task Server task → Function task
✓	Iterators	Each function task Middleware core tasks
✓	Task-safe queues - Holders	Waits for all function tasks to execute initialization code
✓	Multiprocessor affinity	
✓	Synchronized barriers	

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

Use of Ada features - 2/2

Ada₂₀₁₂ non used main features:

- ✗ Type invariants
- ✗ Subtype predicates
- ✗ Ravenscar for multiprocessor systems

AdaCore[®] GNAT Pro tools are also widely used:

- ✓ GNATCheck
- ✓ GNAT.Sockets (stream-sockets)
- ✓ GNAT.OS_Lib
- ✓ XMLAda
- ✓ GNAT.MD5
- ✓ GNAT.Regpat
- ✓ GNAT.Traceback.Symbolic / GNAT.Source_Info

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact



Downsides:

- Bugs in GNATPro 7.2 linked to Ada₂₀₁₂ features use
- Need to re-implement `Ada.Real_Time.Timing_Events`:
 - Non protected call-back (multi-threaded)
 - Isolated from application use of timers
- Performances issue due to combination use of serialization and stream sockets (2MBytes /sec CPU time):
 - Implementation of Stream Memory Buffer
 - Change: write it to stream in one piece (String)
 - Use it as intermediary to stream a message on socket
=> x50 faster and CPU use back to normal !

Systemel Group

Objectives

Introduction

Principles

Ada features



Feedback

Conclusions

Contact



Upsides:

- Relevance of contract programming → quick and efficient for integration and validation phases
- Expressiveness and efficiency of Ada₂₀₁₂ features
- OOP and concurrency management are powerful tools for a message oriented middleware
- Set of properties and features of Ada and GNAT Pro enabled to make hot-redundancy finally work
- Full O.S. portability  ↔ 
- Participate to improve efficiency of AdaCore®
Ada.Containers.Unbounded_Priority_Queues (NF-17-OB05-042)

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

Conclusions

- First important industrial project developed by Systerel using Ada₂₀₁₂ for a railway server (middleware + application)
- Systerel is convinced that Ada₂₀₁₂, as it was the case for Ada₉₅, is a major evolution of the language
- Quick development and finalization of the middleware draw upon Ada features (contracts, etc.)
- Final railway product is robust and efficient (tested with at least 150 trains communicating)
- Despite of a few issues with the compiler, we are satisfied of this new language version.
GNATPro7.4.1 validates this last point.

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact

Systemel Group

Objectives

Introduction

Principles

Ada features

Feedback

Conclusions

Contact



Safe real-time solutions

Vincent Monfort
Senior engineer

+33 1 76 60 40 24
vincent.monfort@systemel.fr

Thank you