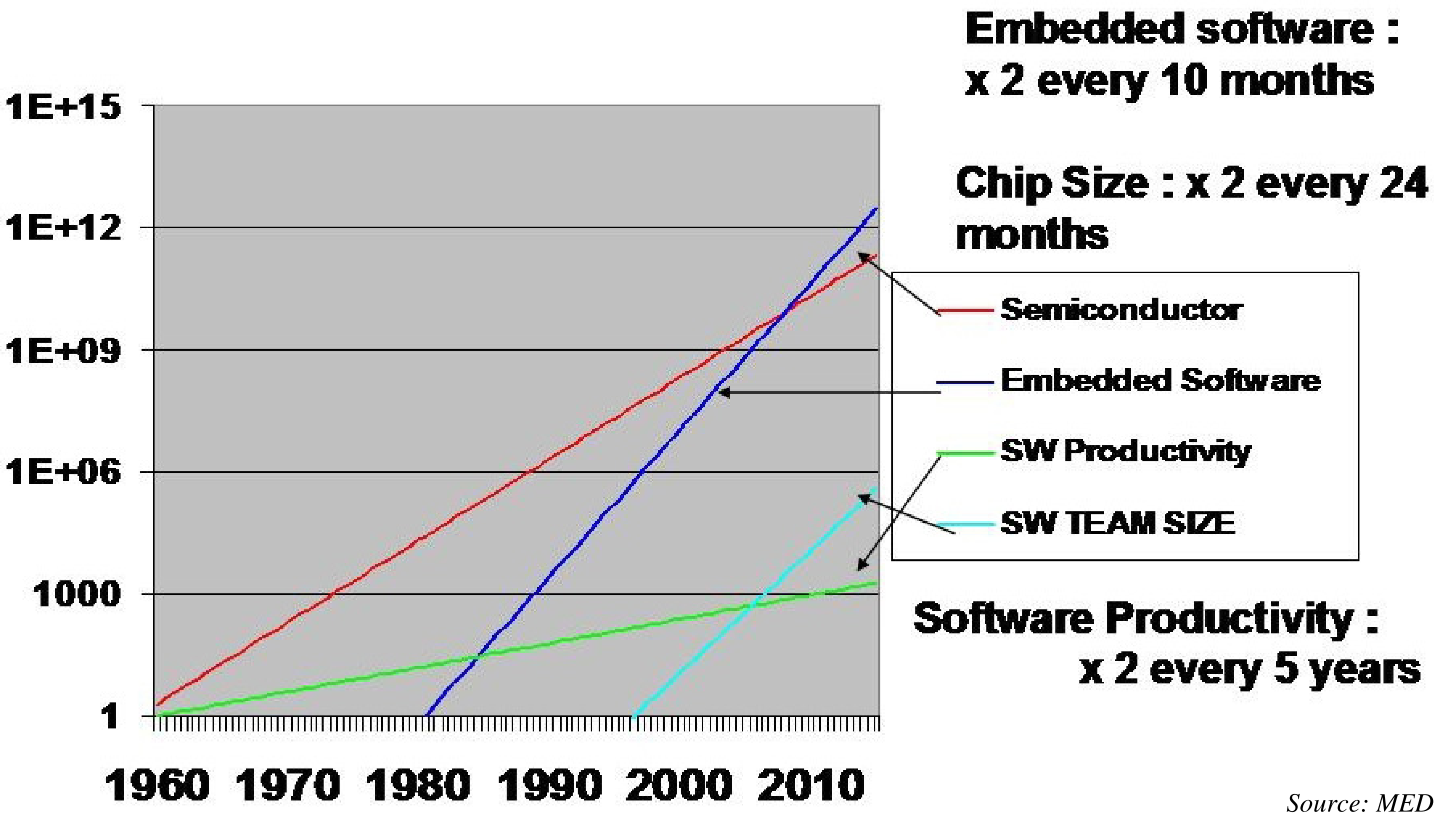
MIND OW2 project

OW2 Technology-Council

2010, March 15

Matthieu Leclercq - STMicroelectronics

Embedded software development and productivity



Embedded software : x 2 every 10 months

Chip Size : x 2 every 24 months

Software Productivity : x 2 every 5 years

Source: MEDEA+ White Book

Context (1)

- Problematic
  - Constant growth of embedded software in electronic devices
  - Hardware platforms more and more complex
    - Mono-processor -> Multi-processor -> Many-processor
  - Different domains, different constrains
    - Reliability (Schneider)
    - Power saving (STEricsson)
- Expected Impacts
  - Reduce development cost and improve reusability
  - Common efforts for the development of a base technology

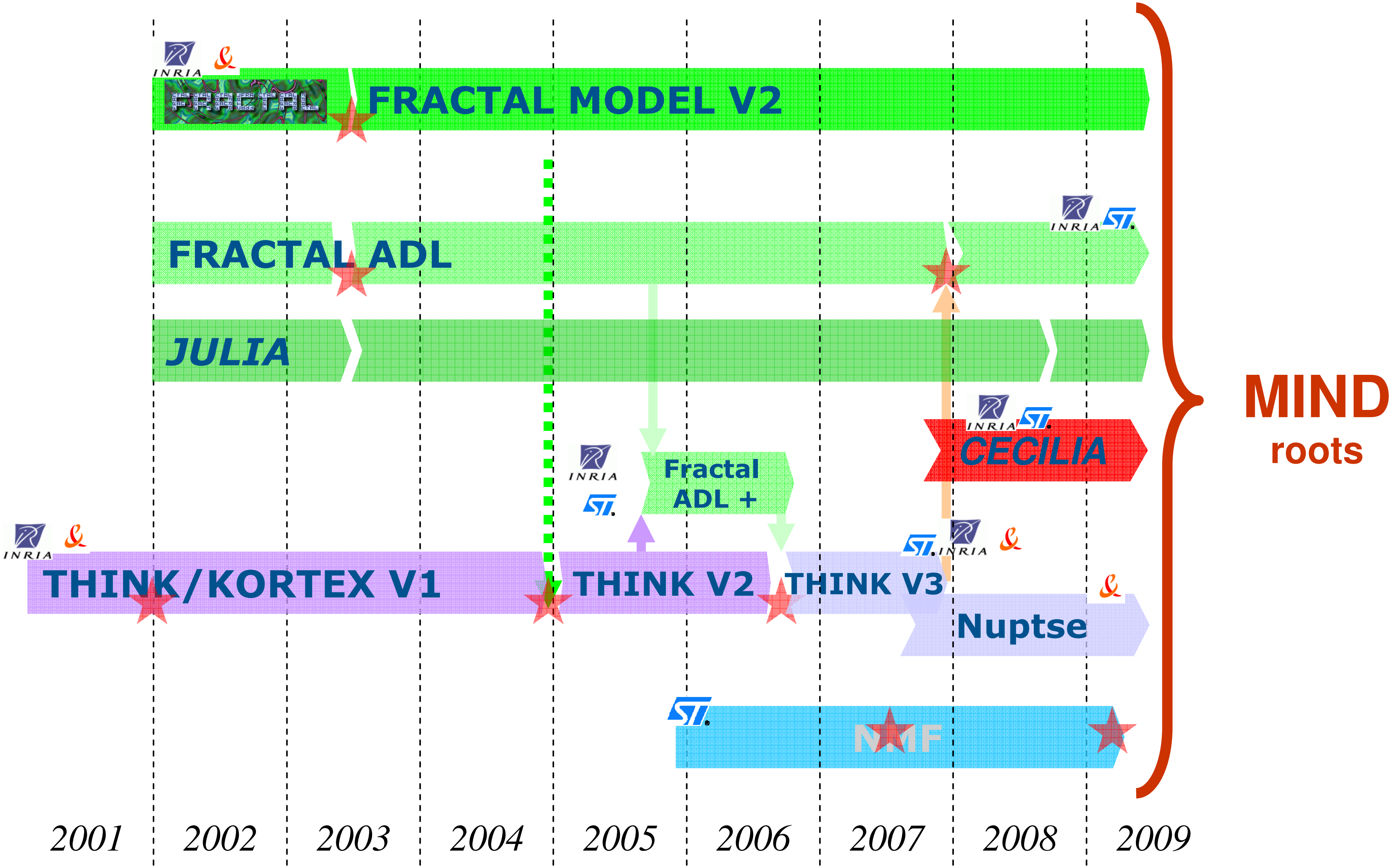
Context (2)

- The MIND Minalogic funded project
- Minalogic : French “Pôle de compètitivité”
- 15 Partners
  - Industrials
    - ST, STE, FT, Schneider, Logica, Sogeti, Open Group, Itris, IS2T,
  - Research Labs
    - UJF (Adele), INRIA (Sardes, Adam), VERIMAG, CEA List, INERIS, ISTIA
- 30 months (Sept 08 - Feb 11)
- More than 70 Men.Year.

Goals

- Provides an industrial-level development environment for component-based embedded software
  - Based on the **Fractal** component model
  - **Extensible** to be adapted to various industrial contexts
  - **Integrated** in standard development environment (Eclipse)
- Based on the long history of component-based software design already developed in OW2
  - Fractal, Think, FractalADL, Cecilia, NMF, …

MIND roots


INRIA
FRACTAL
FRACTAL MODEL V2
INRIA
FRACTAL ADL
JULIA
INRIA
CECILIA
INRIA
Fractal ADL +
INRIA
THINK/KORTEX V1
THINK V2
INRIA
THINK V3
Nuptse
NMF
MIND
roots
2001
2002
2003
2004
2005
2006
2007
2008
2009

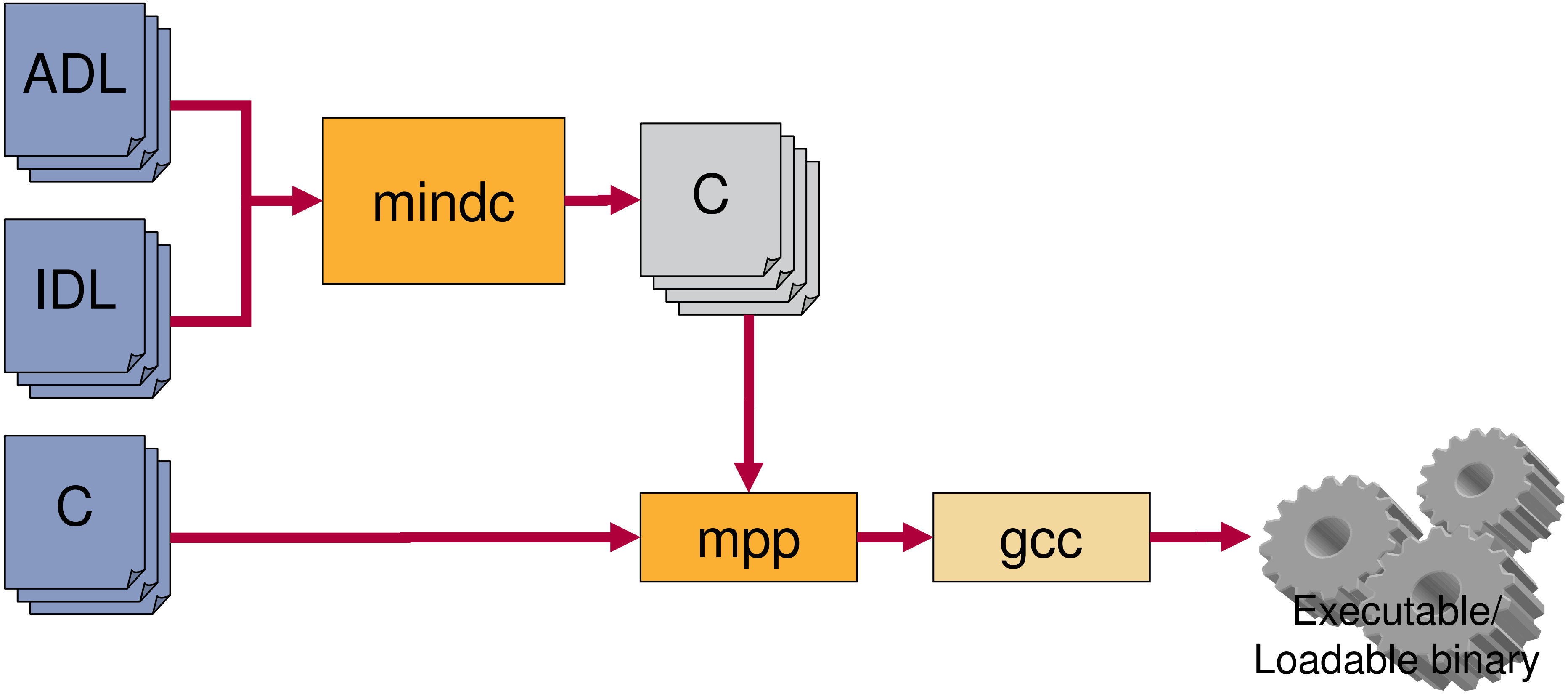
MIND deliveries

- Architecture compilation tool-chain (mindc)
  - Reads software architecture description, produces source-code and compiles binary
- Integrated Development Environment based on Eclipse
  - Graphical and textual editor for architecture description (GMF, XText)
  - Textual editor for interface description (XText)
  - Integration in CDT for the edition of implementation code and debugging
- Component libraries for the development of operating systems and middlewares
- Fully integrated “evaluation toolbox” for an easy and immediate technology hands-on
  - Full Eclipse installation,
  - C compiler and emulators (QEmu) for cross-compilation and emulation of ARM cores
  - Many examples that demonstrates the benefits of the technology.

mindc : Architecture Compilation tool-chain

- Reads ADL, IDL and implementation code and produces C glue-code
- Based on FractalADL
  - Reuses intermediate representation (AST) that uses ASM
  - Adapt its overall architecture
- Designed to be extensible
  - Itself a component-based application
  - Defines a plug-in mechanism

mindc Tool-chain



ADL : Architecture Description Language
IDL : Interface Definition Language
mindc : MIND ADL/IDL parser and C code generator
mpp : MIND PreProcessor

Architecture Description Language

- Non-XML syntax inspired by Think
- Three base constructs : “composite”, “primitive” and “type”
- Java-like naming
  - One definition per file
  - Directory layout that reflect package names
- Provides a strongly typed template system
  - A sub-component of a composite can be generic but must “conforms to” a type
- Includes an Annotation system directly inspired by Java

Architecture Description Language

```
type memory.AllocatorType {
  provides Allocator as alloc;
}

import memory.AllocatorType;
primitive unix.memory.Malloc extends AllocatorType {
  source malloc.c;
}

import memory.AllocatorType;
composite unix.GenericApplication <App conformsto ApplicationType,
                                   Alloc conformsto AllocatorType> {

  contains Boostrap as bootstrap;
  contains App as application;
  contains Alloc as allocator;

  binds bootstrap.entryPoint to application.main;
  binds application.alloc to allocator.alloc;
}
```

Interface Definition Language

- Defines type of component interfaces
- Based on C types
  - struct, enum, union, typedef
- Add a dedicated “interface” construct
- An interface can extends another one (one single inheritance)

Implementation language

- Based on C and adds dedicated macros and associated preprocessor
- Macros are used to capture component-based concept in C code
  - METH/CALL to declare/invoke component’s method
  - PRIVATE to declare and access component’s data
- The preprocessor translates these macros into C-Code
  - Supports two outputs :
    - Re-entrant code : add a “this” pointer to component’s methods; component’s data are accessed through this pointer. Allows the component to be instantiated several times.
    - Singleton code : component’s data are global variables. Useful to integrate legacy code : a normal C-code can access component’s data or invoke methods.

Integrated Development environment

- Based on Eclipse (3.6)
- Uses GMF and XText to provide a graphical and a textual editor for ADL
  - Graphical editor inspired by F4E
- Provides an IDL textual editor
- Extends CDT to support dedicated macros
- Integrates the mindc tool-chain for error reporting
- Provides component awareness in debugger
  - Break-points on interface
  - Abstracts the generated glue-code

OW2 infrastructure

- Source-code hosted in SVN
  - HTTP(s) access would really helps
- JIRA tracker
- Fisheye SVN viewer
- Bamboo continuous integration
- Web-pages generated from simple XML files
  - Use an adapted template
- Maven documentation and doc-book produce HTML pages with the same template as the MIND web-pages

Current status

- A functional mindc tool-chain is already available
  - version 0.1
  - version 0.2-alpha-1 will be released in the new few days
- A prototype version of the IDE is available

- The MIND project is already used for the development of software infrastructures and programming models in ST
  - P2012 program : CEA/ST R&D project for the development of an advanced many-core system-on-chip

Thank you.

Q & A