PHARAON

Parallel and Heterogeneous Architectures for Real-time ApplicatiONs

http://pharaon.di.ens.fr

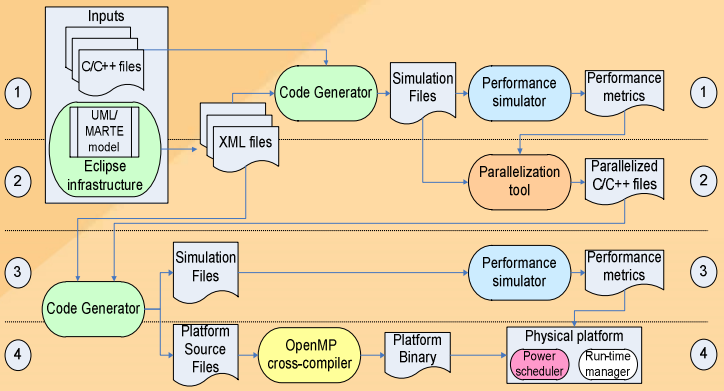
SEVENTH FRAMEWORK PROGRAMME

Objectives & Impact

Develop two sets of techniques and tools, aimed at exploiting low-power capabilities of embedded SoCs with heterogeneous CPU, DSP and GPU cores.

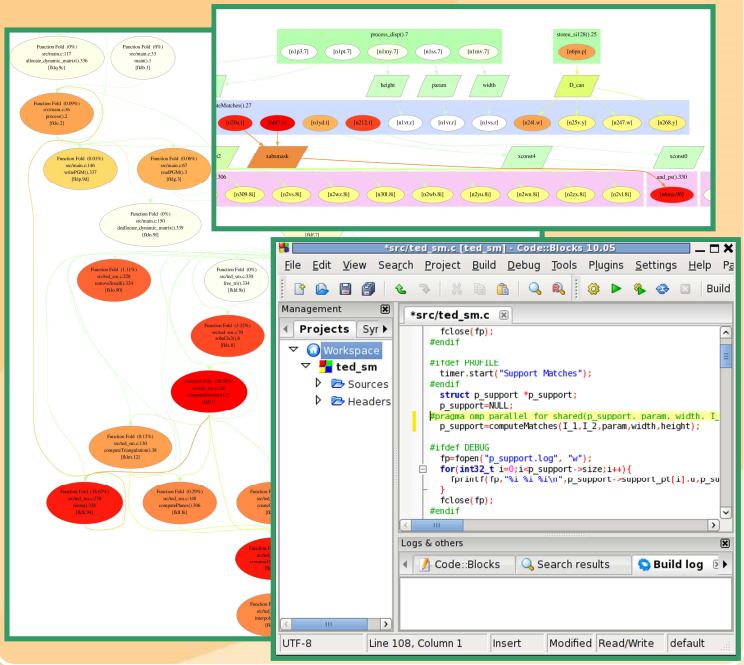
1. Find the most adequate software architecture taking into account hardware constraints.
   - analyze the parallel structure of an application
   - automatically generate multi-processor code.
2. Adapt the platform performance (e.g. frequency & voltage) to consume only the required energy.
   - run-time reconfiguration manager
   - low power scheduler.

Design Flow



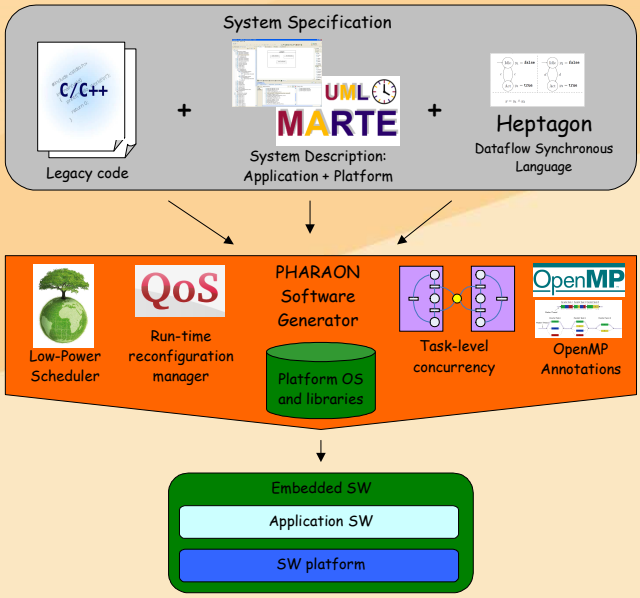
Software parallelization

The legacy software to be parallelized is analyzed to identify and display (in a highly compacted form) data dependencies and opportunities for parallelization.



Code Generator

The complete SW stack to be executed in each node is automatically generated from the UML/MARTE models and the functional code.
The generator produces optimized code, including additional code providing parallelism and run-time optimizations.



Demonstrators

❑Three demonstrators from two domains: radio and image processing, are being produced.

Two radio demonstrators:

1. MAC layer implemented on a multicore ARM based platform
2. Physical layer (L1) with real-time reconfiguration and multi-stream capabilities implemented on an ARM-based platform with a specialized DSP

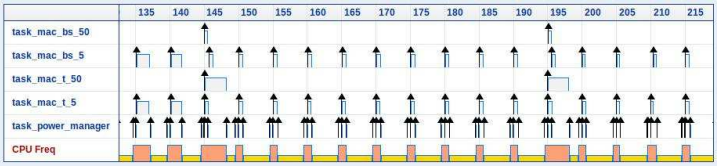
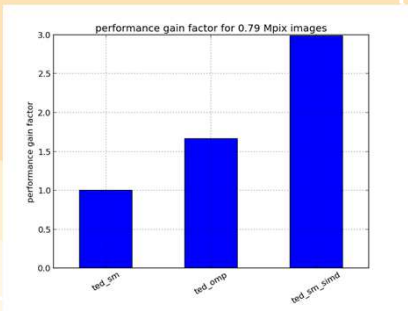


Image processing demonstrator:

3. Advanced 3D stereoscopic application with real-time and high definition constraints targeting the automotive domain for human and obstacle detection



THALES imec TEDESYS UC UNIVERSIDAD DE CANTABRIA Vector Fabrics

A revolutionary approach to fully automatic synthesis of embedded system software

eSSYN
embedded software synthesis

What is eSSYN?

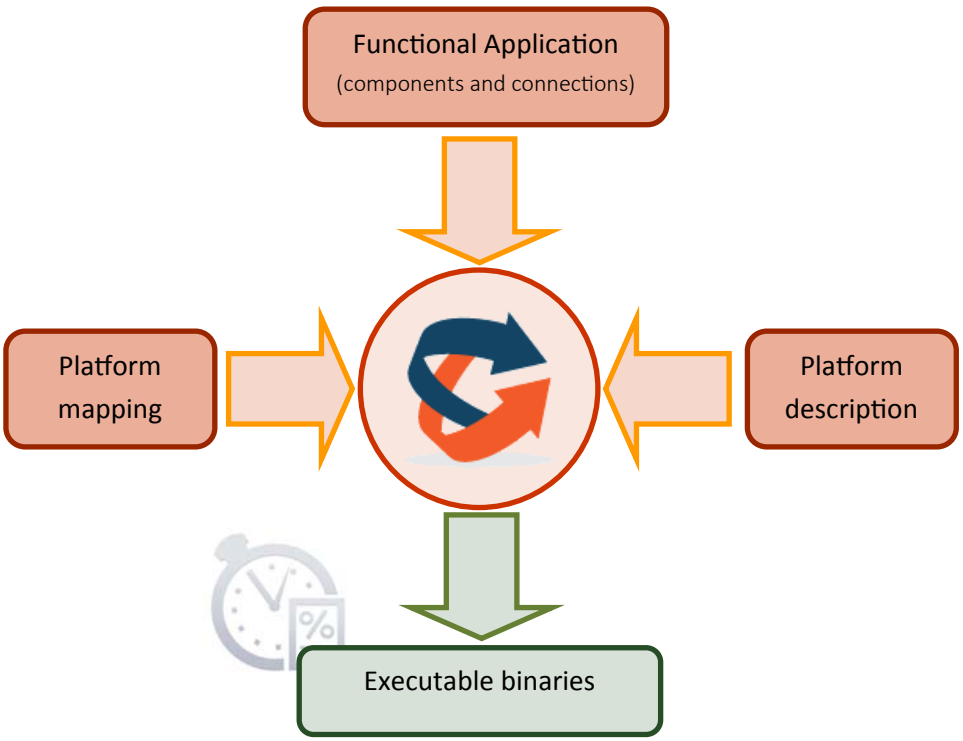
eSSYN is a software synthesis tool that automatically generates, platform specific executable binaries from a component based model of a software application and a simple model of the target hardware platform (supporting complex multicore heterogeneous platforms).

Who can benefit from eSSYN?

Anyone willing to boost the productivity of software design for embedded systems. Specially those involved in architectural design of embedded systems, including selection of target platform, mapping of software modules to available resources (GPP, DSP, GPU,...), code parallelization, application migration to new platforms, etc.

What is the main benefit of eSSYN?

eSSYN is an incredible effort saver for common system level tasks. For instance, splitting an application into two executables and mapping those to two cores of a platform can be done, from start to end (binaries) in five minutes. Doing that by hand is a matter of days.



UNIVERSIDAD DE CANTABRIA

Microelectronics Engineering Group

Universidad de Cantabria

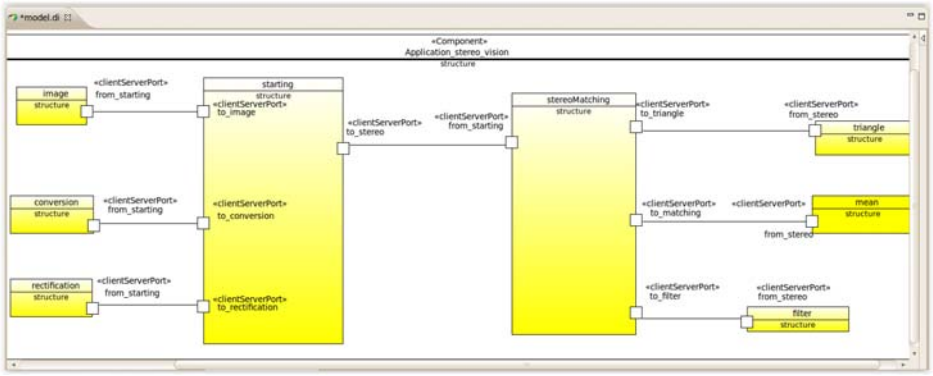
TEISA Department (GIM Group)
Avenida Los Castros
39005 Santander
Spain

email: essyn@teisa.unican.es

eSSYN
embedded software synthesis

Platform Independent Model

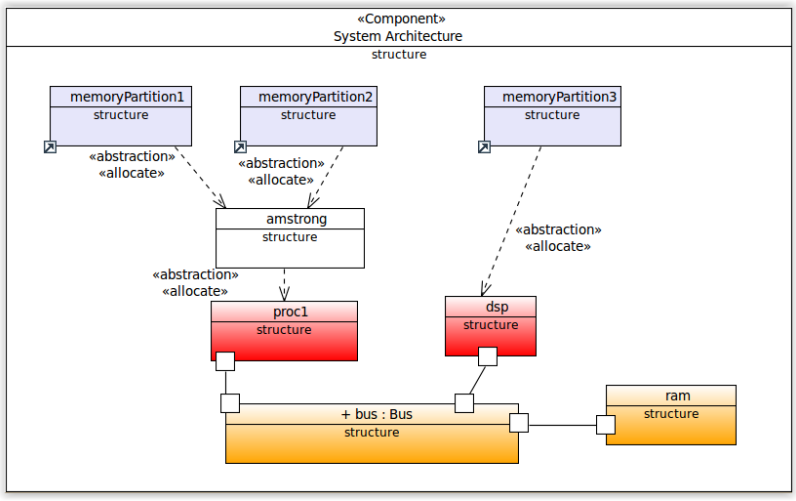
A component based model of the application is generated by the user using a simple semantic (even a wizard is available making it extremely easy to generate). This model is independent of the target HW platform and therefore is reusable among different platforms. As part of the component description, functional C, C++ or OpenCL source code is provided (actually several alternative implementations of the componet funcitionality may be included) and a description of the component interfaces. Last, a system view showing the connections among components complete the platform independent model of the application.



Platform Specific Model

eSSYN allows the user to play with different mappings of components into executables though a convenient GUI. Similarly each executable can be assigned to one of the microprocessor cores, DSP or GPU available in the hardware platform. With a simple click and drag the user can completely redefine the mapping of the application and in five minutes obtain new binaries for the new implantation, exploring this way different parallelisms, resource usages etc.

Equally powerful is the capability to map an application to different platforms in a similar way. eSSYN only needs a very simplified model for each platform to test, an upgraded System Architecture as shown below and with the click of a button a new binary is generated for the new platform.



Want to know more about eSSYN? Contact eSSYN team for a hands on demo and get to know the new way to generate complete embedded software systems in minutes.

eSSYN has been funded and developed under EC Project FP7 288307 PHARAON

More information at:

http://www.teisa.unican.es/gim/es/proyecto?id=95

http://pharaon.di.ens.fr/

email: essyn@teisa.unican.es

eSSYN
embedded software synthesis

PHARAON

SOFTWARE SYNTHESIS FOR HETEREOGENEOUS EMBEDDED SYSTEMS

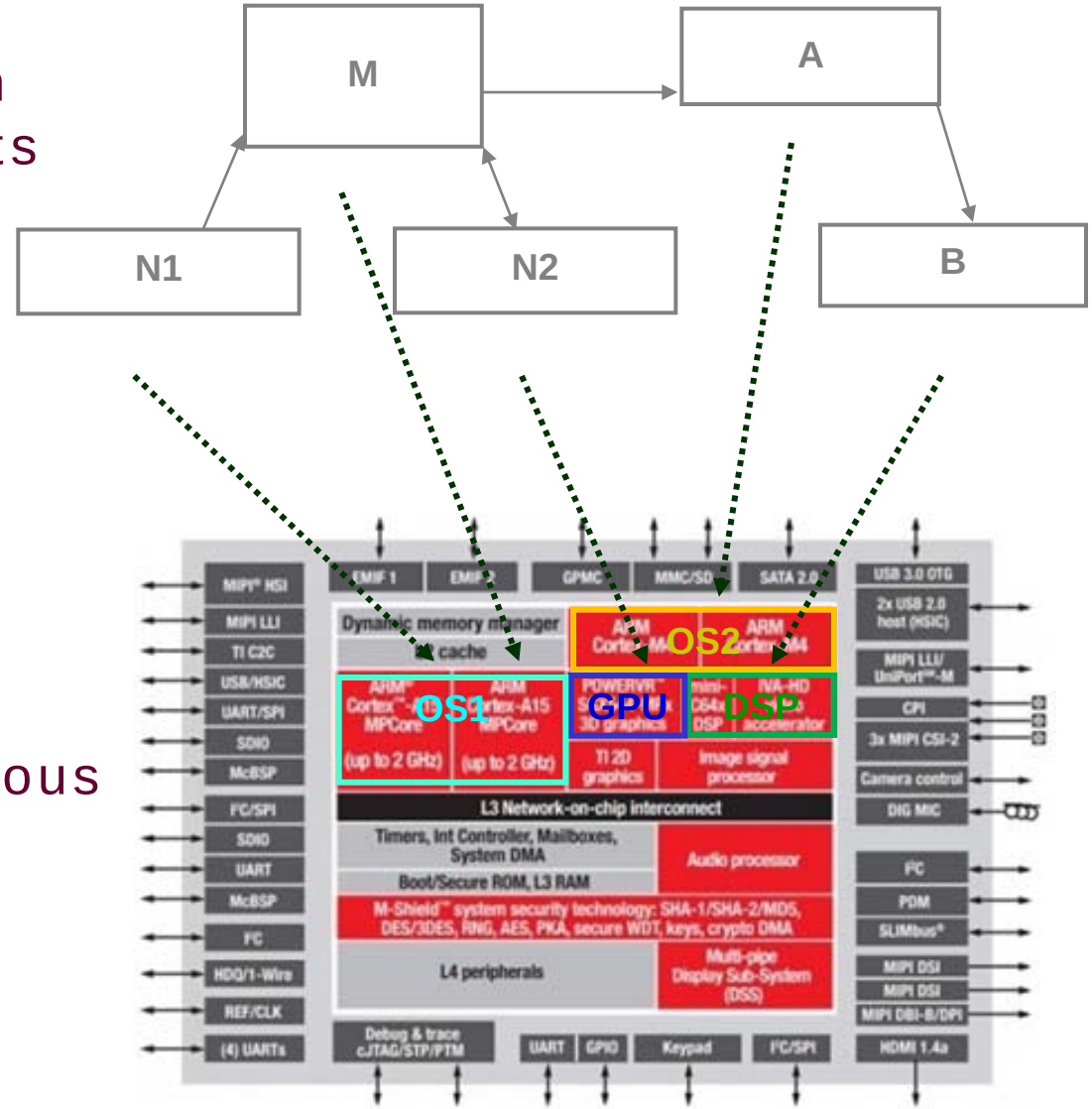
University of Cantabria

PHARAON project number FP7-288307

PHARAON

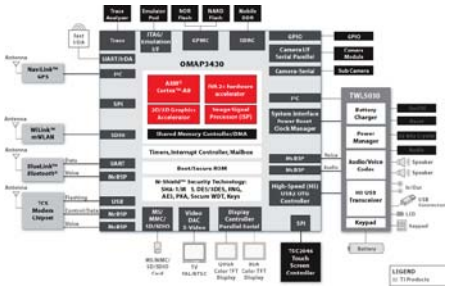
Starting point

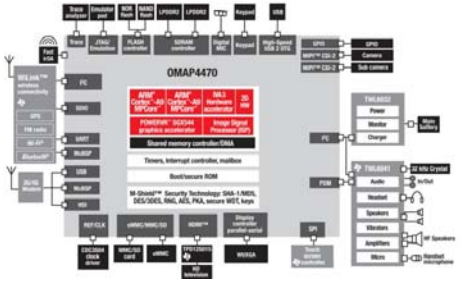
- Application Components
- Mapping
- Heterogeneous platform

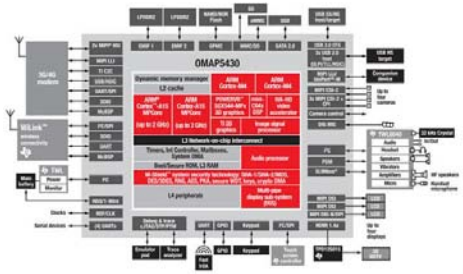


PHARAON

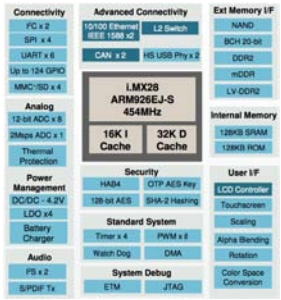
Multiprocessing and heterogeneous platforms

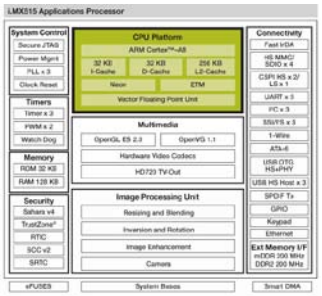


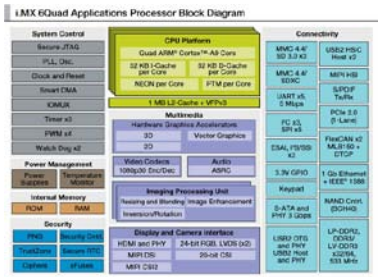




Texas Instruments OMAP Evolution Time



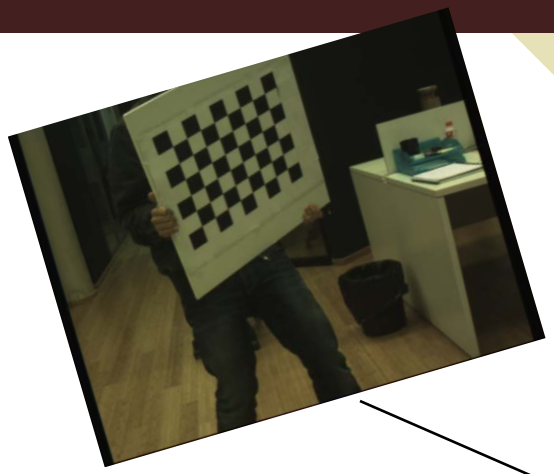




Freescale IMX Evolution Time

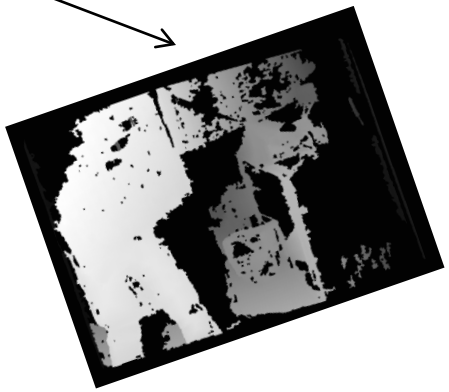
PHARAON

Stereovision algorithm



Stereovision Algorithm
Starting, image, conversion, rectification, stereo-matching, triangle, mean, filter

Disparity map

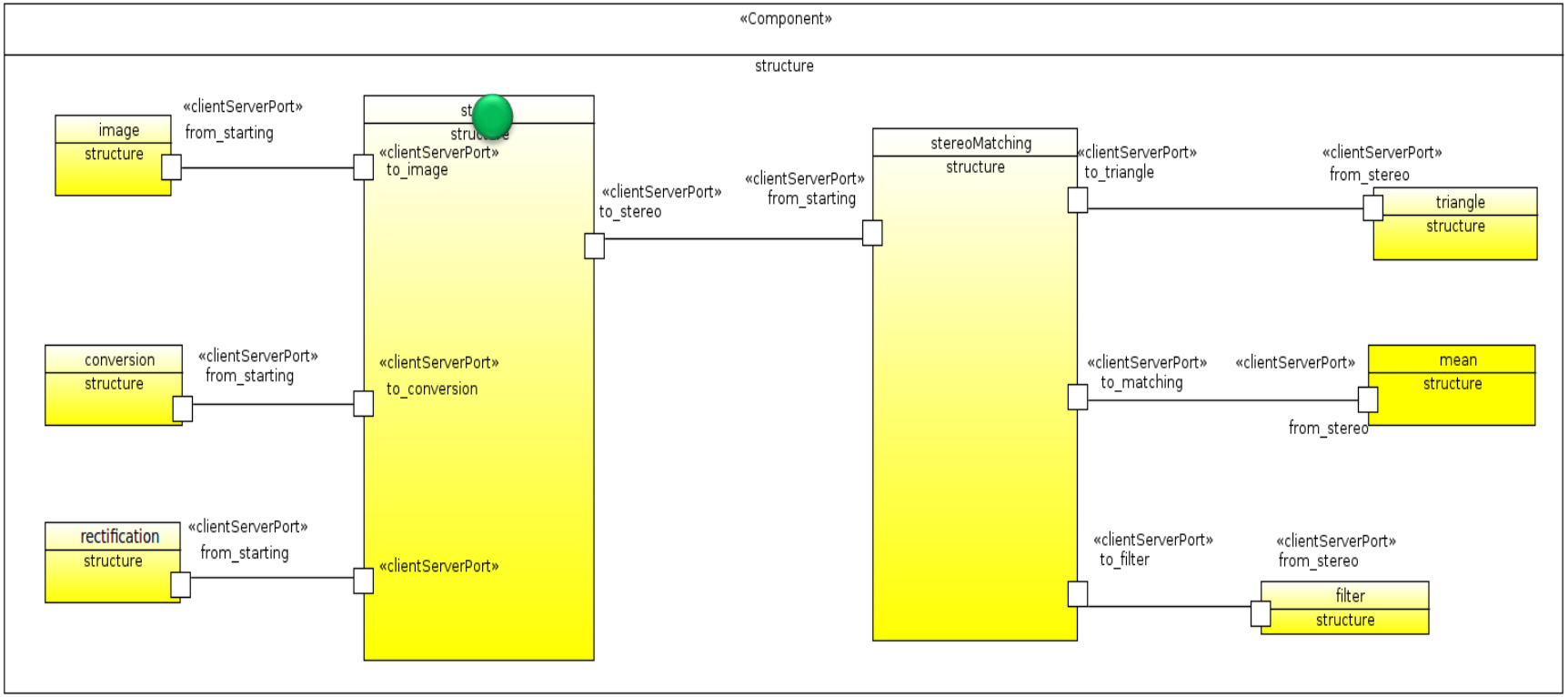


PHARAON

Concepts

- **Platform-independent model (PIM)**
  - Reusability
- **Memory partition → Executable**
  - Data protection, communications, multi-critic…
- **Heterogeneous platforms**
  - CPUs + DSP/GPU

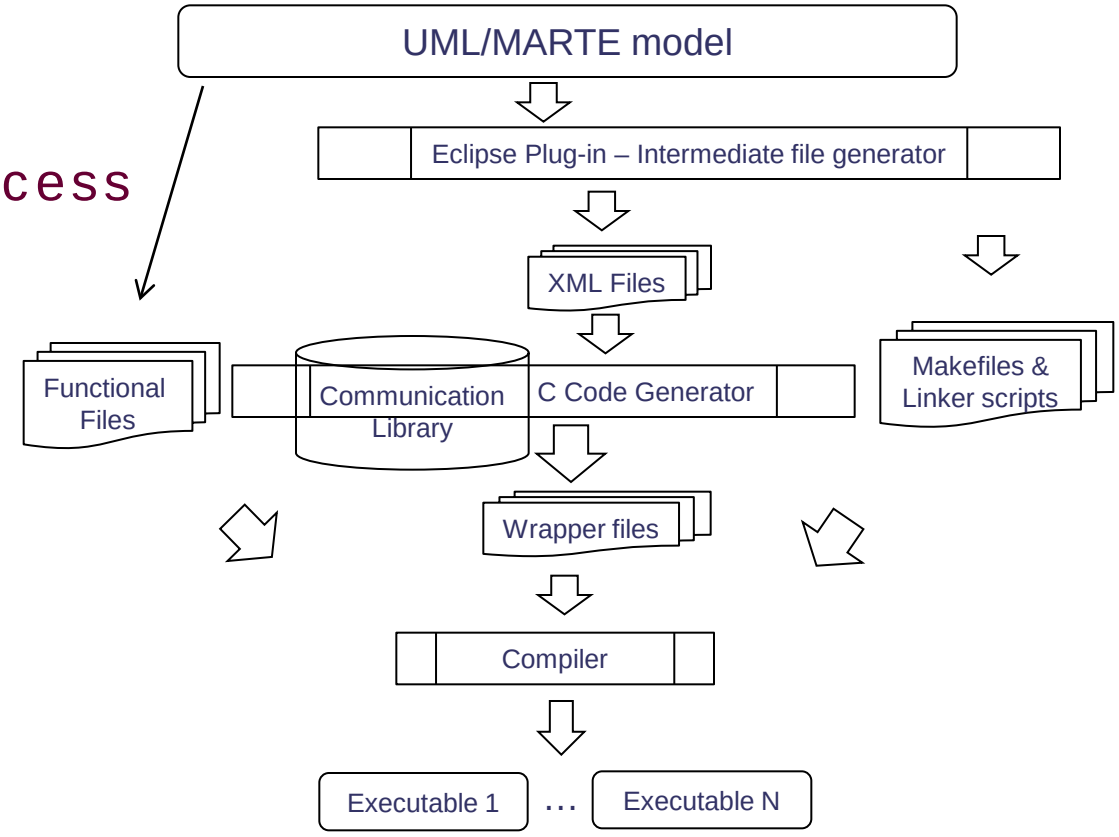
Stereovision Execution Flow



PHARAON

Tool flow

- UML/MARTE modeling
  - **PIM-PDM-PSM**
- SW synthesis process
  - **XML**
  - **Software synthesis**
  - **Makefiles**
  - **Compilation**



PHARAON

Advantages

- Designing at abstraction level (MDA)
- Project Management
- Reusability
  - **Platform-independent code**
- Parallelism
- Reduces in-depth knowledge of platforms
  - **Reduce re-engineering effort**

Tool: 1 min effort time
vs.
Manually: 2-3 days effort time