



# MORPHEUS

---

## Abstract of D2.3.1: Toolset modules report

CONTRACT NO	MORPHEUS IST 027342
TYPE OF DOCUMENT	Publishable abstract of D2.3.1
DATE	15/05/2007
ABSTRACT	This document is the abstract of the D2.3.1 It is available on the MORPHEUS public website
AUTHOR, COMPANY CONTRIBUTORS:	Philippe Bonnot, (TRT , deliverable leader) Elena Moscu Panainte, TUD Florian Thoma, UK Nikos Voros, ICOM Bernard Pottier, Lagadec Loic, Boukhobza Jalil, Alain Plantec, Jean- Christophe Le Lann, UBO Richard Taylor, Critical Blue Arnaud Grasset, Philippe Millet, TRT
WORKPACKAGE	WP2
CONFIDENTIALITY LEVEL	RE
FILING CODE	MORPHEUS-TRT-D2.3.1-R0.3



The D2.3.1 deliverable document describes MORPHEUS project's toolset modules detailed information, complementing the "Toolset specification" (deliverable D2.1.1). Regarding compilation, the two compiler optimisations for reducing the reconfiguration overhead are detailed (intra-procedural and inter-procedural scheduling). The optimisations have been designed, implemented and tested in the CoSy compiler. The PowerPC compiler backend for the CoSy compiler is presented in details. The Configuration Call Graph implemented to provides the RTOS and Configuration Manager with an overall picture of the application behaviour.

Regarding RTOS aspects, the eCos operating system is the base for the other tools. A dynamic memory allocator provides fragmentation optimised memory pools. A service for reconfigurable computing manages the configuration and execution of operations on the heterogeneous reconfigurable engines. The network-on-chip and the DMA are managed through a joint interface to make their difference transparent to the spatial design tools. Regarding Spatial design aspects (memory-centric mapping and synthesis), the definition of the internal structure of computations and the relative mapping techniques for different reconfigurable unit (HRE) architectures is achieved using a Control Data Flow Graph (CDFG) format that can be easily generated from high level language compilers. The CDFG address concurrency as an open way to define process composition, and as fine grain parallel constructs. It also embeds memory accesses. A complete application methodology based on the SPEAR framework where processes are composed by connectors including memory addressing, and behaviours is detailed. Process behaviours are defined in a subset of C and processed by a code analyzer from Critical Blue (Cascade). The document describes how a CDFG is processed to produce an hardware description for an accelerator, and how it can be interpreted during development phases. Finally it describes how memory transfers are achieved in relation with the hardware platform and sequencing requirement of the applications.

The first phase integration of the modules presented here is on-going. The results of this integration will be presented on a future deliverable. A refinement of the Toolset definition will then be performed thanks to the results obtained in the first phase.

