Information and Communication Technologies (ICT) Programme

Project Nº: H2020-ICT-2016-1-732105

D6.6: *CERBERO Toolchain: Lesson Learned, Gap analysis and Development Roadmap*

Lead Beneficiary:	TASE
Workpackage:	WP6
Date:	31-12-2019
Distribution - Confidentiality:	Public

Abstract:
This document summarizes the lesson learned while using the CERBERO toolchain in the use cases and presents a roadmap for future improvement and development. If, on the one side, the most mature features of the CERBERO tool chains have been fundamental for the development of the use case, on the other, newly added features needs to consolidate to arrive at a full marketable version and, from the experience of using the tool chain in

complex use case, we identify few gaps between the desiderata of the users and the current version of the tool. This indeed is an important feedback that needs to be considered for future development of each tool and of the whole toolchain. For each of these desiderata, we assess the feasibility and we propose a development plan are proposed and framed into an overall CERBERO roadmap to market.

The CERBERO Consortium is the following:

Num.	Beneficiary name	Acronym	Country
1 (Coord.)	IBM Israel – Science and Technology LTD	IBM	IL
2	Università degli Studi di Sassari	UniSS	IT
3	Thales Alenia Space Espana, SA	TASE	ES
4	Università degli Studi di Cagliari	UniCA	IT
5	Institut National des Sciences Appliquees de Rennes	INSA	FR
6	Universidad Politécnica de Madrid	UPM	ES
7	Università della Svizzera italiana	USI	CH
8	Abinsula SRL	AI	IT
9	Ambiesense LTD	AS	UK
10	Nederlandse Organisatie Voor Toegepast Natuurwetenschappelijk Ondeerzoek TNO	TNO	NL
11	Science and Technology	S&T	NL
12	Centro Ricerche FIAT	CRF	IT

For the CERBERO Consortium, please see the http://cerbero-h2020.eu web-site.

Document Authors

The following list of authors reflects the major contribution to the writing of the document.

Name(s)	**Organization Acronym**
Pablo Sánchez de Rojas	TASE
Joost Adriaanse	TNO
Bas Pijnacker Hordijk	S&T
Antonella Toffetti	CRF
Stuart Watt, Ayse Goker, Hans Myrhaug, Leszek Kaliciak	AS
Karol Desnos	INSA
Francesca Palumbo	UNISS
Tiziana Fanni	UNISS
Carlo Sau	UNICA
Francesco Regazzoni	USI
Eduardo de la Torre	UPM
Evgeny Shindin	IBM

Document Revision History

Date	**Ver.**	**Contributor (Beneficiary)**	**Summary of main changes**
21-10-2019	0.1	TASE	First draft
20-11-2019	0.2	AS	First AS input created
6-12-2019	0.3	TNO	First TNO input created
9-12-2019	0.4	AS	AS input updated
17-12-2019	0.5	TNO	Update

17-12-2019	0.6	UNISS	UNISS contribution to roadmap and general review of the document
18-12-2019	0.7	UNICA	Update
19-12-2019	0.8	USI	General document revision
15-01-2020	0.9	TASE	Final update
15-02-2020	1.0	IBM	Changes in AOW sections

Table of contents

1. Executive Summary

This document presents a road map towards the consolidation of the CERBERO tool chain. Starting from the lesson learned while developing the use case, we report the gap still existing between the current version of the tool and the desiderata of the tool users. Among the desiderata, we identified useful functionalities to be included in future releases of the CERBERO tool chain and we propose a road map towards this development. This deliverable complements D6.5, that reports the performance of the tool functionalities and the way they have been used to boost the implementation of the use case.

1.1. Structure of Document

This document is organized as follows:

- Section 2 described the lesson learned from UC provides, analyzes the gaps still existing between the current version of the tool and the desiderata from the use case providers, and report a list of useful features to be added to individual tools;
- Section 3 specifies the development roadmap of the individual tools developed from tool providers starting from the suggestion of UC providers;
- Section 4 contains the overall conclusions.

1.2. Related Documents

This document is related to the following deliverables:

- D2.1 - Description of scenarios (Final version)
- D2.2 - Technical Requirements (Final version)
- D6.1 - Demonstration Skeleton (Final version)
- D6.2 – Planetary Exploration Demonstrator (Final version)
- D6.3 – Ocean Monitoring Demonstrator (Final version)
- D6.4 – Smart Travelling Demonstrator (Final version)
- D6.5 - CERBERO Performance report

1.3. Related CERBERO Requirements

Deliverable D2.2 defines the final list of CERBERO Technical Requirements (CTRs) the project should achieve. Each of them is referenced with a unique identifier ranging from 0001 to 0020.

The current document address 2 CTRs as described in the Table 1. It is important to note that other requirements related to this deliverable are covered in the use case demonstrators and will not be included here. The demonstrators were used to validate fulfillment of these technical requirements using the developed methodologies and functionalities available in the CERBERO toolchain (being per-existing components or newly developed ones). During the development of the demonstrators further possibilities for improving the functionalities of the tool chain have been identified (e.g. functionalities that could further improve the design phase and the implementation of the demonstrators).

Table 1 - List of CTR addressed by the D6.6

CTR id	CTR Description	Link with the D6.6 document on CERBERO Toolchain: Lesson Learned, Gap Analysis and Development Roadmap
0015	CERBERO SHALL provide review reports including: • intermediate results, • technical risks evaluation, • plan vs actual effort, • financial risk evaluation.	This document reports the lessons learned, lists the gaps identified in the tools during the projects and presents a possible development roadmap for these technologies.
0017	CERBERO Use-Case providers SHOULD check and provide timely feedback on the usability of CERBERO tools and framework.	Missing points and desirable features and other issues related to usability of CERBERO tools is provided in Section 2 of this deliverable.

2. Gap Analysis

The CERBERO toolchain includes a collection of tools, some of them have been extended an improved during the project, while some others have been newly developed. Each of the tool specifically addresses one or few aspects of the whole system design and management. The tools are complementary, so that the overall tool chain provides a complete environment for CPSs design. These tools operate at different levels of abstraction, from the computation level to the System level, delivering also user-level features (i.e. requirements and model verification). CERBERO tools offer support at design-time, for the design, verification and deployment of self-adaptive CPS, and at run-time for providing self-aware smart system management (see D5.1 for more details about the different levels).

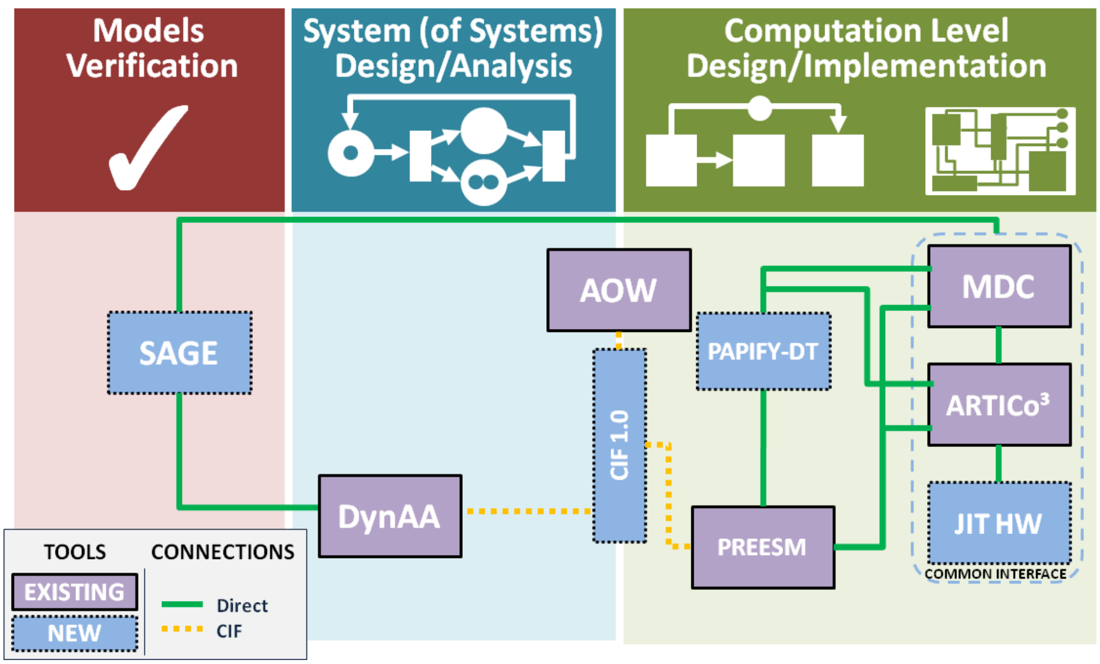

Figure 1 - CERBERO Tools - Design-Time Support

Figure 1 depicts the design time view of the tool chain and specify the connection between different modules. The one adopted in the UC demonstrators are listed hereafter per level of abstraction.

The tools used at *System Level are:*

- AOW that solves large scale hybrid optimization problems to return frontier of Pareto optimal solutions
- DynAA is based on discrete-component models to find optimal solutions by means of model simulations.

At the *Computation Level* the tools used for design and implementation, are:

- PREESM enables parallel-application development with design-time prediction, as well as code generation and re-use capabilities.

- PAPIFY provides monitoring capabilities by means of an event library aimed at generalizing PAPI for heterogeneous architectures.
- MDC tool is an automated dataflow-to-hardware framework for the generation and system integration of CGR accelerators.
- ARTICo^3 exploits a DPR-enabled multi-accelerator computing scheme, to provide scalable parallelism. It also provides an automated toolchain to go from the user-defined application down to the system implementation.

The need of some design and monitoring functionalities have emerged during the project. As a result, some of the tools composing the CERBERO toolchain have been newly developed during the project. Naturally, their level of maturity has not been comparable with other more established components of the toolchain that have been used in the development of the UC demonstrator (demonstrators integration activities started before M18, while new tools were still under development). Nevertheless, the potentiality of the new components of the CERBERO toolchain have been demonstrated as stand-alone proof of concepts, which consisted in stand-alone proof of concepts and the feedback on their improvement is mostly based on this.

At *Model/Verification Level* we have the SAGE Verification Suite that leverages on formal methods and provides two tools:

- ReqV, that allows automatically check consistency of a set of requirements provided by the user
- Hydra, to perform automated synthesis of high-level policies[1].

At the *Computation Level*, the toolchain includes:

- JIT, hardware composition refers to the ability to implement, at run-time, hardware accelerators on FPGAs without a pre-synthesized design. IMPRESS, belonging to the JIT HW design suite, is a TCL script-based tool for the automated generation of relocatable partial bitstreams under Vivado. The JIT neuroevolvable HW that uses block-based neural networks will be incorporated as a use case of IMPRESS, and available in the IMPRESS git repository.

[1] https://gitlab.sagelab.it/sage/hydra

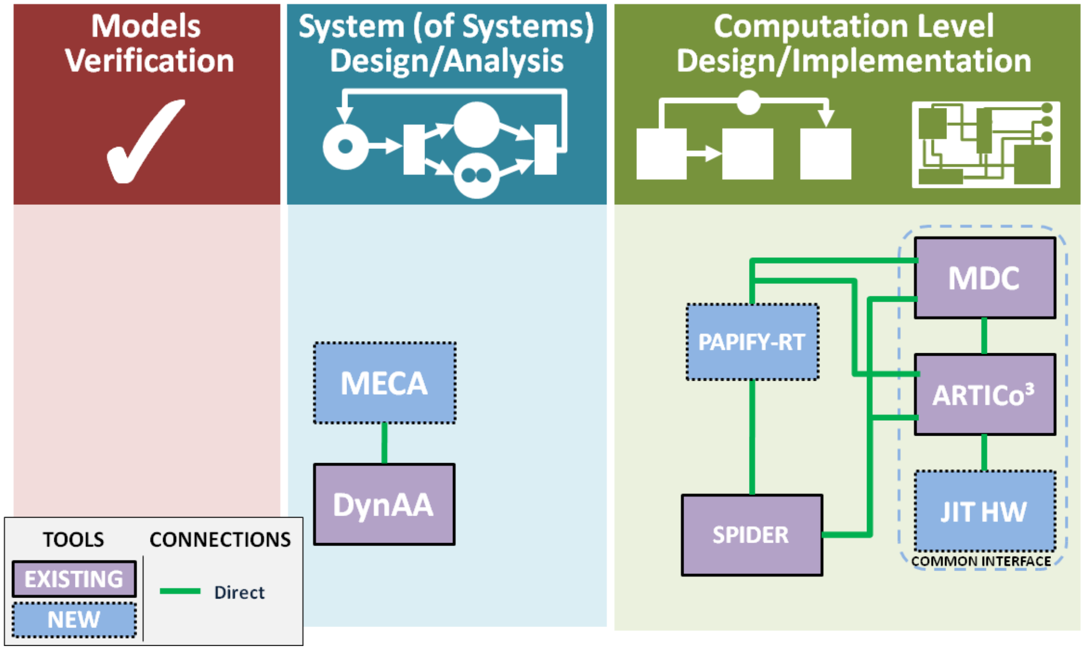

Figure 2 - CERBERO Tools - Runtime Support.

Figure 2 depicts the tools at run-time, and their connection. The one adopted in the UC demonstrators are listed hereafter per level of abstraction.

At *System Level*, we have

- MECA improves the resilience of human-machine teams by providing system, environmental and human monitoring and diagnosis, and high-level decision support in cases of unforeseen conditions and events,
- DynAA explores different solutions at run-time to provide direct interaction with signals that come from the system and the environment.

At the *Computation Level* the tools for run-time support are:

- SPIDER performs dynamic mapping and scheduling of reconfigurable dataflow applications on parallel heterogeneous architectures.
- PAPIFY is meant to provide a large set of run-time execution information to SPIDER.
- MDC and ARTICo^3 deploy and configure proper engines over the physical substrate at design-time. These engines are used at run-time to execute all the actions needed to support run-time reconfiguration of the HW.

Similarly to design-time tools, also some of the run-time tool have been newly developed in the CERBERO project. The evaluation of these tools is carried out, also in this case, mostly as standalone proof of concept.

At the *Computation Level* we have:

- JIT HW - composition addresses fine-grain reconfiguration, providing a way to map circuits at run-time by composing small HW components laid on an overlay architecture. IMPRESS allows reconfigurable module composition to generate custom overlays on the fly, without the need of a predefined floor planning and inter-module communication description.

Table 2 summarizes the gap analysis information for the different tools presented in the following subsections of the document.

Table 2 - Gap Analysis Summary

Tool	Issue	Urgency	Motivation
DynAA	Extended model support	Medium	DynAA builds on a growing library of modules. Increasing the library with useful and realistic simulation models enhances the overall applicability of DynAA.
DynAA	Parallel execution support	Medium	DynAA has a scalable simulation core, however extreme large models can lead to slow simulation execution. Parallel execution allows to speed up simulation runs and design space exploration when simulating several parameter configurations.
DynAA	Increase user friendliness	Medium	DynAA has a coding interface. A graphical interface could facilitate the description of simple systems and provide a natural way to visualize simulation data.
DynAA	Refactor exception systems	Low	The native exceptions provided in the DynAA simulation core and models can include more information to ease debugging and maintenance.
MECA	Improve user friendliness	Medium	Addressing this will reduce the time required for tailoring MECA to future projects. However, it will not be limiting core functions if left unsolved.
MECA	Improve scalability of design space	Medium	Even though a more scalable design space will improve

			performance, computation times can also be reduced by running multiple distributed DynAA instances.
MECA	Self-adapting user model	Low	It is highly project-dependent if this will be a core feature. For most it will be a nice-to-have only.
CIF	Complex merge operation	High	With complex merge operations, CIF will be able to combine two existing model views (e.g. from different tools) into one.
PREESM	Comprehensive continuous integration	High	Continuous Integration (CI) is already used in the PREESM Project and will be extended to apps and GUI.
PREESM	Moldable parameters support	Low	Feature extension.
PREESM	Reference documentation	Medium	To complement existing tutorials and support for advanced tool usage.
SPIDER	Extended tutorials and documentation	High	To spread the use of the run-time.
SPIDER	Public continuous integration	Medium	There is currently local CI, it will be opened to public visibility.
PAPIFY	Extended tutorials and documentation	High	To spread the use of monitoring tools.
PAPIFY	Compatibility with SPIDER 2.0	Medium	It is currently compatible to SPIDER 1.0 and the work already done would help on the integration.
MDC	Command prompt version	Medium	Maximizing usability for MDC is important. Therefore, there is a plan to work both on improving user interface and providing a command prompt version.
MDC	Improve usability through OpenCL	High	Already planned extension with top priority.
ARTICo^3	Complexity of usage	Medium	High Level Synthesis flow requires knowledge on how to efficiently implement accelerators that may profit

			from the scalability achievable in ARTICo3. More examples will be provided in the framework repository.
ARTICo^3	Extension to other COTS devices	Medium	Despite the tutorial on how to achieve this, more built-in porting examples to other families/devices will be incorporated.
ARTICo^3	Production of relocatable bitstreams within ARTICo^3 toolflow	High	Ongoing activity. This is a very interesting feature that will enrich the framework.

2.1. DynAA

It has been extensively used and extended to support the Smart Travelling demonstrator and for the design of the Ocean Monitoring demonstrator. During the development of these use case, we identified as potential improvements the followings:

- Add the support for configuring different models for system in the loop simulations.
- Increase the support for parallel execution. Parallelism is a new feature included in DynAA during the CERBERO project. In next releases of DynAA it would be important to improve such support by including functionality to simplify parallelism and to monitor the execution of distributed simulations (now only master is able to visualize simulations).
- Increase the user friendliness of the tool by including a graphic user interface to support the easy creation and configuration of simulation modules on top of the DynAA core. The initial DynAA software was based on MatLab and equipped with a graphical interface. Because the DynAA core was reimplemented in Java to better support the system in the loop functionality and scalability, the initial graphical interface could no longer be used.
- Allow the simulation engine to be used as instances as well as singletons. This will make it easier to use multiple models inside a single JVM.
- Refactor the DynAA exception systems to use a ModelException and a ModelRuntimeException, so that exception handlers don’t need to enumerate all possible exceptions and are better forwards-compatible.

Overall, the use of DynAA required the direct support from the developing partner. This fact was not representing a main limitation, since the developers of the tool and its users were actively collaborating. However, prior to release the tool openly to the community, it would be necessary to produce a more detailed and self-contained documentation and a number of examples to allow new users to get familiar with the tool and use it independently.

2.2. MECA

MECA has been used as the decision support system in the Smart Travelling use case fulfilling this need. Also, it has shown to be well usable in conjunction with DynAA. Among possible improvements there have been identified:

- The user-friendliness of MECA can be improved for both system designers and operators. Several new routines are Smart Travelling specific, for instance to improve logging and configurability. Ideally, these would be made more well-defined and accessible to developers. This would improve debugging capabilities and facilitate faster development cycles for new use cases, and improve the system understanding for operators.
- The current interaction between MECA and DynAA requires to exhaustively explore the design space of itineraries. All route options identified by the Open Source Routing Machine (OSRM), along with all feasible combinations with charging points, must be evaluated by DynAA. When several charging point options are present along the way, this becomes an inefficient process that does not scale well. Removing clearly infeasible itineraries by MECA before evaluation by DynAA can improve the scalability of this process substantially.
- As in the case of DynAA, the current user preference model of MECA is static. Also, in this case, a self-adapting model would allow MECA to learn from previous choices and provide improved decision support on future decisions.

2.3. CIF

A tool to simplify interfacing between different tools within the CERBERO toolchain (e.g., to share data on the same simulation models) was still missing in the beginning of the project. During the project a first version of the CERBERO Intermediate Framework (CIF) tool was developed to share data between different tools. Since the problem to be solved was very complex, only a subset of possible data transformations has been addressed and implemented. The implemented CIF tool currently supports merge operation for class equivalence and property equivalence between data sets of two tools. Complex merge operation, which would involve merging different models from different tools at the same time were out of the scope of this project but are definitely worth to be explored in the future. This means that CIF can be used to transfer the data set of one tool to the data set of another tool.

2.4. AOW

AOW remains closed source. The main reason for this is that in the current development stage AOW is coupled with CPLEX optimization solver, so it is impossible to provide AOW code without CPLEX. While CPLEX is free for academic usage commercial license is expensive. Moreover, in general, CPLEX requires substantial amount of computational resources to run, making AOW design-time tool. Thus, during the Ocean Monitoring use case, it was possible to do design time work with AOW by handing over data and parameters to receive suggestions for optimization, however, it was unavailable for run-time use. This means that run-time optimization would require alternative tools and algorithms.

2.5. PREESM

The dependency on Eclipse to provide a graphical interface was a challenge. A more clean differentiation between the core of the tool and its graphical interface, leaving the user also the possibility to use the tool engine from the command prompt, has emerged as a desirable feature.

Other identified improvements in the context of PREESM are:

- Comprehensive Continuous Integration: Although it was completely re-designed and improve during CERBERO, the continuous integration of PREESM, built using travis-ci.org, still lacks coverage on some features of the tool. Notably, because of time limitation of builds on the used CI service, open-source PREESM projects are currently not tested by the CI.
- Support for moldable parameters of dataflow models of computation remains to be implemented within PREESM. These parameters will enable the DSE engine to explore more functional and non-functional design choices, thus offering more design trade-offs to the system designers.
- Since PREESM is already available as open source, documentation is extremely important. In addition to the tutorials that "teach" the user how to use the PREESM tool to model, parallelize and optimize an application on multicore targets, a reference documentation listing all the features of PREESM should be created. Currently, this documentation exists for workflow tasks that are part of the design-space exploration process of PREESM[2].

2.6. SPIDER

Identified gaps in the context of SPIDER:

- The tutorial for SPIDER only covers the basic usage of the run-time. Comprehensive tutorial and documentations on its usage with HW/SW targets, notably through its connection to ARTICo[3] and MDC remains to be made available online.
- Public continuous integration. Although continuous integration of SPIDER exists on private servers of INSA, this process should be made available online to improve the trustability of the run-time for third-party users.

2.7. PAPIFY

PAPIFY has been used to instrument the Planetary Exploration computing platform and access the performance monitor counters at execution time. Identified improvements for this tool include:

[2] https://preesm.github.io/docs/workflowtasksref/

- Extended documentation and tutorials on how to monitor ARTICo^3 accelerator slots, and how to feedback the PMCs information to SPIDER in order to refine workload distribution in the run-time would be desirable in order to improve usability of the tool.
- A new, stable version of SPIDER is soon-to-be-launched. Therefore, compatibility with PAPIFY is not guaranteed and it will be necessary to perform further integration activities between the two tools.

2.8. ARTICo^3

ARTICo^3 has been used to provide dynamic reconfiguration capabilities to the computing platform in the Planetary Exploration use case. Possible improvements are listed below:

- Despite targeting "system designers with no prior knowledge/expertise in hardware development", this tool was still a challenge for those with that skill set.
- Integration with COTS-based designs would also be dramatically improved by support for USB-based FPGA components.

2.9. MDC

MDC is developed as a set of Eclipse plugins, mainly written in Java language. MDC user executables have been released as graphical tools, still based on Eclipse environment[3], offering a graph-based editor for modeling applications and an extremely powerful text editor for modeling functional units (actors) within such applications. Even if based on Eclipse, MDC exploits only a limited set of features of such IDE. However, similarly to PREESM, the dependency on Eclipse increases the effort when getting started, for teams not using Eclipse as a development environment. This issue is mitigated by the tutorials, and related material, that have been provided on-site and on-line in different occasions:

- CPS Summer School 2019[4],
- CPS & IoT Summer School 2019[5],
- CPS Summer School 2018[6],
- CPS Summer School 2017[7]

Despite this, similarly to PREESM, a clear separation between the graphical interface and the core of the tool, allowing the invocation of the tool engine also from the command prompt has emerged as a desirable feature.

MDC is meant to help digital designers in building a multi-purpose accelerator, by providing user friendly high-level graph-based and text editors. Following an approach widely accepted in literature, by means of High-Level Synthesis tools, particularly suitable to people with limited expertise in hardware development, the whole generation of

[3] https://sites.unica.it/rpct/download/
[4] http://www.cpsschool.eu/tutorial-cerbero/
[5] https://sites.unica.it/rpct/download/
[6] http://www.cpsschool.eu/previous-editions/cps-summer-school-2018/schedule/multi-grain-reconfiguration-advanced-adaptivity-cyber-physical-systems-2/
[7] http://www.cpsschool.eu/wp-content/uploads/2017/10/Tutorial-Fanni-MDC.pdf

reconfigurable hardware accelerators by MDC can be abstracted, so that users have only to deal with high-level specification languages, such as C and XML. Optionally, it is possible to take decisions on lower level aspects of the MDC functionalities, such as the usage of DMA or the kind of coupling between the host processor and the hardware accelerator. Some users, from the software programming domain, reported that links with OpenCL may improve the usability of the tool, making easier to delegate computing tasks to custom MDC compliant accelerators.

3. Development Roadmap of the CERBERO Framework Components

Starting from the gap between the user desiderata and the current version of the tools, CERBERO tool providers have envisioned a development roadmap. This section is meant to discuss it, providing the plans for improvements and the roadmaps that the different developed and evaluated tools will follow. Table 3 recaps on the Technology Readiness Level (TRL) of the different framework components.

Table 3 - CERBERO Framework Components TRL evolution

CERBERO Framework Components	Lead Partner(s)	TRL@M0	TRL@M36
SAGE Verification Suite (SAGE-VS)	UniSS	0/1	4
MECA core framework	S&T	3	6
DynAA simulation tool	TNO	5	7
Architecture Optimization Workbench (AOW)	IBM	2/3	4/5
PREESM	INSA	3	5
SPIDER	INSA	2	4
PAPIFY/ PAPIFY-VIEWER	UPM	3	5
Multi-Dataflow Composer (MDC)	UniSS, UniCA	2/3	4
ARTICo^3	UPM	3	5
Just-In-Time HW Composition (JIT-HW)	UPM	1	2/3
CERBERO Interoperability Framework (CIF)	IBM, TNO, AI, INSA	0	5/6

TRL is on average low, but for some framework components coming from industrial partners, as IBM and S&T, or from TNO, a Research and Technology Organization that closely cooperate with industries and a wide array of public actors. All the other CERBERO framework components are settled around lower TRL, which is normally the case for academic tools. Moreover, it must be noted that SAGE-VS, JIT-HW and CIF where not even there as tools at the beginning of the project, which is why it was not possible for them in three years to go beyond simple proof of concepts.

At any rate, in the subsections below all the tools are discussed either to provide an answer to the issues and comments that are included in Section 2 within the gap analysis or just to provide a sketch of what will be next for them after CERBERO completion. A tentative Gantt diagram showing the possible future evolution for each tool (except MECA and AOW which are proprietary tools) is displayed in Figure 3:

Figure 3 - CERBERO Development Roadmap tentative Gantt diagram

3.1. DynAA

For DynAA an open source route is planned, possibly via Eclipse, where focus will be on generating supporting documentation and examples.

It is also the intention to use DynAA in other system in the loop setups, which could require additional functionality to support possibly different modes of operation. Furthermore, it is intended to use DynAA for larger simulation in which the parallelism added in CERBERO will be used and possibly further extended.

3.2. MECA

Being a proprietary technology, MECA will not be made available open-source. However, the intention is to reuse the core engine of MECA in similar projects and to extend it based on lessons learned from the Smart Travelling use case. To make the MECA core framework more developer/operator-friendly, the core Application Programming Interface (API) of MECA should be extended to support modular insertion of components and to include a library of developed components, such as the ones specifically developed for the smart traveling demonstrator that are indeed deemed reusable.

Starting from these components, in case use cases involving the exploration of alternative itineraries, the search space of itineraries could be reduced to make the computation process more scalable. This can be done by MECA via smart removal of unwanted routes, reducing the time DynAA needs to evaluate possible routes. Alternatively, the standard OSRM route planner could be replaced by a more efficient planning algorithm. In addition, the knowledge base of user preferences can be augmented with machine learning techniques to better match the user preferences over time. This way, suggested decisions become more tailored to specific users. This fits well with current trends on consumer software for decision support. Note that these steps are most likely not required for other use cases in general.

3.3. CIF

To make the CIF more widely adoptable it would be good to be able to support complex merge, since during the duration of the project, where CIF was conceived and developed, it was possible only to realize the support for simple merge. A simple merge is conversion of a data set from one tool to another. A complex merge is the ability to merge multiple data sets in different directions, which would be needed to convert data back and forth between different tools or between more than two tools at the same time. Complex merge is required to merge models of multiple tools in CIF so models can be shared by multiple tools in multiple directions (read model, update it and merge it back again with the already existing models).

By making CIF open-source we can attract additional developers to further help to improve the CIF software.

TNO, IBM, INSA and Abinsula plan to further extend the functionalities of CIF in future projects and will sign an open source agreement before the end of the project to ensure every partner can freely use the current tool as a start for further development.

3.4. AOW

To make AOW available for wide usage IBM plans to separate actual AOW code and remove any dependencies from CPLEX optimization solver. This will be done by producing standard optimization problem formulation in MPS format that can be solved by many other optimization solvers. Thus, AOW will be independent from optimization solver and potentially can be used also in run-time mode, if corresponding optimization solver will be available. Code separation will also allow to distribute AOW under open source license. IBM will strive to release AOW under open source license at the next year.

3.5. PREESM

Following the gap analysis, a better separation of PREESM code from the eclipse framework will be studied to build a more lightweight version of PREESM, usable through a command line interface.

PREESM intends to extend its support to new kinds of architectures, notably in the high-performance computing domain, with the support of heterogeneous distributed computing nodes embedding CPUs and HW accelerators, but also FPGAs.

Usability of the PREESM tool-suite for the deployment of domain specific DSL (e.g. Machine Learning or Computer Vision DSL) is also planned.

3.6. SPIDER

Performance of SPIDER will be further improved, following the direction taken during the CERBERO project, notably by using more efficient model-based heuristic for the management of HW resources, but also by adopting a divide-and-conquer approach in a distributed version of the run-time.

3.7. PAPIFY

It is planned to continue supporting PAPIFY during following months in order to increase the maturity of this instrument-related tool.

Models of architecture and machine learning techniques will be utilized to provide to SPIDER not only the monitored value of performance monitors, but also Key Performance Indicators estimation capabilities, which gives more useful information for adaption triggers.

Also, extended support will be provided in order to integrate PAPIFY PMCs onto manycore architectures.

3.8. ARTICo^3

ARTICo^3 increased maturity beyond CERBERO will be enforced by using the CERBERO results with other projects/use cases, on present projects (such as Platino project, open and active until end of 2020 [PLATINO]), and applying for new project proposals where these results might be of interest by other partners.

Moreover, it is intended to produce convergence with UPM-CEI IoT products and expertise. Computing at the edge (and maybe fog) with IoT devices with hybrid MPSoC technologies with dynamic adaptation is envisaged. To this respect, UPM will participate in an ECSEL project called Insectt, to start in 2020, where these technologies will be applied in the context of IoT device in the edge with FPGAs, HW acceleration, neuroevolvable HW (such as the PoC developed within CERBERO), among other tasks.

Additionally, the recently added functionality of IMPRESS of being able to produce relocatable bitstreams for area-compatible slots is being integrated with the ARTICo3 framework. This will produce important area memory footprint for bitstream storage, since just one position-independent bitstream could be used for several destination slots.

3.9. MDC

MDC tool is currently dependent on Eclipse and the set of documentation available, including how to install the development environment on Windows and Linux, are there to help users with the set-up of the environment. UNISS and UNICA run all along the project timeframe, as mentioned in the previous section and recap below, different tutorials to get feedback on the functionalities and extensions, and on the general usability of the tool to improve and plan the currently available version. Given the current TRL of the tool, which is among 4 and 5, the creation of a stand-alone release of the tool outside Eclipse and the definition of a more evolved user-interface was not prioritized. Nevertheless, providing a better user-interface and the possibility of enabling command-line execution is already in MDC roadmap, after the open-source release of the current version. In fact, in parallel to the graphical executables of MDC, a command line version of the tool is being developed and will be released in the near future in order to provide a valuable alternative to users not prone to the Eclipse environment or aiming at integration of MDC into more complex executable scripts.

The target of the tool will remain in future software programmers with limited expertise in hardware development. MDC abstracts the user from the underlying technology, that is digital system design and reconfiguration, by providing user friendly high-level graph-based and text editors. To mitigate the burden of programming flexible custom accelerators that can be developed with MDC, UNISS plans to extend MDC with an OpenCL interface backend as part of the developments of the FitOptiVis ECSEL project [FIT]. Documentation and tutorials about how to use MDC without being expert of digital hardware design and management are available online and have been released all along the project timeframe.

UNISS and UNICA already planned to give new tutorials in 2020 at the CPS & IoT Summer School 2020 and at the CPS Summer School 2020, which will certainly lead to improve the general status and level of maturity of the distributed and online material, and a training day towards a group of local SMEs has been already planned in February 2020 as part of the technology transfer project PROSSIMO [PROS]. Training will be provided by UNICA and UNISS and the idea is to push MDC closer to market needs exploiting the received feedback.

4. Conclusion

By applying the CERBERO methodology and tool chain for the implementation of the use case demonstrators, a number of gaps between the current version of the tools composing the CERBERO tool chain and the desiderata of the tool users were identified. Some of these gaps were resolved during the project, e.g. by adding or adapting the functionalities of the tools, while some are left for future developments. This document summarizes the non-resolved gaps and define a roadmap for further development and extensions of the CERBERO tools, which could not yet be accomplished with in the project timeframe and budget.

In terms of technology readiness level evolution for the different tools and use-cases, a substantial growth has been produced during the timeframe of the project, fulfilling the fundamental goals and challenges set in early stages. Since most of the tools are academics, a specific path to market does not exist but their maturity will keep increasing in upcoming months as they are involved in specific tech transfer activities.

5. References

[CERBERO 2018]	http://www.cerbero-h2020.eu
[D2.1]	CERBERO D2.1 Scenarios Description (Final version)
[D2.2]	CERBERO D2.2 Technical Requirements (Final version)
[D6.1]	CERBERO D6.1 Demonstration Skeleton (Final version)
[D6.2]	CERBERO D6.2 Planetary Exploration Demonstrator (Final version)
[D6.3]	CERBERO D6.3 Ocean Monitoring Demonstrator (Final version)
[D6.4]	CERBERO D6.4 Smart Travelling Demonstrator (Final version)
[D6.5]	CERBERO D6.5 CERBERO Performance Analysis (Final version)
[FIT]	https://fitoptivis.eu/
[PLATINO]	https://platino.iuma.ulpgc.es/
[PROS]	http://www.cluster-prossimo.it