Document title: Initial definitions for the Arrowhead Tools design principles

Version	**Status**	**Date**
1.1	final	2019-12-11
Author	**Contact**	
Marek Tatara	marek.tatara@dac.digital, +48 507 864 994	

Initial definitions for the Arrowhead Tools design principles

AITIA:	Svetlin Tanyi szvetlin@aitia.ai
BnearIT:	Hans Forsberg Hans.Forsberg@bnearit.se
BME:	Pal Varga pvarga@tmit.bme.hu
DAC:	Mateusz Bonecki Mateusz.bonecki@dac.digital Anna Kwaśnik anna.kwasnik@dac.digital Krzysztof Radecki krzysztof.radecki@dac.digital Marek Tatara marek.tatara@dac.digital
EUROTECH:	Paolo Azzoni paolo.azzoni@eurotech.com
IUNET:	Federico Montori federico.montori2@unibo.it
IQL:	Géza Kulcsár geza.kulcsar@incquerylabs.com
LTU:	Ulf Bodin ulf.bodin@ltu.se Cristina Paniagua cristina.paniagua@ltu.se
POLITO:	Gianvito Urgese gianvito.urgese@polito.it

Abstract

This document contains the most essential definitions for the Arrowhead Tools design principles. The definitions of a tool, a toolchain and an engineering process unit are provided.

ECSEL EU project 826452 - Arrowhead Tools
Project Coordinator: Professor Jerker Delsing | Luleå University of Technology

Table of contents

1. Initial definitions

This document contains a concise summary of the initial definitions for the Arrowhead Tools design principles, which address Arrowhead tools, toolchains and their relation to the engineering process. Those design principles involve formal definitions as well as informal guidelines. The Arrowhead Tools reference architecture realizes these principles in an abstract manner and it is supported by example tools and toolchain implementations.

The Arrowhead engineering process has design-time and run-time elements.

In the Arrowhead ecosystem there are functional systems that contribute to the main target of SoS and are always a part of a local cloud, and there are non-functional systems which support reaching this target and are not necessarily a part of any local cloud.

1. A **tool** is a software or a hardware (with an adequate software on board) entity/artifact that supports CP SoS and SoS engineering activities. The phases of the engineering process in principle can be managed without tools (i.e. with a strong human component), but probably will use some.
   a. It could be a design-time or a run-time tool, depending on its place within the process.
   b. It can be either service provider, consumer, both or none; i.e., in short, if it is compliant with the Arrowhead Framework (the first three cases) or not. We stress that it is not necessary for a tool to support some Arrowhead design phase to implement any services in the strict Arrowhead Framework sense; such a tool is called *Arrowhead-enabled.* In contrast, a Framework-compliant tool is called an *Arrowhead Native Tool.*
   c. The output of a tool should be processable by other tools adopted in the other phases of the engineering process.
   d. The output of a tool should be processable by other toolchains.
   e. A tool is an atomic part of a toolchain, and cannot be broken down in subtools that can work autonomously.

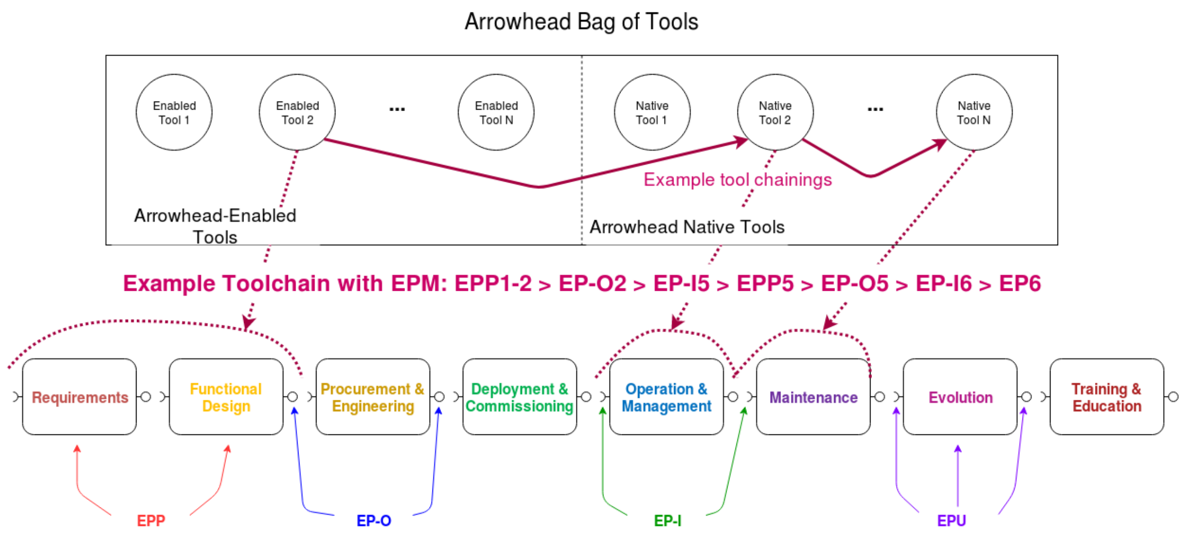
2. A **toolchain** is a collection of tools and of the definitions of the corresponding interfaces potentially organized in chain-based or parallel structures. Tools in a toolchain can be substituted/replaced with other tools with the same input/output interfaces.
   a. It can be design-time, run-time or both.
   b. It aims for a certain level of automation in information processing/transfer throughout the engineering process.
   c. It can allow iterative use of its parts (tools and toolchains).
   d. It can cover only some (not necessarily consecutive) parts of the engineering process, or the whole product lifecycle (typical for general infrastructural tools), even iteratively until the end-of-life phase.

3. An **Engineering Process Unit** (EPU) is either an **Engineering Process Phase** (EPP), based on the EAEM (Extended Automation Engineering Model) or an **Engineering Process Interface**, which can be in turn either an input interface (EP-I) or an Output interface (EP-O) between them. An **Engineering Process Mapping** (EPM) relates a tool to one or more EPU it covers.

2. Tools and Engineering Process Mapping

The following figure shows an abstract overview of the Arrowhead Tools landscape: The upper box represents the so-called Bag (i.e. loose collection) of Arrowhead Tools, where the division into Arrowhead-enabled and Native tools is also indicated. The bottom row of blocks with connectors is a concise representation of the engineering process (see details below the figure). The central part, in purple, is an abstract example of what an EPM for a fictive toolchain might be, and how this could be graphically represented.

The solid arrows between three (fictive, non-concrete) tools represent their chaining, while the single tools are also mapped vie dashed EPM arrows to the engineering process phases and their input/outputs.



For the sake of intuitiveness and logical consequentiality, we represent graphically the EPPs in a sequence, however this does not force the use case designer to follow such phases in this succession; rather, a phase could be executed iteratively, phases could be skipped and connected to “previous” ones and so on. Note that EP-I and EP-O outline inputs and outputs that are meant to be produced/consumed by other tools in the toolchain. A generic input or output of the tool is therefore not necessarily represented this way.

For a better clarity, we give EPPs alternative names as follows:

- EPP1: Requirements
- EPP2: Functional Design
- EPP3: Procurement & Engineering
- EPP4: Deployment & Commissioning
- EPP5: Operation & Management
- EPP6: Maintenance
- EPP7: Evolution
- EPP8: Training & Education

Similarly, EP-I and EP-O are numbered as follows:

- EP-I1: Input for Requirements
- EP-I2: Input for Functional Design
- EP-I3: Input for Procurement & Engineering
- EP-I4: Input for Deployment & Commissioning
- EP-I5: Input for Operation & Management
- EP-I6: Input for Maintenance
- EP-I7: Input for Evolution
- EP-I8: Input for Training & Education

- EP-O1: Output of Requirements
- EP-O2: Output of Functional Design
- EP-O3: Output of Procurement & Engineering
- EP-O4: Output of Deployment & Commissioning
- EP-O5: Output of Operation & Management
- EP-O6: Output of Maintenance
- EP-O7: Output of Evolution
- EP-O8: Output of Training & Education

This means that, for instance, EP-O4 does not necessarily feed only EP-I5, but it can serve as an input of any other phase.

To conclude, in order to clarify better, there may be different conceptual kinds of tools and toolchain: existing tools in the toolchains that are currently used to develop the use cases, potentially covering all the engineering phases (e.g. gcc, Eclipse IDE, Synopsys SW, CAD, e.g for operation Eclipse Kura, NILM algorithms, …); tools that we will develop in the project that could be Arrowhead Framework compliant or not, because of lack of functionalities. Furthermore, there are some artifacts produced by the toolchain in the engineering process (typically in EPP1-4) that are not tools in such phases, but become tools in the following phases (e.g. an IoT framework used for manage fleet of devices, is a software produced by EPP1-4 and in EPP1-4 it is not a tool, but in EPP5-6 it becomes a tool).

3. List of abbreviations

Abbreviation	Meaning
SoS	System of Systems
CP SoS	Cyber-Physical System of Systems
EPU	Engineering Process Unit
EPP	Engineering Process Phase
EAEM	Extended Automation Engineering Model
EP-I	Engineering Process Input Interface
EP-O	Engineering Process Output Interface
EPM	Engineering Process Mapping

4. Revision history

4.1 Contributing and reviewing partners

Contributions	Reviews	Participants	Representing partner
X	X	Marek Tatara	DAC
X	X	Federico Montori	IUNET
X	X	Géza Kulcsár	IQL
	X	Svetlin Tanyi	AITIA
	X	Hans Forsberg	BnearIT
	X	Pal Varga	BME
	X	Mateusz Bonecki	DAC
	X	Anna Kwaśnik	DAC
	X	Krzysztof Radecki	DAC
	X	Ulf Bodin	LTU
	X	Cristina Paniagua	LTU
X	X	Paolo Azzoni	Eurotech
X	X	Gianvito Urgese	Polito

4.2 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2019-09-24	0.1	First Draft	Marek Tatara
2	2019-09-30	0.2	Interface I/O Mapping	Federico Montori
3	2019-10-04	0.3	Refinement of the definitions	Marek Tatara, Paolo Azzoni
4	2019-10-16	0.4	Refinement of the definitions, definition of the mapping	Géza Kulcsár
5	2019-10-25	1.0	Final Version	Marek Tatara, Federico Montori
6	2019-12-11	1.1	Final Version with conclusions	Federico Montori, Marek Tatara

4.3 Quality assurance

No	Date	Version	Approved by
1	2019-12-11	1.1	Jerker Delsing