Post-print of: G. Kulcsár, M. S. Tatara and F. Montori, "Toolchain Modeling: Comprehensive Engineering Plans for Industry 4.0," IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, Singapore, 2020, pp. 4541-4546, doi: 10.1109/IECON43393.2020.9254855.

Toolchain Modeling: Comprehensive Engineering Plans for Industry 4.0

Géza Kulcsár*, Marek S. Tatara†, Federico Montori‡

*IncQuery Labs Ltd., Budapest, Hungary
†Department of Robotics and Decision Systems, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland
‡Advanced Research Center for Electronic Systems (ARCES), Department of Computer Science and Engineering (DISI), University of Bologna, Italy
E-mail: geza.kulcsar@incquerylabs.com, marek.tatara@pg.edu.pl, federico.montori2@unibo.it

***Abstract*—The fourth industrial revolution (Industry 4.0) elevates the complexity and autonomy of industrial systems and engineering environments to levels not seen before. The novel challenges involve not only the software running on the partaking autonomous devices, but also architectural considerations and the technological infrastructure around the entire engineering process. In this paper, complementing the trends in industrial systems design, we propose an approach to toolchain modeling, i.e. an integrated specification for the interoperability of tools along with the holistic architectural framework, designed in the context of the Arrowhead Framework. In particular, we propose an intuitive, yet founded definition for toolchains and their mappings to a versatile engineering process model. Those definitions then serve as a basis for proposing our comprehensive toolchain modeling approach. The methodology is demonstrated using (simplified) real-world engineering case studies based on the Arrowhead Framework and platform.**

***Index Terms*—Arrowhead framework, interoperability, platform integration.**

I. Introduction

The fourth industrial revolution (Industry 4.0) is happening now, and, similarly to its historical predecessors, it requires a paradigm shift which is not automatically happening, but which requires a thorough re-evaluation of our existing notions and concepts in and around industry and engineering.

The novel challenges in Industry 4.0 are manifold. However, in one way or another, they all revolve around different forms of *interoperability* as often summarized in the *Internet of Things* (IoT) concept taken in a broad sense here [1]. In particular, the most direct form of interoperability arising in *Industrial IoT* (IIoT) is on the level of the partaking devices, i.e., systems with increasing level of autonomy w.r.t. their behavior and decisions. Recently, the notion of *Systems of Systems* (SoS) has been proposed as a high-level concept for reasoning about the emerging overall behavior of multiple systems connecting to deal with complex industrial scenarios neither of them could solve on their own [2], [3].

Another, strongly connected take on interoperability is offered by the *Service-Oriented Architecture* (SOA) community. In this context, the *Arrowhead* initiative has a strong focus on fostering interoperability via service and interface descriptions between systems and components in modern SoS scenarios [4]. Indeed, the Arrowhead community has originated from a joint European effort of more than 80 industrial and academical partners with the goal of bridging the interoperability gaps for applications and tools in IoT-based automated industrial scenarios. The results were promising and inherently fostered a set of related projects.

However, the new industrial revolution also revealed a substantial gap of the engineering landscape: *tools and processes prevalent in large-scale industry and manufacturing do not live up to the interoperability expectations of the systems (of systems) they produce and by which they are productive*. In other words, tool interoperability and architecture interoperability (involving also comprehensive engineering processes) still lack an accepted foundation; in particular, there is no accepted notion of *what a tool actually is* when it comes to supporting modern (IIoT) engineering; also, there is no established means to express a *unified* view of how a collection of tools, the engineering process where they are employed and the actual systems (of systems) they consist of.

Therefore, towards filling this complex gap, we examine synergies of Arrowhead (SOA), as well as tools which aid, and processes which guide engineering. In particular, we are making the following contributions:

- we propose a clear definition for what qualifies as a *tool* in this context, with intuitive examples for different kinds of tools and non-tools;
- building on that, we propose an intuitive, yet founded definition for toolchains
- along with their mapping to the recently updated Arrowhead engineering process, finally ending up with a comprehensive *toolchain modeling concept* called AHTTC;
- finally, we elaborate on real-life use-cases for singular toolchain models as well as their integration in a novel *Toolchain of Toolchains* (ToT) vision.

II. Related Work

The fourth industrial revolution postulates to elevate the factories and their production capabilities to the next level,

by employing highly connected, interoperable and reconfigurable Cyber-Physical Systems (CPS). To achieve these goals, high level interfaces for implementing the interoperability are needed. Indeed, over the years numerous projects and research aimed to develop methods and models that increase the interoperability level of CPS.

One of the first widely accepted approaches to systematize the automation process (that emerged before the term Industry 4.0 was formulated) was ANSI/ISA-95 standard (IEC 62264) [5], which somehow organized the entire IT automation architecture in five levels with associated time-spans and involved systems, often presented as so-called automation pyramid. Subsequent projects tackling the issue of industrial automation and the interoperability were SOFIA[1] and Socrades[2]. An extension of the ISA-95 standard in the context of Industry 4.0 was proposed as Reference Architecture Model Industrie 4.0 (RAMI4.0) [6], presented in 3 dimensions with axes for "hierarchy levels" taken from IEC 62264, "life cycle & value stream" based on IEC 62890, and "layers" containing functional representation of assets along with recommendations (*e.g.* OPC Unified Architecture [7] was recommended for the realization of communication layer). It was perceived as truly important due to its precursory insight in the idea of Industry 4.0. Among important and already concluded projects contributing to the development of industrial automation one should enumerate EMC2[3], FAR-EDGE [8], Arrowhead [4] and OpenCPS[4]. There are also ongoing European projects contributing to further advancement in widely understood field of CPS, IoT and automation engineering, *e.g.* Productive4.0[5] or Arrowhead Tools[6], where the latter was the initiative under which the ideas presented in this paper were formulated.

Another concept important for the paper is the application development life cycle modeling. The simplest approaches include waterfall model (with clearly defined subsequent phases), iterative or spiral models (including cyclic review of past phases). As a cornerstone for further deliberations one should mention the IEC 81346 standard [9], describing the automation engineering model introduced in the further part of the paper, which extension is proposed as a part of the Arrowhead Tools project. As the complexity of systems is constantly increasing and the desire for continuous adaptation to new requirements and technologies forces nonlinearities of the automation procedure, standardized yet simple waterfall, iterative or spiral approaches do not work anymore, and more sophisticated models are needed.

III. Arrowhead Engineering: Processes and Tools

In this section the main background concepts are presented, on top of which we build our toolchain modeling proposal.

[1]https://cordis.europa.eu/project/id/100017
[2]http://www.socrades.net/
[3]https://www.artemis-emc2.eu
[4]https://www.opencps.eu
[5]https://productive40.eu/
[6]https://www.arrowhead.eu/arrowheadtools

A. The Arrowhead Engineering Process

The envisioned Arrowhead ecosystem involves not only the technical capacities of the Arrowhead Framework (still constituting the technological basis), but aims at proposing a holistic engineering ecosystem for complex System-of-Systems engineering. The architectural foundations of such an approach lie in a flexible *engineering process model*. In fact, such a model has been recently proposed for Arrowhead by Urgese *et al.* [10]. Following that paper, we will refer to this process model as AHT-EP (short for Arrowhead Tools Engineering Process).

The main principle for designing this engineering process has been to combine the flexibility and adaptivity to meet current, but also to retain a level of solidity from earlier similar process models, being sufficient for incorporating legacy solutions, which is indispensable while driving a large-scale industrial revolution; here, key fields like manufacturing bring with them the material inertia which we cannot ignore.

In particular, to retain a link to legacy approaches, AHT-EP is principally built upon the so-called *Extended Automation Engineering Model* as defined in the ISO/IEC 81346 standard [9]. The main addition is that AHT-EP reflects on state-of-the-art engineering practice by allowing for the use of engineering process phases in an arbitrary order if some use-case motivates that. Still, the way how the eight AHT-EP *engineering process phases* (EPP) are displayed hints at a non-mandatory, yet traditional sequencing, thus, retaining a bridge to earlier, fixed-order engineering processes.

The eight blocks with colored inscriptions at the bottom of Figure 1 show the EPPs with an explanation of further accompanying elements: in concrete instances of AHT-EP, the actual EPP ordering is shown by connecting the EPPs via *interfaces*. Accordingly, each EPP has an *incoming* (EP-I) and an *outgoing* (EP-O) interface. The term engineering process unit (EPU) can mean any of the three concepts. We elaborate on the upper part of this figure, illustrating our central *toolchain* concept, in Section IV.

B. The Arrowhead Framework

We have seen how the IoT has revolutionized the way in which industrial processes take place, in fact, concepts like "loose coupling" and "late binding" acquired paramount importance. We observe a stable shift from consolidated SCADA/DCS-driven industrial monolithic (legacy) approaches to distributed and networked ecosystems in which each entity has a definite set of responsibilities. Actions are then translated into offering and/or consuming services, which are the basic definitions of Service-Oriented Architectures (SOA). The Arrowhead Framework, originally the result of an effort of more than 80 partners in the EU project Arrowhead [4], is one of the platforms that aims to lead the way to such a revolutionary vision. The Arrowhead Framework consists of a set of interconnected local clouds, each of them consisting of a set of Arrowhead Systems, *i.e.* software artifacts that provide or consume one or more services and, therefore, they are inherently called *Service Providers* or *Service Consumers*

(a system can be both). In order then to accomplish the main features of SOA, namely lookup, late binding and loose coupling, each local cloud hosts a number of Arrowhead Core Systems (CS), that support and dynamically manage the information flow amongst all the other systems in the cloud, addressed to as "applications systems". Arrowhead CS are subdivided in "Mandatory CS" and "Support CS", which are devoted to other optional functions [11]. Mandatory CS are described below:

- **Service Registry**: it supports service lookup, discovery and registration, in other words, plays the role of service broker in an SOA. Each service provider in the local cloud needs to register itself to the Service Registry, which stores its service record in a database alongside with its definition, interfaces, supported protocols and endpoint. In such way, the service exposed by the provider is reachable to authorized consumers requesting its information.
- **Authorization & Authentication**: it manages the correct authentication and interaction rights between application systems, so that consumers cannot have access to providers to which they are interdicted within the cloud.
- **Orchestration**: it guides the interactions between providers and consumers so that there is no need for specifying their preferences at design time. In particular, it is capable of choosing dynamically a service provider that suits best a consumer request on top of a set of orchestration rules as well as the type of service requested. This can ensure handling of load imbalance and faults.

A number of support CS has been implemented and proposed. Although they are not mandatory, they can serve as a concrete aid depending on the application or the use case considered. Some of them are: the Historian, the Event Handler, the Plant Description, the Translation System, the Configuration Manager, the QoS Manager. We also mention the Gatekeeper System and the Gateway System, which are the enablers for inter-cloud communication and allow for data flowing between two different local clouds [12].

Up to now, local clouds have always been considered as a connected set of application systems that concur, with the support of the CS, – this whole picture is often addressed to as a *System of Systems (SoS)* – in essentially carrying out the activities of the use case only in its operational state, whereas several other activities are performed in its design phases, which cannot be neglected. This brings us to the definition of a new entity of the Arrowhead Framework other than CS and application system: the Arrowhead Tool, which is defined more in detail in Section III-C. Arrowhead Tools are aiding artifacts that originated from the need of automatizing all – design-time and run-time – phases of the industrial engineering process. We will actually show how such artifacts lead the use case throughout its whole life cycle.

C. Tools in Arrowhead

As for the particular scope of the ongoing Arrowhead Tools project, the goal is to explicitly address the *tool landscape* in and around the Industry 4.0 context. As emphasized before, the growing complexity of system (and even more, SoS) behavior necessarily calls forth an increasing involvement of tools and tool interactions that support the engineering process.

But what *is* a tool in this context? The key observation here is that an Arrowhead Tool is not necessarily something *technically attached* to actual deployments of the Arrowhead Framework. In contrast, in the context of a digital revolution and after decades of advancements in industrial computing, we can be sure that an Arrowhead Tool *is* or *has* a piece of software. It is instructive to quote the officially accepted definition of an Arrowhead Tool as proposed by the present authors, reflecting and elaborating on the above observations. Afterwards, we summarize the key points and provide some examples. It is important to stress that henceforth we will use the word "tool" to identify "Arrowhead Tools" just as they are defined below, not by its English dictionary definition.

A tool is a software or a hardware (with an adequate software on board) entity/artifact that supports Arrowhead SoS engineering activities. The phases (EPPs) of the engineering process AHT-EP in principle can be managed without tools (i.e. with a strong human component), but probably will use some.

- *It could be a design-time or a run-time tool, depending on its place within the process.*
- *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 EPP 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.*
- *The output of a tool should be processable by other tools, potentially addressing different phases of the engineering process.*
- *A tool is an* atomic *part of a toolchain, and cannot be broken down in sub-modules that can work autonomously.*

As the line between a tool and an application system, and also between tools and non-tools, might be unclear in some cases, we proceed with a number of examples in different categories.

Non-tools. Let us first imagine an industrial scenario in which an automated machine prints silicon boards. We could say that silicon boards could be also hand-made with a significantly increased effort and, therefore, the machine is a tool that aids the production. In fact, such a machine supports the Operation & Management engineering phase; however, it seems so indispensable that we consider it as a *baseline*; theoretically, manual work is imaginable, but in an industry scenario, not anymore. A similar consideration applies for, e.g., compilers, which could be seen as tools because they translate human-readable code into machine-readable code. However, using a compiler for software engineering is obviously a baseline now. Summarizing, a tool is not only a replacement for manual work in an industrial engineering process, but it also *improves an*

already established industrial baseline (for now, our judgment of baselines is based on common sense and do not elaborate further on a proper definition).

General-purpose industrial tools. Here, we extend the examples above to become actual tools in our conceptual framework. Getting back first to the silicon board printer, we could imagine to plug it into a software system that is able to determine, based on historical sales data, what is the optimal amount of boards to produce every day without wasting resources, how much time the machine should be up and running in order to achieve this number and, based on the curve of the daily price of electricity, when it is optimal to turn it on in order to have the expenses reduced. Now, this software system significantly improves the industrial baseline by cutting its costs without altering its main purpose; therefore it is a tool, specifically devoted to the Operation & Management phase. For a software-oriented example, let us revisit compilers: Integrated Development Environments (IDE) used for programming usually have an integrated pre-compiler that suggests potential bugs at design time: those are tools that help speeding up the Procurement & Engineering phase of producing a software artifact.

Multi-purpose (abstract) Arrowhead tools. To illustrate the above tool concept in its originating Arrowhead context, we first provide some common examples of tools encompassing multiple EPPs, planned to be fully implemented in the future, mature state of the Arrowhead Tools platform. The list is far from being exhaustive but aims at showcasing the diverse scope range of thinkable Arrowhead Tools from early software validation to actual production.

- *Test Tool:* A general software solution to test basic validity requirements for any Arrowhead local cloud before its deployment; e.g., every local cloud should have a running Service Registry, systems which are known to be communicating from the beginning should have data interfaces with compatible encodings, etc. This belongs to Procurement & Engineering as well as Deployment & Commissioning phases.
- *Deployment Tool:* Software running on a central computer, overseeing the deployment procedure of an Arrowhead SoS design, i.e. installing software systems to their dedicated hardware and establishing communication (Deployment & Commissioning phase).
- *Local Cloud On-boarding Tool:* A piece of software that can be executed on a device with basic wireless and Arrowhead capacities. It can be used for detecting local clouds in the environment that the device is entitled to join, and possibly even manage basic negotiations.
- *Component Presence Detector:* A tool that takes as input camera stream, and produces a binary output denoting whether a physical component (e.g., a piece to assembly) is present at the desired position with a correct orientation. This is an example of Operation & Management tool (that could be either Arrowhead-enabled or not, depending on its implementation).

Real-life Arrowhead tools. Here, we provide a representative selection of tools taken from real-world use-cases. The main message is that tool support for Arrowhead use-cases combines the utilization of established, existing tools and the implementation of new ones, where even the former become Arrowhead tools in our particular conceptual setting.

- *MagicDraw:* MagicDraw[7] is a popular systems modeling tool with strong ties in the MBSE and SysML communities. Within Arrowhead Tools, the feature-richness and extensibility is utilized to lay the foundations for a comprehensive SoS design approach [13]. As such, this is an example for an established, multi-purpose tool employed as an Arrowhead tool. As for the EPPs, Functional Design is a trivial fit, but MagicDraw might also support Requirements, Engineering, and even Operation & Management to a certain extent.
- *Arrowhead Management Tool:* An intuitive, easy-to-use browser UI for interacting with a running Arrowhead local cloud and changing its basic configuration parameters.[8] This is an example for a trivially needed user-centered tool created specifically for Arrowhead.
- *Arrowhead-enabled Reconfigurable Sensor Data Provider:* This is a run-time tool consisting of a hardware component (an STMicroelectronics micro-controller) with bare-metal firmware enabling communication with the Arrowhead Framework over the HTTP protocol, built on LWIP's RAW TCP/IP API stack, along with the implemented support for the developed on-boarding process of new measurement nodes. This is, also being an example of a hardware-software combination, a run-time tool, an (Arrowhead-enabled) service provider, and cannot be subdivided into tools (as expected). The input of the tool is the information about the interface, on which the sensed data should be read. The output of the tool is the data measured on the configured interface. The tool falls under Deployment & Commissioning and Operation & Management phases.

IV. Engineering Process Modeling and Toolchains

The characteristics of an Arrowhead tool as detailed in the previous section (interoperability and atomicity in particular) make this notion appropriate as a basic constituent of well-founded *toolchain* descriptions: (i) the requirements on tool interoperability ensure their integrability into tool sequences and (ii) their atomicity facilitates the clarity of the resulting toolchain architecture.

Just as we did for Arrowhead tools, let us first revisit the accepted definition of an Arrowhead toolchain (as proposed by the authors):

A toolchain is a collection of tools and of the definitions of the corresponding interfaces potentially organised in chain-based or parallel structures. Tools in a toolchain can be sub-

[7]https://www.nomagic.com/products/magicdraw

[8]https://github.com/arrowhead-tools/mgmt-tool-js

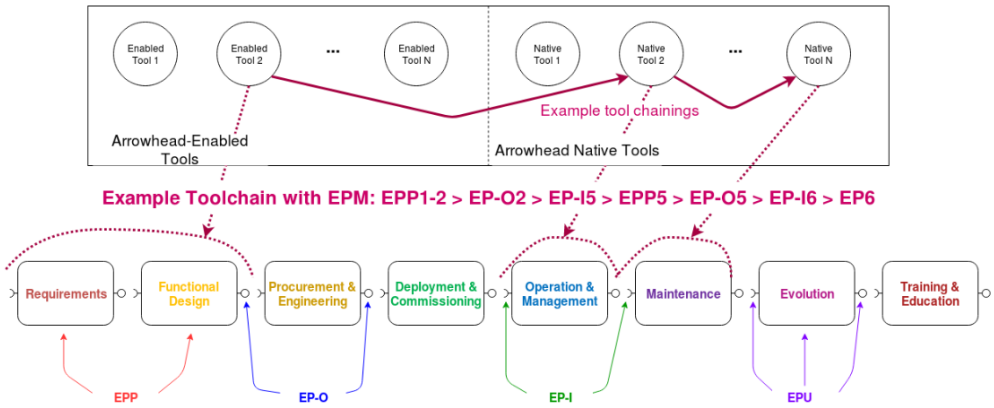

Fig. 1. Abstract Toolchains: Graphical Representation

stituted/replaced with other tools with the same input/output interfaces.

- *It can be design-time, run-time or both.*
- *It aims for a certain level of automation in information processing/transfer throughout the engineering process.*
- *It can allow iterative use of its parts (tools and toolchains).*
- *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.*

Thus, the toolchain concept proposed here naturally integrates with AHT-EP and its flexible phase ordering principle. In turn, specifying a toolchain with an *engineering process mapping* (EPM) combines this flexibility with an adequate amount of requirements to fulfill: whenever two EPPs are attached to each other via their corresponding interfaces, we pose a requirement on the toolchain to realize data exchange in the corresponding direction between those (or generally, some of those) tools addressing the respective EPPs.

The upper part of Figure 1 shows an abstract representation of the overall model resulting from the combination of AHT-EP with our toolchain concept. Circles in the upper row represent an initially unordered collection of existing or even potential (Arrowhead) tools, divided into Arrowhead-enabled and Arrowhead Native (i.e., Framework-capable) tools, emphasizing that those categories are easily combined within a single toolchain. The actual toolchain is then specified by selecting its constituting tools and defining an EPM such that the data exchange links (i.e., the chainings) between the tools are reflected in an EPP configuration via interfaces. Moreover, a correct toolchain specification should be synchronized on both sides: while every contained EPP and interface has to be covered by at least one tool, also, every tool chaining has to correspond to at least one EPP interface pairing (i.e., an output interface connected to an input interface). The figure also hints at a noticeable corner case: sometimes, a single tool might cover multiple EPPs *and also their connecting interface* (cf. the left-hand side of Figure 1).

Thus, we have summarized the principles of a comprehensive high-level engineering workflow model for Arrowhead Tools. In order to align it with AHT-EP, we propose the name *Arrowhead Toolchain Model*, or AHT-TC for short.

While AHT-EP could formally be considered as a part of AHT-TC, we stress that their actual usage highlights their different origins and both models, while sharing a common baseline, are utilized with different focal points and in slightly different contexts. The usage of AHT-EP is elaborated on in [10]; as for the present contribution, AHT-TC, the next section is dedicated to the ways it is and can be employed.

V. Use-Cases

In the following, we consider two kinds of usages for AHT-TC: first, we provide examples for real-life toolchain modeling in Arrowhead; afterwards, building on the capacities of a systems modeling toolchain we sketch future dimensions for collaborative, integrated toolchain modeling, resulting in a *toolchain of toolchains* (ToT) and bringing the Arrowhead Tools vision closer to fulfillment.

On-boarding Toolchain: Developed at DAC digital,[9] the toolchain is a set of tools required to deploy and configure new provider nodes in the context of a distributed measurement system. The toolchain consists of four tools. The first and the core one is provider (sensor) node, suited for the on-boarding process. The node connects through NFC with on-boarding mobile application (second tool) that reads the credentials (e.g. public key) of the node, after which the user can configure the node. The configuration is then sent (over the Internet) to a management (cloud) infrastructure (third tool), that is a central point for connecting a fleet of devices. The last tool in the toolchain is a field gateway (connected to the Internet, and thus - to the management infrastructure), that hosts Arrowhead Framework and aggregates provider nodes in a local cloud. The gateway and nodes are communicating over WiFi (HTTP protocol). It should be mentioned that after a node is admitted to a local cloud, the developed autodiscovery mechanism is triggered to pair the gateway and node.

[9]http://dac.digital

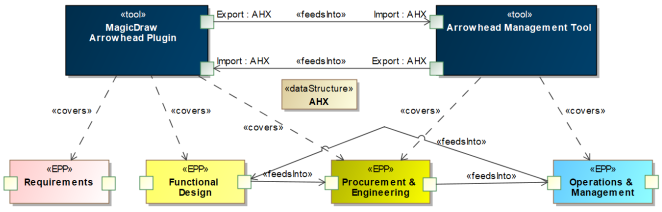

Fig. 2. AHT-TC in SysML

Design & Operation Toolchain: This toolchain, originally proposed in [13] and elaborated and extended since then, contains an SoS design tool (MagicDraw in particular, cf. Sect. III-C), an exchange component and the Arrowhead Management Tool. This toolchain allows for bridging design and operation aspects of the AHT-EP , involving a combination of Arrowhead-enabled "external" tools and Arrowhead-specific extensions.

Toolchain of Toolchains (ToT): Here, we shortly reflect on future perspectives of our toolchain modeling concept, proposing an extension towards *toolchains of toolchains* (ToT), an AHT-TC-centered reflection on complexity trends as seen in, e.g., the SoS notion.

The introduction of AHT-TC into the Arrowhead paradigm will result in a number of compound, yet initially isolated and singular toolchains trying to reach synergies, establishing interconnections, identifying potential conjunctions as well as reductions, etc. We envision ToT to foster that process: it is both an overarching concept of *toolchain analysis* based on existing *model analysis* techniques over the accumulated collection of toolchain models, and an extended toolchain on its own. The IncQuery Model Analysis Suite[10] (IQ MAS) offers advanced analysis means integrated with the model formats appearing here and, therefore, could help addressing both aspects of the ToT vision. Figure 2 represents a real-life artifact connected to both SoS design and the ToT vision, being a SysML-based custom AHT-TC model. (We omit details for now due to space restrictions.)

VI. Conclusion

In this paper, we have proposed a novel concept of *toolchain modeling* called AHT-TC, in the context of System-of-Systems (SoS) engineering as emerging in Industry 4.0 and industrial IoT. Although the interoperability of systems and devices has seen substantial improvements in the recent past, the starting observation of the present paper is that *tool and process interoperability* is still lagging behind, despite the obvious importance for large-scale engineering and, especially, manufacturing. The Arrowhead initiative is fostering a service-oriented perspective on the topic and, in the ongoing *Arrowhead Tools* project, actively investigates tool interoperability for Industry 4.0. In this paper, we proposed a corresponding well-founded, yet intuitive tool and toolchain definition; building on that, reflecting on the engineering process aspects, we ended up with a comprehensive *toolchain modeling* approach also considering the relation between toolchains and process implementations.

The approach is called AHT-TC (for Arrowhead Tools ToolChains) and it concludes the conceptual tool interoperability work within Arrowhead so far—also opening up the horizon for an integrated, collaborative *toolchain of toolchains* (ToT) setting, a vision statement concluding the paper. We see the elaboration on ToT as an immediate task in future investigations. Besides that we would like to develop a versatile tool support for AHT-TC, involving different modeling languages and formats.

[10]https://incquery.io/

Acknowledgements

This research is funded by ECSEL, the *Electronic Components and Systems for European Leadership Joint Undertaking* under grant agreement No 826452 (Arrowhead Tools), supported by the European Union Horizon 2020 research and innovation programme and by the member states.

Project no. 2019-2.1.3-NEMZ_ECSEL-2019-00003 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-2.1.3-NEMZ_ECSEL funding scheme.

References

[1] H. P. Breivold and K. Sandström, "Internet of things for industrial automation–challenges and technical solutions," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 532–539.

[2] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering: The Journal of the International Council on Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[3] M. Jamshidi, *System of systems engineering: innovations for the twenty-first century*. John Wiley & Sons Incorporated, 2009, vol. 58.

[4] J. Delsing, *Iot automation: Arrowhead framework*. CRC Press, 2017.

[5] ISO Central Secretary, "Enterprise-control system integration," International Organization for Standardization, Geneva, CH, Standard IEC 62264, 2013. [Online]. Available: https://www.iso.org/standard/57308.html

[6] P. Adolphs and U. Epple, "Status report RAMI4.0," VDI/VDE Gesellshaft Mess- und Automatisierungstechnik, Tech. Rep., July 2015.

[7] J. Lange, F. Iwanitz, and T. J. Burke, *OPC – From Data Access to Unified Architecture*. VDE VERLAG GMBH, 2010.

[8] M. Isaja, "D2.4 far-edge architecture and components specification," Tech. Rep., 2017.

[9] ISO Central Secretary, "Industrial systems, installations and equipment and industrial products — structuring principles and reference designations," International Organization for Standardization, Geneva, CH, Standard IEC 81346, 2019. [Online]. Available: https://www.iso.org/standard/75265.html

[10] G. Urgese, P. Azzoni, J. van Deventer, J. Delsing, and E. Macii, "An engineering process model for managing a digitalised life-cycle of products in the Industry 4.0," in *Proc. of IEEE NOMS Workshop on Management for Industry 4.0*, 2020.

[11] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, and I. M. de Soria, "Making system of systems interoperable–the core components of the arrowhead framework," *Journal of Network and Computer Applications*, vol. 81, pp. 85–95, 2017.

[12] P. Varga and C. Hegedus, "Service interaction through gateways for inter-cloud collaboration within the arrowhead framework," *5th IEEE WirelessVitae, Hyderabad, India*, 2015.

[13] G. Kulcsár, K. Kadosa, T. Szvetlin, B. Péceli, A. Horváth, Z. Micskei, and P. Varga, "From models to management and back: Towards a system-of-systems engineering toolchain," in *Proc. of IEEE NOMS Workshop on Management for Industry 4.0*, 2020.