The IEC 61499 Function Block Standard: Software Tools and Runtime Platforms

JAMES H. CHRISTENSEN
Holobloc Inc, Cleveland Heights, OH USA

THOMAS STRASSER
AIT Austrian Institute of Technology, Vienna AT

ANTONIO VALENTINI
O^{3}neida Europe, Padova IT

VALERIY VYATKIN
University of Auckland, NZ

ALOIS ZOITL
Technical University of Vienna, AT

KEYWORDS: IEC 61499, function blocks, technology adoption, O^{3}neida, software tools, runtime platforms

ABSTRACT

Part 1 of the IEC 61499 Standard defines an implementable reference architecture for the development, reuse and deployment of Function Blocks in distributed and embedded industrial control and automation systems, and Part 2 of the Standard defines requirements for software tools to support the required engineering tasks. In recognition that these requirements alone are insufficient to guarantee *portability* of software elements among tools, *configurability* of distributed systems and devices by these tools, and *interoperability* of devices within these systems, Part 4 of IEC 61499 defines requirements for **Compliance Profiles** intended to ensure the attainment of these qualities by compliant distributed devices and software tools. For instance, Annexes A and B of IEC 61499-2 define XML DTDs (Document Type Definitions) for the exchange of information among software tools, while IEC 61499-4 requires that Compliance Profiles specify the extent to which compliant software tools are able to employ the syntax and semantics of these DTDs to achieve portability of software elements among tools. This paper presents: (1) an overview of commercially available, freeware and open source software tools and runtime platforms for the IEC 61499 architecture; (2) an comparison of the means by which these tools and platforms meet the requirements for portability, configurability and interoperability; and (3) recommendations for the further development of Compliance Profiles to ensure the attainment of these qualities in multi-vendor systems.

I. INTRODUCTION

Part 1 of the IEC 61499 Standard[1, 2] defines an implementable reference architecture for the development,

[1] IEC 61499-1, *Function Blocks: Part 1 - Architecture;* IEC 61499-2, *Function Blocks: Part 2 - Software tool requirements*;

reuse and deployment of Function Blocks (FBs) in distributed and embedded industrial control and automation systems, and Part 2 of the Standard defines requirements for software tools to support the required engineering tasks. Part 4 of IEC 61499 defines requirements for **Compliance Profiles** intended to ensure the that compliant distributed devices and software tools can possess the following attributes of **open systems**:

- **Portability:** Software tools can accept and correctly interpret software components and system configurations produced by other software tools.
- **Configurability:** Devices and software components can be selected, assigned locations, interconnected, and parameterized by software tools from multiple vendors.
- **Interoperability:** Devices can operate together to perform the functions specified by one or more distributed *applications*.

For instance, Annexes A and B of IEC 61499-2 define XML DTDs (Document Type Definitions) for the exchange of information among software tools, while IEC 61499-4 requires that Compliance Profiles specify the extent to which compliant software tools are able to employ the syntax and semantics of these DTDs to achieve portability of software elements among tools. The relationships among these qualities are illustrated in Figure 1.

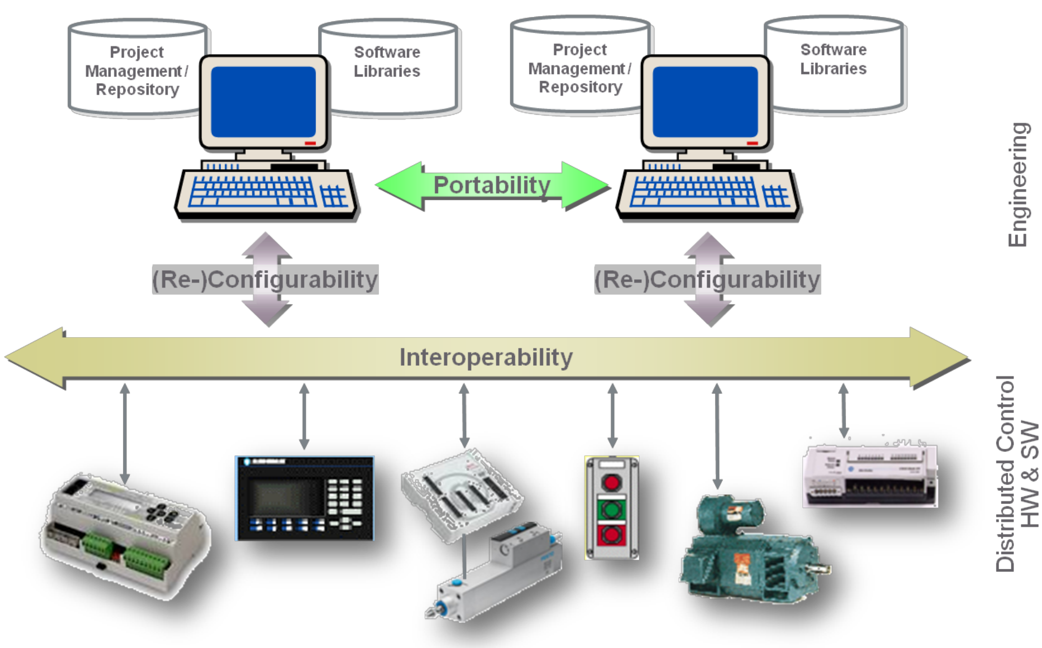

Figure 1 - Portability, Interoperability and (Re-)Configurability

As shown in Figure 2, *software tools*, *runtime platforms,* and libraries of *software components* are indispensable to the widespread adoption and deployment of the IEC 61499 architecture for the next

IEC 61499-4, *Function Blocks: Part 4 - Rules for compliance profiles* (all published by International Electrotechnical Commission, Geneva, 2005).

[2] Second Editions of IEC 61499-1, -2 and -4 are scheduled for publication in late 2012.

generation of intelligent devices, machines, systems and enterprises. In this paper, we present (1) an overview of commercially available, freeware and open source software tools, runtime platforms and component libraries for the IEC 61499 architecture; (2) an evaluation of the extent to which these tools and platforms meet the requirements for portability, configurability and interoperability; and (3) recommendations for the further development of Compliance Profiles and component libraries to ensure the attainment of these qualities in multi-vendor systems.

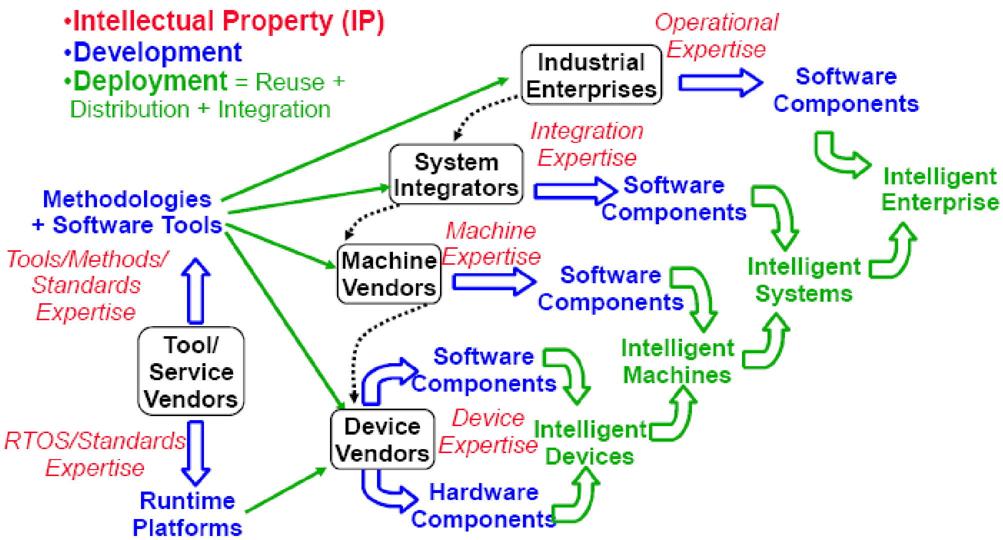

Figure 2 - From Intelligent Devices to the Intelligent Enterprise

II. SOFTWARE TOOLS

This section summarizes currently available IEC 61499 software tools that are either commercially supported or available as open source. It does not include open source projects that are not currently active, such as the CORFU EES[3], TORERO IDE[4], OOONEIDA Workbench[5] or FBench[6].

FBDK[7]

This is the original IEC 61499 software tool. It was initially developed as a simple Java[8] applet to draw FBs and FB networks, because its author, a member of the IEC Working Group (currently Working Group 15 of IEC Subcommittee SC65B - IEC SC65B/WG15) got tired of drawing them with presentation software. It evolved as a tool for testing the graphics model and XML file exchange format defined in IEC 61499-2, and concurrently was used in the development of the first IEC 61499 feasibility demonstration by the Holonic

[3] http://seg.ece.upatras.gr/Corfu/dev/index.htm

[4] C.Schwab, M.Tangermann, and A.Lueder, "The modular TORERO IEC 61499 engineering platform - Eclipse in automation," *10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA) 2005*, vol.2, p.272.

[5] http://oooneida-wb.sourceforge.net/

[6] https://sourceforge.net/projects/oooneida-fbench/

[7] http://www.holobloc.com/doc/fbdk/index.htm

[8] Java is a registered trademark of Oracle Corporation and/or its affiliates.

Manufacturing Systems (HMS) Consortium[9]. To guide and coordinate the efforts of the globally distributed HMS Partners developing this feasibility demonstration, a “Feasibility Demonstration Compliance Profile” (FDCP) was developed[10]. This became the model for the IEC 61499-4 Standard for Compliance Profiles, and is continuously updated as necessary to track the requirements of IEC 61499.

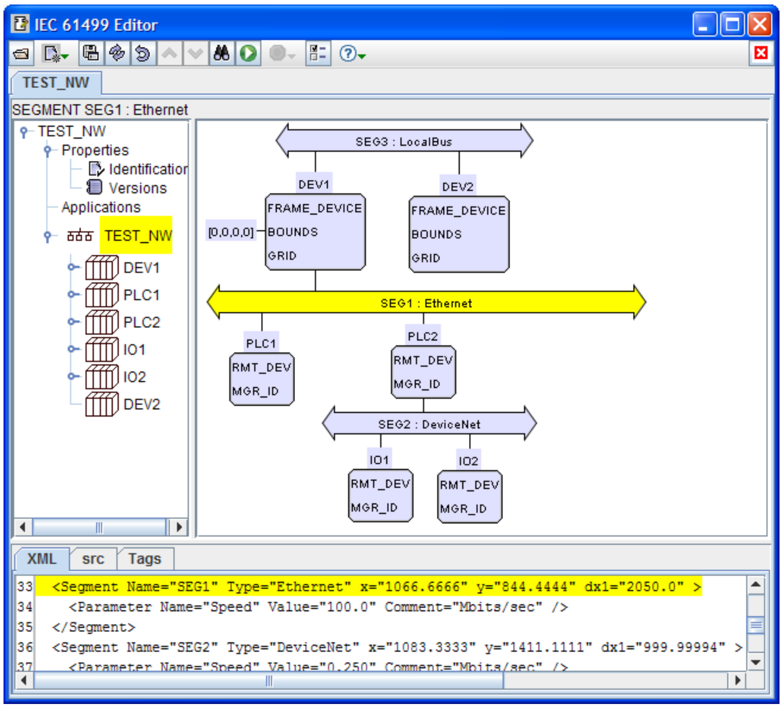

Figure 3 - IEC 61499 development environment FBDK (showing a complete system configuration)

The FBDK is currently maintained and made freely available by Holobloc, Inc[11]; an example of the current release is shown in Figure 3. It is continuously updated to reflect and test proposed changes in the IEC 61499 Standard, and is widely used in research projects and to check the compatibility of commercial products with the FDCP. Its most recent release was in March 2011, and it is currently undergoing extensive code audit and JUnit testing to improve its robustness and maintainability, using the CodePro Analytix toolset from Google[12]. Its next release is expected to coincide with the publication of the Second Edition of IEC 61499 late in 2012.

Unlike more recently developed software tools, the FBDK is currently unable to perform automatic generation of the required communication Service Interface Function Blocks (SIFBs) when a FB is mapped from an (abstract) application to a (concrete) resource. In partial compensation the FBDK supports the “tagged data” design pattern[13].

---

[9] http://www.ims.org/?s=HMS

[10] *IEC 61499 Compliance Profile for Feasibility Demonstrations,* http://www.holobloc.com/doc/ita/index.htm

[11] http://www.holobloc.com

[12] https://developers.google.com/java-dev-tools/codepro/doc/

[13] http://www.holobloc.com/doc/despats/tags/index.htm

ISaGRAF Workbench[14]

In December 2005, ICS Triplex ISaGRAF, a Rockwell Automation company, released version 5 of its well-known and widely used ISaGRAF Workbench for IEC 61131-3 programming with support for IEC 61499, thereby becoming the first commercially supported software tool for IEC 61499. In May 2007, ISaGRAF announced the certification of its IEC 61499 solution compliance by TÜV Süd[15]. The TÜV Compliance Report[16] does not confirm portability of IEC 61499 library elements produced by the ISaGRAF Workbench to or from any software tool other than itself, not does it confirm configurability by the Workbench of any runtime platform other than the ISaGRAF runtime. Currently, ISaGRAF is shipping Version 6 of its Workbench, built on the Microsoft Visual Studio Shell, which enables value-adding extensions by hardware and machine vendors. An example is shown in Figure 4.

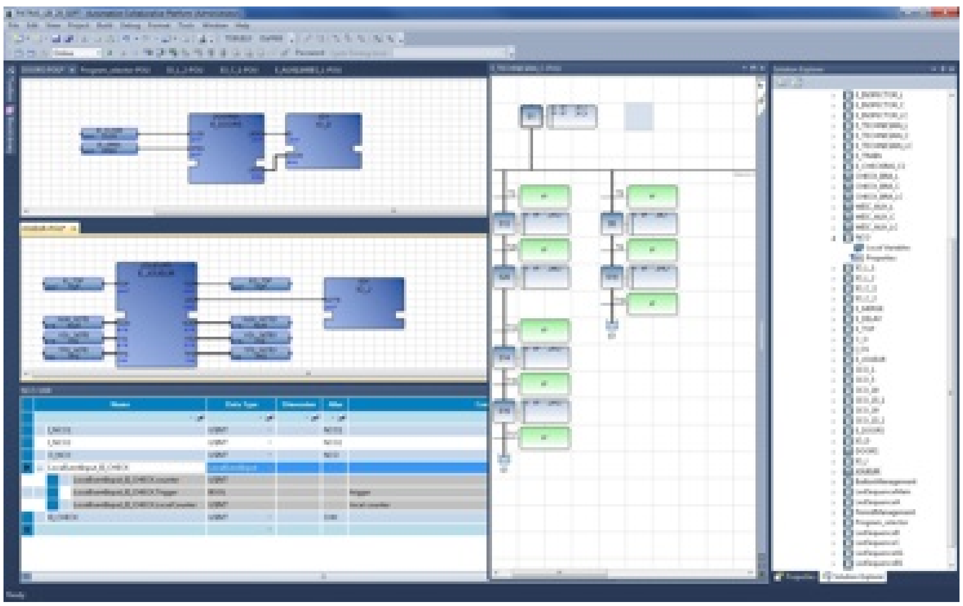

Figure 4 - ISaGRAF version 6 tool

4DIAC-IDE[17]

This software tool, a product of the 4DIAC open source project, is distributed as a set of plug-ins for the Eclipse[18] Integrated Development Environment (IDE). Its first (beta) version was made available in May of 2008, and as of April 1012 it is at release version 1.1.2, with a roadmap in place for ongoing development with two major releases per year. Its freely available and extensible nature makes it a popular tool for research and development[19]. As shown in Figure 5, the 4DIAC-IDE supports the specification of function block types as well as the development of system configurations including the application model and the device

[14] http://www.isagraf.com/pages/products/Isagraf/appworkbench_detail.htm
[15] http://www.isagraf.com/pages/news/PR_TUV-1499_may2007.htm
[16] http://www.isagraf.com/get/ISaGRAF_5-1_1499-TUVreport.pdf
[17] http://www.fordiac.org/
[18] http://www.eclipse.org/
[19] See, for example, http://www.oooneida.org/PDF_docs/ETFA2011_4DIAC_UW_Talk_3_Hofmann_BinPicking.pdf

configurations, as well as deployment of the application to distributed devices. In its most recent release, it also supports the debugging and testing of distributed control applications via online display, setting and forcing of remote data, in a manner similar to ISaGRAF Workbench and nxtSTUDIO. It has been informally tested for exchange of library elements (portability) with the FBDK and nxtSTUDIO software tools, and for configurability of the FBRT and nxtRT61499F runtime platforms as well as its own FORTE platform.

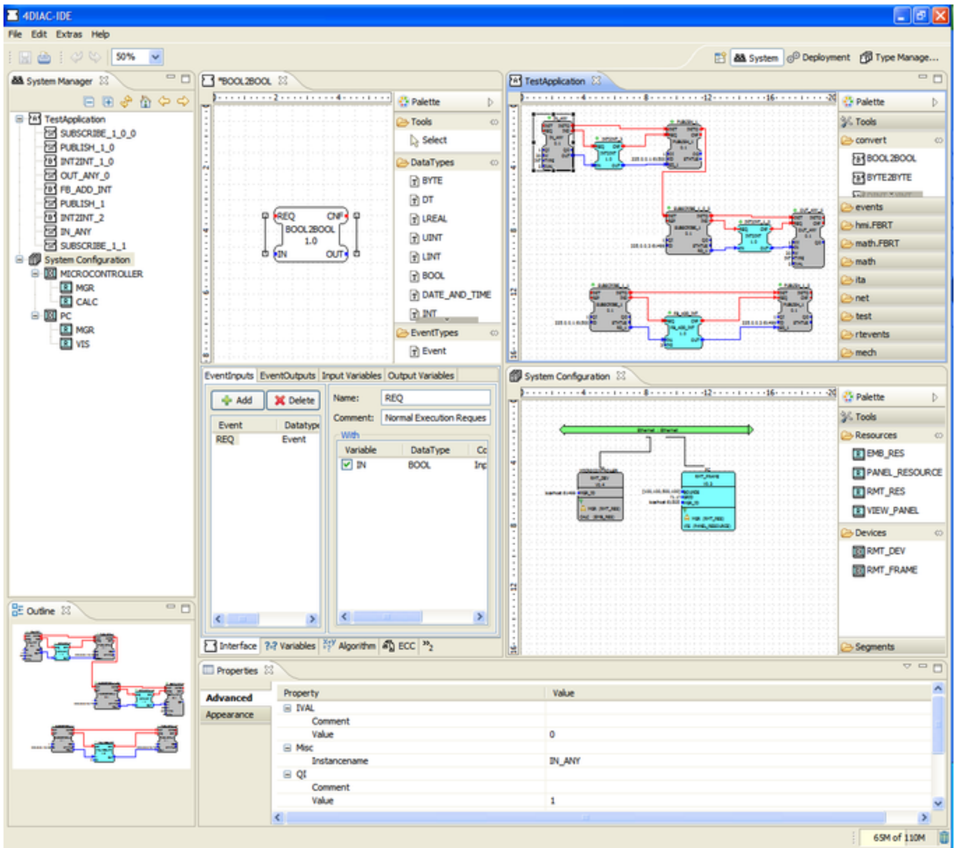

Figure 5 - IEC 61499 development environment 4DIAC-IDE

nxtSTUDIO[20]

This commercially supported IEC 61499 development environment was first introduced at the SPS/IPC/Drives show in Nuremberg in November of 2008 by the Austrian company nxtControl. It is currently at version 1.4 and is used by a number of device and machine vendors from the building automation and manufacturing domains. An interesting feature is its use of Compound Automation Types (CATs) which include control engineering via IEC 61499; HMI/SCADA visualization including symbols, operating dialogues, etc; interconnection of hardware-specific inputs/outputs; and documentation, as illustrated in Figure 6. It has been informally tested for exchange of library elements (portability) with the FBDK and 4DIAC-IDE software tools, and for configurability of the FBRT and FORTE runtime platforms as well as its own nxtRT61499F platform.

[20] http://www.nxtcontrol.com/en/products/nxtstudio.html

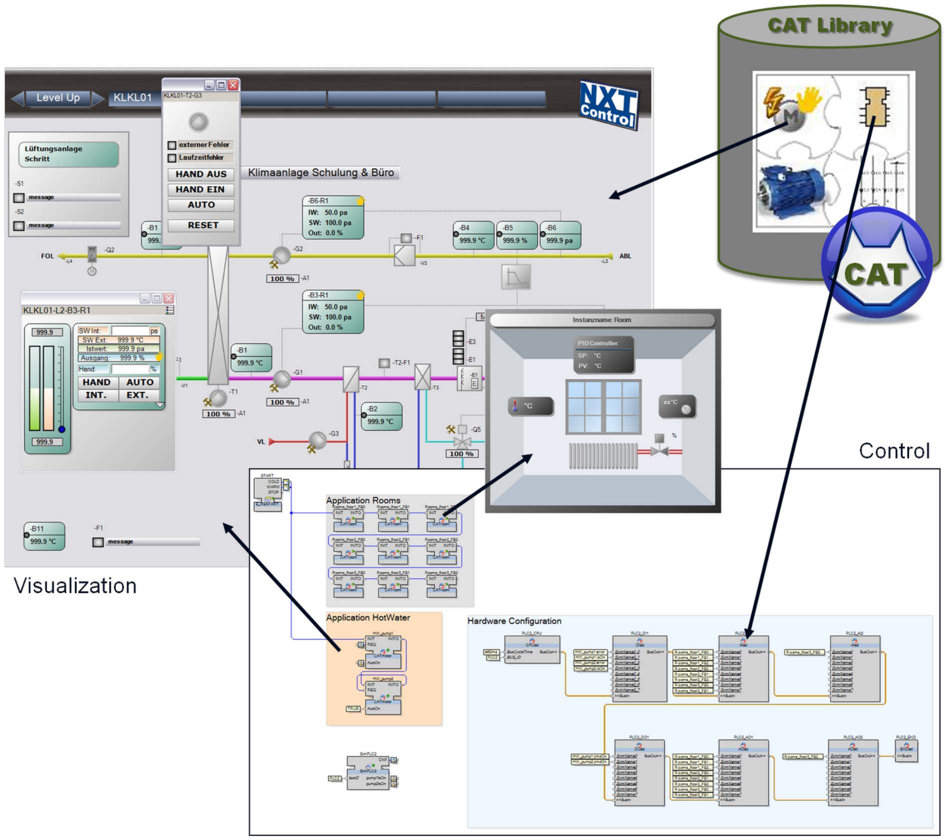

Figure 6 - Integration of control applications and visualisation information via CATs in nxtSTUDIO

III. SOFTWARE/FIRMWARE PLATFORMS

This section summarizes currently available IEC 61499 software tools that are either commercially supported or available as open source. It does not include open source projects that are not currently active, such as CORFU ESS embedded runtime[3] or Fuber[21].

FBRT

Like the FBDK software tool, this runtime platform was developed in the Java language and used for early feasibility testing and demonstration of the IEC 61499 architecture, using embedded Java technology from Imsys Technologies[22]. Subsequently, it was used in a number of research projects with the now obsolescent Netmaster[23] hardware platform.

Now that more economical embedded IEC 61499 runtimes are available, the embedded version of the FBRT is no longer supported. However, the non-embedded version is still available as part of the FBDK software bundle, and continues to be useful for research, teaching and testing, as well as development IEC 61499

[21] http://fuber.sourceforge.net/
[22] http://www.imsystech.com/
[23] http://www.elsist.it/WebSite/Html/English/Products/Hardware/PLC/Netsyst/EnNetmasterII.php

library elements and applications based on the Java SE[24] platform. In addition, an experimental applet-based version of the FBRT is currently in development[25], and the next release of the FBDK will include the generation of Web pages incorporating this applet. Future FBDK releases will include the generation of FBRT-based apps for the Android[26] platform.

Configurability of the Java SE-based FBRT by the 4DIAC-IDE and nxtSTUDIO tools, as well as by the FBDK, and its interoperability with the FORTE and nxtRT61499F runtime platforms, have been informally tested.

ISaGRAF Runtime[27]

This popular, virtual machine-based IEC 61131-3 runtime platform added support for IEC 61499 in 2005. It is certified by TÜV Süd[28] to be configurable by the ISaGRAF Workbench, and interoperable among devices using the same runtime platform, based on a compliance profile supplied by the vendor (in contrast to FBRT, FORTE, and nxtRT61499F, which are essentially compliant to the FDCP, ISaGRAF is using its own Compliance Profile). Interoperability with other runtime platforms and configurability by other software tools have not been demonstrated.

FORTE[29]

The 4DIAC-RTE (FORTE) is a small portable implementation of an IEC 61499 runtime environment targeting small embedded control devices (16/32 Bit). It provides the execution of basic function blocks, composite function blocks, and service interface function blocks. The execution mechanisms in FORTE allow the real-time constrained execution of IEC 61499 control configurations triggered by external events, where different parts of the configuration can fulfill different real-time constraints and the execution of low priority processes does not disturb the execution of higher priority processes[30]. FORTE has been written to be platform-independent, to make it easy to target to diverse hardware and operating system platforms. Currently the open source version has been ported to four operating systems: POSIX (mainly Linux), Microsoft Windows (Win32), ThreadX, and eCos, the last two being real-time operating systems for embedded control systems. These ports have been tested on different hardware platforms ranging from a normal PC to small embedded control devices (e.g., for motion control). Currently the smallest platform supported by the open source version is the Lego Mindstorms NXT controller.

Like the 4DIAC-IDE software tool, FORTE is a product of the 4DIAC open source project, and updated versions of both FORTE and 4DIAC-IDE are provided simultaneously in the same release cycle. Under its Eclipse Public License[31], changes to the FORTE open source code have to be provided under the EPL, but

---

[24] http://www.oracle.com/us/technologies/java/standard-edition/overview/index.html

[25] See http://www.holobloc.com/doc/navigator.htm for examples.

[26] http://www.android.com/about/

[27] http://www.isagraf.com/pages/products/Isagraf/runtime.htm

[28] TÜV Rheinland, *op.cit.*

[29] http://www.fordiac.org/8.0.html

[30] A. Zoitl: "Real-Time Execution for IEC 61499"; ISA, Durham, North Carolina, 2008, ISBN: 978-1-934394-27-4.

[31] http://www.eclipse.org/org/documents/epl-v10.php

vendor-specific additions do not have to be made open source. Thus, this code can serve as the basis for multiple commercially-supported versions; for instance, the commercial runtime environment nxtRT61499F described below is based on FORTE.

Configurability of the FORTE platform by the FBDK and nxtSTUDIO tools, as well as by the 4DIAC-IDE tool, and its interoperability with the FBRT and nxtRT61499F runtime platforms, have been informally tested.

nxtRT61499F[32]

This efficient and cross-platform portable IEC 61499 runtime system for control hardware was released at the same time as the nxtSTUDIO software tool. The core of this runtime is based on the open source solution FORTE but enhanced with additional services and functions. It is optimized for distributed control engineering and has available extensions such as OPC-UA servers, WEB servers, etc. Configurability of this platform by the FBDK and 4DIAC-IDE tools, as well as by the nxtSTUDIO tool, and its interoperability with the FBRT and FORTE runtime platforms, have been informally tested.

IV. COMPARISONS OF TOOLS AND PLATFORMS

Tables 1, 2 and 3 are based on the authors' best estimates of the degree to which the open systems attributes of software portability, device configurability, and interoperability are achieved by the software tools and runtime platforms discussed in this paper. However, these estimates are only approximate and are based on observations of informal tests and demonstrations, except in the case of the ISaGRAF Workbench and runtime platform, where they are based on the TÜV Süd compliance report. More reliable comparisons could be obtained if the following were agreed upon by consensus: (1) a multi-vendor compliance profile; (2) a set of automated test suites for various compliance levels; and (3) formalized test and certification procedures.

Table 1 - Portability of Library Elements between Software Tools

	FBDK	4DIAC-IDE	nxtSTUDIO	ISaGRAF Workbench
FBDK	✓	✓	✓	
4DIAC-IDE	✓	✓	✓	
nxtSTUDIO	✓	✓	✓	
ISaGRAF Workbench				✓

[32] http://www.nxtcontrol.com/en/products/nxtrt61499f.html

Table 2 - Configurability of Devices by Software Tools

	FBDK	4DIAC-IDE	nxtSTUDIO	ISaGRAF Workbench
FBRT	1+	1+		
FORTE	1+	1+		
nxtRT61499F			2	
ISaGRAF Runtime				2

Configurability classes:
(0) = Read, Write, Start, Stop, Reset, Query All Data Types or FB Types
(1) = (0) + Create/Delete FBs and Connections + Kill + Query FBs, Connections, Applications
(1+) = (1) + Query DataTypes/FBTypes
(2) = (1+) + Create/Delete DataTypes/FBTypes

Table 3 - Interoperability of Devices

	FBRT	FORTE	nxtRT61499F	ISaGRAF Runtime
FBRT	✔	✔	✔	
FORTE	✔	✔	✔	
nxtRT61499F	✔	✔	✔	
ISaGRAF Runtime				✔

V. NEXT STEPS

EXTENDED COMPLIANCE PROFILES

As we have seen, an increasing number of tool vendors are now providing software tools and runtime platforms for the IEC 61499 architecture. In addition, a number of vendors are beginning to offer domain-specific libraries for reuse[33]. However, the lack of multi-vendor consensus compliance profiles and corresponding test suites and certification procedures imposes serious barriers to entry and limits flexibility of choice for potential early adopters.

One possible expeditious way forward might be to generalize the existing, widely used "Compliance Profile for Feasibility Demonstrations" by removing implementation dependencies, specifically its references to the Java language and to particular communication protocols. The latter could be easily handled by making the ID input of CLIENT, SERVER, PUBLISH and SUBSCRIBE FB types a string containing a general-purpose

[33] See, for instance, http://www.nxtcontrol.com/en/products/nxtlib.html

Universal Resource Identifier (URI)[34]. The details of the communication protocols could then be specified in a manner dependent upon the URI scheme, e.g., "`devicenet:`", "`canopen:`", "`iec61499-fbdata:`", etc. It may be possible to define interfaces to a large number of well-known networking architectures, as well as hardware interfaces, simply by defining the syntax and semantics of the corresponding *URI schemes*[35].

Another possible use for URIs may be to support a generic means of downloading library element types into devices and resources. Currently, there is no standardized way of doing this because the contents of the download string are implementation-dependent. It may be possible to remove this implementation dependency by using a URI as a download string telling the device "where to get this type."

This still leaves the issue of how to obtain generally accepted certification procedures for such a compliance profile. One possibility could be to create a set of widely accepted IEC 61499 system configurations and supporting library elements to perform automated testing of both the runtime platform and the ability of software tools to accept these library elements and perform the associated testing. The production of these test configurations could be done as an open source project, for instance by using the facilities provided by the SourceForge platform[36].

LEVERAGING OPEN TECHNOLOGIES

As we have seen in the preceding, **free** and **open source** software and runtime platforms have been indispensable in the Launch phase of IEC 61499 adoption through:

- supporting the development of the Standard by testing the **feasibility** of implementing its provisions;
- supporting the development of Compliance Profiles by evaluating their ability to achieve the **open system attributes** of cross-platform software *portability*, device *configurability* and *interoperability*;
- supporting the development of **runtime platforms** through the evaluation of the real-time behaviors of alternative implementations;
- providing platforms for research and development demonstrating the ability of IEC 61499 to deliver sophisticated, **distributed intelligent systems**;
- providing tools for training in the application of the Standard; and
- in some cases, providing **transferable technology** for the implementation of commercially supported tools and runtime platforms.

[34] http://www.ietf.org/rfc/rfc3986.txt

[35] See http://en.wikipedia.org/wiki/URI_scheme

[36] See http://sourceforge.net/

During the Takeoff phase, as the availability and deployment of industrially robust, commercially supported software tools and runtime platforms increases throughout the value-adding chain shown in Figure 2, the emphasis of the free and open source software community will increasingly shift from advanced technology research and development to **risk mitigation** for IEC 61499 adoption:

1. The community will continue to develop prototype software to test the feasibility of proposed changes to the Standard, particularly those that will be proposed for trial use in the projected Technical Specification[37] IEC/TS 61499-5.
2. The community will continue to test ongoing improvements in Compliance Profiles, including the development of automated compliance tests, to ensure that IEC 61499-based systems continue to enjoy increasing levels of openness.
3. The community will continue to research and develop capabilities for increasingly intelligent automated systems, and to work to transfer these technologies to the automation and control marketplace.
4. The community will continue to develop teaching and training materials, including Web-based teaching.
5. The community will increasingly serve as a source of technology transfer and consultation to the automation and control market.

While open-source software and freeware are free, their development and ongoing maintenance are not. Considering the value of risk mitigation alone, the for-profit players in the automation and control market should consider ways in which they can support the ongoing effort of the community with contributions of code, facilities, programmer time, and perhaps even finances.

VI. CONCLUSIONS

An increasing number of commercially supported, freeware and open source software tools and runtime platforms is becoming available to ensure increasing adoption of the IEC 61499 Standard. In order to minimize the risk presented to early adopters as the technology transfers from Launch to Takeoff phase, it is essential that these tools and platforms meet the open system requirements for portability, configurability, and interoperability. The use of Compliance Profiles is indispensable to ensure that these requirements are met. The work of the open source and free software communities, as well as the efforts of commercial suppliers of software tools and runtime platforms, can make a significant contribution to the ongoing development and improvement of the IEC 61499 Standard, its associated software technology and its Compliance Profiles.

[37] J. Christensen, T. Strasser, A. Valentini, V. Vyatkin, A. Zoitl, "The IEC 61499 Function Block Standard: Overview of the Second Edition," ISA Automation Week 2012.