INSTITUTE FOR DEFENSE ANALYSES

A survey of current tools to develop and manage assurance cases

Kevin P. Roback

Other participants
David A. Sparrow
David M. Tate

August 2022
IDA Publication P-33140
Log: H 2022-000262

Approved for public release; distribution is unlimited.

INSTITUTE FOR DEFENSE ANALYSES
730 East Glebe Road
Alexandria, Virginia 22305

The Institute for Defense Analyses is a nonprofit corporation that operates three Federally Funded Research and Development Centers. Its mission is to answer the most challenging U.S. security and science policy questions with objective analysis, leveraging extraordinary scientific, technical, and analytic expertise.

About This Publication

This work was conducted by the IDA Systems and Analyses Center under contract HQ0034-19-D-0001, Project AX-1-3100.51, "DTE&A Initiative," for the Director, Developmental Test Evaluation and Assessments. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Approved for public release; distribution is unlimited.

Acknowledgments

The authors would like to thank IDA committee, Dr. Stephen Ouellette (chair), Dr. Leonard Truett (USAF AFIT), Mr. Michael H. McLendon, Mr. Stuart M. Rodgers, and Mr. Brian L. Williams for providing technical review of this effort.

For More Information

John S. Hong, Project Leader
jhong@ida.org, (703) 845-2564

Stephen M. Ouellette Director, SED
souellet@ida.org, (703) 845-2443

Copyright Notice

© 2022 Institute for Defense Analyses
730 East Glebe Road, Alexandria, Virginia 22305 • (703) 845-2000

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (Feb. 2014).

INSTITUTE FOR DEFENSE ANALYSES

IDA Paper P-33140

A Survey of Current Tools to Develop and Manage Assurance Cases

Kevin P. Roback

Other participants

David A. Sparrow

David M. Tate

Executive Summary

Assurance cases (ACs) are structured, clear arguments intending to demonstrate that a system is sufficiently trustworthy for a particular intended use. It's often hard to build ACs due to their complexity. To help manage this complexity, practitioners in academia and industry have been developing tools that help others to create, manage, assess, visualize, report and integrate their cases into the broader life cycle of a system's development, testing and deployment. Makers of assurance cases (ACs) benefit from the existence of a wide variety of tools that are available to support a pathway to trusted autonomy. We provided an up-to-date survey of these tools with the goal of assisting engineering authorities who wish to use ACs in performing test, evaluation, verification and validation of systems. In particular, we surveyed this landscape of assurance case tools, building on a prior study by *Maksimov et al.* [2]. We evaluated the capabilities of the different assurance case tools by using the criteria from the prior study, in addition to characterizing tool access. We also investigated tool usage by tracking tool mentions in papers introducing AC tools, and in recent papers dealing more generally with assurance cases.

We found nine tools developed since 2018, adding to the 37 found previously by *Maksimov et al.* Of these 46 tools, 22 were available as of this publication; 17 were openly available, while five were commercial tools with paid access. In the "core" functionalities of AC creation, maintenance, and assessment many openly available and paid tools offer strong functionalities (i.e., they use partial automation to make creating, assessing and maintaining assurance cases more efficient). However, in the more auxiliary "quality of life" functionalities including multi-user collaboration, integration with external tools, and reporting of the AC, the most capable tools are usually commercial tools. This difference likely reflects the core advantage of commercial tools; they tend to be more user friendly, setting up a tradeoff for project managers: pay up front for a tool that is easier to use, or use a free tool that may require more time in training, and more time spent on functions such as report generation.

In documenting assurance case tool mentions, we found the most frequently mentioned tools tended to be open access tools; this suggested they may be more well-known and widely used overall. AdvoCATE was the most frequently mentioned tool; it combines strong functionality in the core areas of assurance case development with unusually strong (for open access tools) functionality in assurance case reporting and integration. Eight tools (AdvoCATE, AMLAS, AutoFOCUS3+ExplicitCase, CertWare,

D-Case, OpenCert, Resolute, and NOR-STA) offered moderate or better support across all three core functional areas,

Contents

1. Introduction 1
2. Methods 5
3. Results 9
4. Discussion....13
5. Conclusions 17
Appendix A. References A-1
Appendix B. Acronyms B-1

Tables

Table 1. From Maksimov et al. (2018) 2
Table 2. List of capability evaluations for open access tools found both by this survey and Maksimov et al. (2018) 10
Table 3. List of capability evaluations for paid-access tools found both by this survey and *Maksimov et al.* (2018)....10
Table 4. Number of papers mentioning each tool, split to show results from papers with relatively old and new publication dates....12

Figures

Figure 1. Sources of assurance case tools characterized by this work, and results of access checks on tool availability. 11

(This page is intentionally blank.)

1. Introduction

Assurance cases (ACs) are structured, clear arguments intending to demonstrate that a system is sufficiently trustworthy for a particular intended use. They are growing more complicated as increasingly complex systems challenge efforts to develop comprehensible arguments for far reaching claims about a system's overall safety. This complexity is mitigated in part by using structural notations to organize the argument; the most commonly used is *goal structuring notation* (GSN) [1]. With GSN, a top-level safety *goal* is met through a *strategy*, based on accomplishing *sub-goals* proven true via a *solution* (which commonly incorporates evidence from testing and other sources). A similar notation is *claims, arguments and evidence* (CAE), in which a top-level *claim* mirrors the role of GSN's *goal*. *Arguments* analogous to GSN's *strategies* support claims, which are in turn supported by *sub-claims* (e.g., GSN's *sub-goals*), that are ultimately supported by *evidence*. The branching upside-down "trees" of GSN and CAE diagrams show how broad safety assurances arise from a large body of focused work, and aid stakeholders with safety concerns in identifying, understanding, and evaluating the origins of a claim to safety.

It's often hard to build ACs due to their complexity. Thankfully, practitioners in academia and industry have been developing tools that help others to create, manage, assess, visualize, report and integrate their cases into the broader life cycle of a system's development, testing, and deployment. We provided an up-to-date survey of these tools with the goal of assisting engineering authorities who wish to use ACs in performing test, evaluation, verification and validation of systems. To this end, we investigated various aspects of AC tools. Understanding what the tools can do – their functionality - is of course fundamental, but we also investigated tool access; as some tools are open source, while others require payment to access. Some tools are unavailable as they are still in development or discontinued, and some have access constraints that are unclear. We also investigated relative metrics of interest in and usage of assurance case tools among practitioners, as tools are probably more widely used and discussed because they are of higher quality and/or are easier to use.

Our study builds on previous work [2]; a prior survey of AC tools was undertaken in 2018 by *Maksimov et al.* (2018). These authors identified 46 assurance case tools that existed at the time. For 37 of the tools, enough documentation existed to enable *Maksimov et al.* to evaluate the tool's capabilities across six functional focus areas*;* (1) creation, (2) maintenance, (3) assessment, (4) collaboration, (5) reporting, and (6) integration. Table 1 provides details on the evaluation process used by *Maksimov et al.*

Table 1. From Maksimov et al. (2018)

Describing the evaluation criteria used to judge the functionality of assurance case tools. We re-use this evaluation criteria in our own study

Feature category	D (No Support)	C (Minimal Support)	B (Moderate Support)	A (Strong Support)
Support for AC creation (Creation)	None	Basic support for the manual creation of ACs.	Partial automation or re-use in creating ACs available (e. g., argument patterns and templates)	Automatic creation of complete ACs.
Support for maintaining ACs as they evolve (Maintenance)	None	Manual editing with no guidance on affected parts provided.	Tracking of relevant artefacts (e. g., system models and evidence), notifying user of changes and/or indicating their potential impact on the AC.	Automatic updates of ACs to reflect changes in the relevant artefacts (e. g., evidence, system models, requirements specifications).
Support for assessing ACs (Assessment)	None	Support for manual annotations to indicate potential problems.	Support for syntactical checks (e. g., for well-formedness, completeness and/or consistency).	Syntactic and semantic checking (e. g., validity of overall argument given its supporting arguments and evidence).
Support for collaboration between users (Collaboration)	None	A basic concurrent multi-user environment.	Additional features such as user access/permission management.	A complex multi-user environment (e. g., change requests and change reviews).
Support for creating reports from ACs (e. g., for certification purposes or for different stakeholders) (Reporting)	None	Generic reports with no user configurability, limited range of document formats and/or limited content.	Some user configurability, in multiple document formats and/or containing more content.	High user configurability, extensive document formats and/or detailed/interactive content (e. g., generating different reports).
Support for other design/assurance lifecycle processes (e. g., RE specs, hazop, verification) (Integration)	None	Manual integration.	Some support (e. g., bundling with specific third-party tools).	Extensive support for many other design/assurance life cycle processes.

Maksimov et al. evaluated the 37 aforementioned tools in accordance with this rubric, and assembled the results into a table. They found the only tools offering strong support in the area of assurance case creation were domain specific tools (ENTRUST and Resolute). Domain specific tools are only applicable to specific types of systems; ENTRUST is to self-adaptive software systems, while Resolute is to distributed real-time embedded systems. These tools are able to offer strong support via application of underlying system/behavioral models specific to their respective domains. Strong support in assurance case maintenance was not limited to domain specific tools; of the two tools offering strong maintenance support (ENTRUST and Evidential Tool Bus), one (Evidential Tool Bus) was not domain specific.

In assurance case assessment, seven tools showed strong functional support. Mechanisms of assessment, such as evaluating the presence of supporting evidence, are generally not domain specific; as a result, 6 of the 7 strong tools in this area are non domain specific. Collaboration, reporting and integration capabilities were generally less developed, particularly among tools not developed for industry. *Maksimov et al.* assessed that most tools at the time were likely focusing primarily on development of the "core" assurance case functionalities of creating, managing and assessing ACs; while collaboration, reporting and integration were viewed as "quality of life" features best addressed later.

(This page is intentionally blank.)

2. Methods

Our study builds on the earlier work *Maksimov et al.* performed in evaluating AC tools. To begin, we searched IEEE Explore, ACM Digital Library, Springer Link, Google Scholar and Google for new tools published since the previous survey. We applied the search methods of *Maksimov et al.,* which means we used the terms ("Safety Assurance" or "GSN" or "SACM" or "Safety Case" or "Safety Cases" or "Assurance Case" or "Assurance Cases" or "Safety Compliance"). In an effort to expand our search to locate and include more specific domains, we added "cybersecurity" and "medical device" to the list of search terms.

We expanded upon the earlier work of *Maksimov et al.* by categorizing the tools found by their and our searches as having either open access, paid access, being likely unavailable or unknown access. Tools designated open access were tools for which we could find a web link to a Github or other download page that permitted access to the tool without payment. Paid access tools require payment for full, continuous access; we do not count free trial periods as open access. Tools that were designated "likely unavailable" commonly had no information on access provided in the journal article(s) describing them; subsequent web searches also failed to find any sort of website for these tools. Some of these tools may still be in development, while others may be discontinued. In some cases, tools with associated web links in a publication were moved to the discontinued category when we found the web links were no longer functional. Finally, for some tools the access situation was simply categorized as "unknown"; commonly, these were situations in which publications on the tool simply told readers to "contact the authors" regarding access, with no clarity provided on why the tool must be accessed only through its authors.

For new and updated assurance case tools, available through either payment or open access, we characterized the tools' relative capabilities by re-applying the rubric developed by *Maksimov et al.* (e.g., Table 1). We re-used this rubric because it was straightforward, suitable for characterizing tools for assurance cases, and also enabled us to compare new and updated tools to older tools from *Maksimov et al.*'s list. It also streamlined our study by saving time that would be required to formulate a new rubric, classification scheme, and re-evaluate older tools.

The rubric of *Maksimov et al.* is mostly objective, defining specific functionalities that a tool can be evaluated to either have or not have. Often, these functionalities relate to automation of the processes involved in working with the case; for example, tools assigned "strong support" (an "A" grade) in AC creation are tools that can automatically create full

ACs. Tools with "B" grades can automatically assemble part of the AC. Thus, in theory, tools with high grades should be faster and easier to use, though the rubric does not take into account factors like glitchiness, readability and organization of user interfaces and other factors that might make a tool easier or harder to use.

In some areas, particularly collaboration, reporting and integration, subjectivity is introduced in the contrast between "moderate" (B grade) and "strong" (A grade) support, as no specific functionality is introduced between the A and B grades. In AC reporting, for example, the rubric defines the contrast between the A and B grades as a contrast between tools having "some" user configurability in automatic report generation, and tools having "extensive" user configurability.

Boundaries like this are inherently subjective, so the differences between "A" and "B" grades on collaboration, reporting and integration are very much arguable. We also note that, due to time constraints, we did not work with the tools directly. Therefore, our evaluation assumes the tools' capabilities reflect those advertised in publications.

We also investigated the relative usage and popularity of the various AC tools. Unfortunately, there is no direct way to measure assurance case tool usage; many companies that are involved in safety assurance may not reveal their methods to maintain competitive advantages. As a proxy, we used the group of papers we collected as references for AC tools to look into tool popularity by tracking the number of papers that mentioned each tool. This group of papers is authored by tool developers, who are reporting on their progress in development. Authors often wish to compare their new tool's capabilities to existing tools; this is the primary source of tool mentions. It is in authors' best interests to compare their tools to widely used tools that people know about, rather than little known or poor quality tools. Papers describing new tools also sometimes mention other tools in sections describing related work.

This method of quantifying tool usage has its limitations. Some tools are associated with multiple papers, and some groups of authors have put out multiple associated tools. In an effort to give each group of authors an equal representation, we only considered one tool-description paper from each author group. When multiple papers had been produced by the same author group to describe a tool and its updates, or a family of related tools, we picked the most recent paper to make our group of papers more current. We also did not count "self-mentions"; the only "mentions" we considered were instances of authors citing another group of authors' work, not their own. Our dataset of papers is generally old; 18 of 30 papers were published before 2016, and 27 of 30 were published before 2020. This method is thus heavily biased against newer tools. Our method will also not find any unpublished proprietary tools; however, this detracts little from our effort as such tools would be obviously unavailable.

On May 18, we used a further search on IEEE Xplore to generate a list of more recent papers. We generated 209 results by using the term “assurance case” and considered only papers published in 2021-2022. We scanned this group of papers for mentions of AC tools. Unfortunately, this method is still limited as it is biased toward the academic literature, and most of the papers found through IEEE Xplore did not deal with assurance cases; many were flagged through mention of the terms “assurance” or “case”, and did not contain both. In total, 102 of the 209 results were accessible to us and actually involved some sort of assurance or safety case. These papers did not yield any additional tools for assurance cases. A more thorough survey of tool usage should be considered for a future survey.

(This page is intentionally blank.)

3. Results

The previous search [2] had identified and characterized 37 assurance case tools; we found an additional nine. Of the eight new tools, three had open access, five were unavailable as they were still in development or discontinued, and one had an unknown access situation. Of the 37 pre-existing AC tools from *Maksimov et al.*, we identified 14 as open access, five as paid access, 15 as likely to be unavailable and three as unknown access. We thus reached a list of 22 available AC tools to further study for this work; 17 of these can be accessed openly without payment, and five required payment.

Of the 14 open-access tools found previously by *Maksimov et al.*, we found evidence of updating for four tools. Of the five paid-access tools, we found evidence of updating for three tools. Other tools may have also been updated, but we did not see any evidence, which included changes on Github pages dated to 2018 or later, changelogs posted on project websites, and releases of new publications describing changes. If we couldn't find any of these signs of changes, we assumed the tools had not been updated, and re-used the evaluation made by *Maksimov et al.* Combining the new and updated available tools, there were a total of ten that needed to be evaluated or re-evaluated. These are included in the full list of tools and their capabilities Table 2 and Table 3). A diagram to clarify the tool-categorization results is given below (Figure 1).

Table 2. List of capability evaluations for open access tools found both by this survey and Maksimov et al. (2018)

Tools that are new or updated since 2018 are in bold. Evaluations for non-bolded tools are taken from Maksimov et al. (2018) Evaluations for bolded tools are done by us.

Tool name	Creation	Maintenance	Assessment	Collaboration	Reporting	Integration
ACEdit [3]	C	C	B	D	D	D
AdvoCATE [4]	B	B	A	D	A	B
AGSN [5]	C	C	B	D	C	D
AMLAS	B	B	B	D	B	D
AutoFOCUS3 + ExplicitCase [6, 7]	B	A	A	D	D	B
CertWare [8]	B	B	A	C	D	B
D-Case suite [25, 26]	B	B	B	C	D	B
Eclipse & Papyrus ext. [9]	C	C	A	D	D	D
ENTRUST [10]	A	A	C	D	D	B
Evidential Tool Bus [11]	C	A	D	C	D	B
FASTEN [12]	C	C	B	D	D	D
MMINT-A [13]	C	A	B	D	D	B
OpenCert [14, 15]	B	B	B	B	B	B
Resolute [16]	A	B	A	D	C	B
SafeEd [17]	C	C	A	D	B	C
VERDICT [18]	B	C	C	D	D	D

Table 3. List of capability evaluations for paid-access tools found both by this survey and *Maksimov et al.* (2018)

Names of tools that were updated since 2018 are in bold. Evaluations for non-bolded tools are taken from *Maksimov et al.* (2018)

Tool name	Creation	Maintenance	Assessment	Collaboration	Reporting	Integration
ASCE [19, 20]	C	B	B	B	A	C
Astah GSN [21]	B	C	B	D	C	D
ISCaDE [22]	B	C	C	B	A	B
NOR-STA [23, 24]	B	B	B	A	A	B
TurboAC [25]	B	C	C	D	A	B

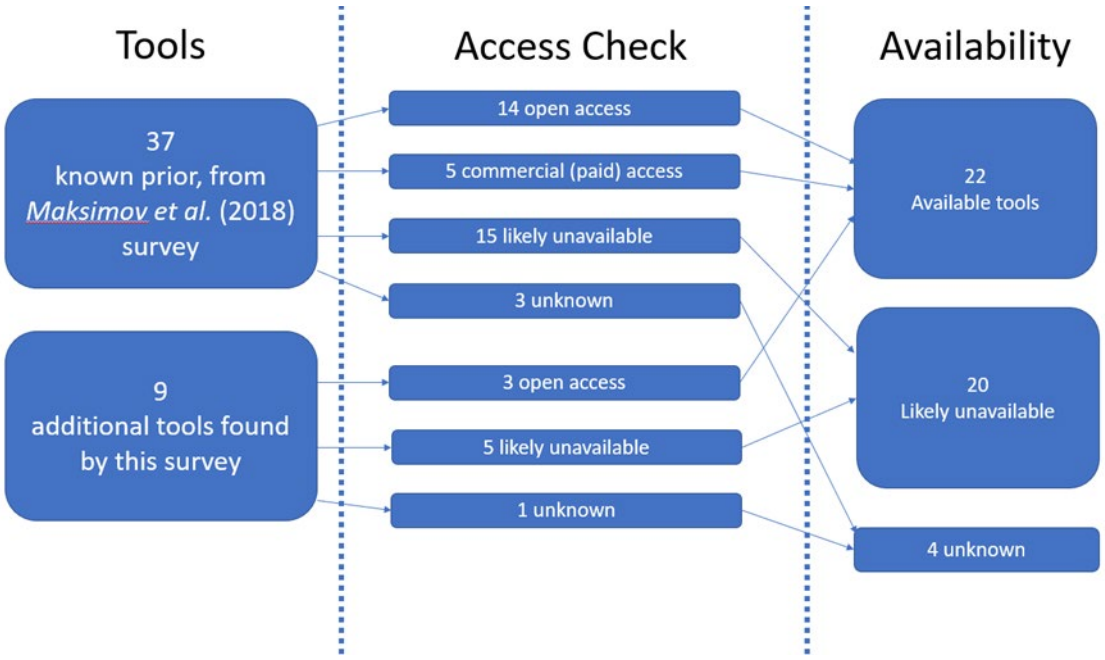

Figure 1. Sources of assurance case tools characterized by this work, and results of access checks on tool availability

Our investigation of mentions of tools across 30 papers describing new or updated AC tools is summarized in Table 4. The most mentioned tool was AdvoCATE, in both older and most recent papers. The next three most frequently mentioned tools were, in order, the D-Case set, Assurance and Safety Case Environment (ASCE), and CertWare. Open access tools were most frequently mentioned.

Table 4. Number of papers mentioning each tool, split to show results from papers with relatively old and new publication dates

Only tools that were mentioned multiple times are shown in this table, and we did not count "self-mentions"; we only counted instances of an author mentioning a different author's tool.

Tool	Mentions before 2016	Mentions 2016-Present	Total	Availability
AdvoCATE	5	7	12	Open
D-Case	2	6	8	Open
ASCE	2	5	7	Paid
CertWare	2	4	6	Open
ACCESS	1	3	4	Likely none
Evidential Tool Bus	0	4	4	Open
ACEdit	1	2	3	Open
Visio plug-in	1	2	3	Likely none
AutoFOCUS3	1	2	3	Open
Resolute	0	3	3	Open
NOR-STA	1	2	3	Paid
ISCaDE	1	1	2	Paid
Safety.Lab	0	2	2	Likely none
EviCA	0	2	2	Likely none
SafeEd	0	2	2	Open
Astah GSN	0	2	2	Paid

Our search for recent (2021-May 2022) papers posted to IEEE Xplore dealing with assurance cases provided 102 results. Mentions of AC tools were infrequent across these papers; however, we recorded only eight mentions of tools across the 102 papers. The tools mentioned once were AdvoCATE, CertWare, D-Case, ENTRUST, the Event-B extension [26], and MMINT-A; Evidential Tool Bus was mentioned twice.

The tools mentioned in *Maksimov et al.* (2018)'s literature search that are excluded from the above tables as they were judged to be likely unavailable, or with an uncertain availability, were ACBuilder, ACCESS, Assure-It, Artisan GSN Modeler, D-MILS, eDependabilityCase, eSafetyCase, Event-B extension, EviCA, GAGE, HiP-HOPS extension, Safety.Lab, SAM, SBVR/GSN Editor, SCT: Safety Case Toolkit, and Visio add-on. The new tools we found that were judged to be unavailable, or with an uncertain availability, were SADL-AT, CyberSAGE, CyberGSN, OpenArgue, VERDICT, and WEFACT.

4. Discussion

In comparing the capabilities of free and paid tools, some general patterns stood out. The strongest capabilities in the "core" functionalities of assurance case creation, assessment and maintenance were found in the freely available tools, while the strongest functionalities in "quality of life" areas such as multi-user collaboration support, automatic generation of reports, and integration with external tools were generally found in commercial, paid-access tools; although the commercial tools also possessed at least basic functionality in "core" areas as well. Unfortunately, the costs of commercial tools were not always stated up front. Of the five commercial tools, only one website (Astah GSN) provided the costs ($1,190 per year as of May 25, 2022) for an organizational license. However, free trials are available from all providers of commercial tools, except possibly TurboAC (no trial arrangements were stated on their website). Most free trials are 30 days in length, although the length of trials for ASCE (the Assurance and Safety Case Environment, from Adelard, a company with a long history of work in safety assurance [27]) are unstated.

The metrics developed by *Maksimov et al.* and re-used in our study have their limitations as previously discussed. Adding understanding of these factors to our survey would require in-depth use and testing of these tools, and was not done given time constraints on this survey. We noted, however, that changelogs for commercial tools (e. g., [21]) often emphasized quality of life improvements and bug fixes in patch notes. Perhaps many commercial tools continued to be used in spite of the presence of free alternatives because they were easier to use, and passed along a savings in time which would compensate for their extra cost. However, given that our work does not directly survey the community of people building assurance cases, we can only speculate on their motivations for using or avoiding particular tools.

Our metrics, despite their limitations, highlight a group of tools frequently mentioned in the literature, implying high use and interest, and rated as highly capable according to the evaluations of both studies. These tools likely warrant first looks from project managers, so we provided a bit of information about them in the following paragraphs.

In the core AC functionalities of creation, maintenance and assessment, Resolute [16] and AutoFOCUS3 with ExplicitCase [6,7] offer the strongest overall support with each offering strong support in two of the core functionalities and moderate support in one. Resolute is a domain-specific tool tailored to distributed real-time embedded systems, such as those found on unmanned aerial systems (UASs). This tool may thus be unsuited to ACs

in some areas, though it is likely a top option for assurance of systems with embedded autonomous or 'AI' capabilities. Resolute is available at https://github.com/loonwerks/Resolute.

AutoFOCUS3 is also geared toward real-time embedded systems, as it permits modeling of software and hardware platforms, and their interactions. It has a more generalized architecture. Though it does not permit the fully automated assembly of ACs, it does permit automated assembly of AC fragments, and is applicable to a wider array of systems. It also supports automated maintenance of ACs, with automatic indication of the impacts of changes to a piece of the argument, as well as automated assessment, with a quantitative measure of overall confidence in the assurance argument. AutoFOCUS3 and its user documentation are available at https://www.fortiss.org/en/results/software/autofocus-3.

Other top performers among free-to-use tools include AdvoCATE [4] and CertWare [8]. Both offer strong support in one of the three core areas and moderate support in the other two. AdvoCATE is a tool developed by researchers at the National Aeronautics and Space Administration's (NASA) Ames Research Center which has capabilities in automated assembly of parts of assurance arguments, and relatively advanced capabilities in the maintenance and assessment of ACs. AdvoCATE has also been applied to perform assurance for UASs developed by NASA, but its tool elements are more generalizable to a variety of different kinds of systems. AdvoCATE is freely available to anyone who contacts the authors (Ewen Denney and Ganesh Pai).

CertWare is another tool frequently mentioned and was developed at NASA, but by a different group of authors. CertWare's GitHub was last updated in 2016, and Adobe Flash Player video tutorials on its GitHub page had not been replaced since the discontinuation of Adobe Flash Player at the end of 2020 made them unviewable. These facts suggest that support for CertWare is winding down, though the tool is still available for now at http://nasa.github.io/CertWare/. CertWare, aside from offering the typical core functionalities in assurance case development, quantifies metrics related to the overall computing effort and project management.

D-Case, a highly mentioned tool in the literature, differs from its alternatives in that it runs in a web browser, hosted at https://mlab.ce.cst.nihon-u.ac.jp/dcase/login.html, rather than as a downloadable program. D-Case supports the manual construction of structures and patterns for assurance arguments, with some reusability of patterns and argument structures and some automated assessment and maintenance capabilities.

Among commercial tools, the most highly rated is NOR-STA, developed by Argevide, an independent software and consulting company. NOR-STA's capabilities combine most of the core capabilities found in the best-performing free tools with capabilities in supporting areas such as multi-user collaboration and automatic report

generation, and a user-friendly interface. More frequently mentioned in the literature is ASCE. Astah GSN [21] is a software tool with relatively limited functionality in automated AC generation and checking, but a long history of tweaks and improvements to enhance user friendliness for manual AC construction. Much like NOR-STA, these tools combine core AC functionalities with a friendlier user interface and expanded capabilities in support for multi-user collaboration and report generation that eases adoption of the tool. All mentioned companies also provide periodic training courses to users.

(This page is intentionally blank.)

5. Conclusions

Our review of available assurance case tools found at least 21 tools available to the prospective assurance case builder as of this writing. Many tools, both paid and free, are highly capable, with strong functionalities in assurance case creation, maintenance, and assessment, along with some capability in reporting, multi-user collaboration, and integration. Many are already being used to do assurance on systems with autonomous capabilities (e.g., [28]).

(This page is intentionally blank.)

Appendix A. References

[1] Goal Structuring Notation Working Group GSN Community Standard Version 1 (Nov 2011), http://goalstructuringnotation.info/.

[2] Maksimov, M., Fung, N. L. S., Kokaly, S., and M. Chechik. 2018. "Two Decades of Assurance Case Tools: A Survey" *In:* Gallina, B. et al. (eds) SAFECOMP 2018. LNCS 11094, 49-59, https://doi.org/10.1007/978-3-319-99229-7_6.

[3] Larrucea, X., Walker, A., and R. Colomo-Palacios. 2017. "Supporting the management of reusable automotive software". IEEE Softw. J. 34(3), 40–47.

[4] Denney, E., and G. Pai. 2018. "Tool support for assurance case development". J. Autom. Softw. Eng. 25(3), 435-499.

[5] Luo, Y., van den Brand, M., Li, Z., and A. Saberi. 2017. "A systematic approach and tool support for GSN-based safety case assessment". J. Syst. Archit. 76, 1–16.

[6] Carlan, C., Barner, S., Diewald, A., Tsalidis, A., and S. Voss. 2017. "ExplicitCase: integrated model-based development of system and safety cases". *In:* Tonetta, S., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2017. LNCS 10489, 52–63. Springer, Cham.

[7] Fortiss. "AutoFOCUS3 – User Documentation." Accessed May 2022.

https://download.fortiss.org/public/projects/af3/help/index.html.

[8] Barry, M.R. 2011. "CertWare: a workbench for safety case production and analysis". In: Proceedings of Aerospace Conference 2011, 1–10.

[9] Huhn, M., and A. Zechner. 2009. "Analysing dependability case arguments using quality models". *In:* Buth, B., Rabe, G., Seyfarth, T. (eds.) SAFECOMP 2009. LNCS 5775, 118–131. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-04468-7_11.

[10] Calinescu, R., Weyns, D., Gerasimou, S., Iftikhar, M.U., Habli, I., and T. Kelly. 2017. "Engineering trustworthy self-adaptive software with dynamic assurance cases". IEEE TSE PP, 99, 1-30.

[11] Cruanes, S., Hamon, G., Owre, S., and N. Shankar. 2013. "Tool integration with the evidential tool bus". *In:* Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) VMCAI 2013. LNCS 7737, 75–294. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-35873-9_18.

[12] Carlan, C. and D. Ratiu. 2020. "FASTEN.Safe: A Model-driven Engineering Tool to Experiment with Checkable Assurance Cases". *In:* Casimiro, A., Ortmeier, F., Bitsch, F., Ferreira, P. (eds). SAFECOMP 2020. LNCS 12234. Springer, Cham. https://doi.org/10.1007/978-3-030-54549-9_20.

[13] Di Sandro, A., Selim, G., Salay, R., Viger, T., Chechik, M., and S. Kokaly. 2020. "MMINT-A 2.0: Tool Support for the Lifecycle of Model-Based Safety Artifacts". *In:* MODELS '20 Companion Proceedings: no 15, 1-5.

[14] Lopez, A., Espinoza, H., Debiasi, A., Ruiz, A., Amparan, E., Sljivo, I., Martinez, J., Adedjouma, M., Atif Javed, M., Puri, S., Larrucea, X., Blondelle, G., Juez, G., and J. Mauersberger. 2021. "Eclipse OpenCert". Accessed May 2022. https://projects.eclipse.org/projects/polarsys.opencert.

[15] Larrucea, X. 2016. "Modelling and Certifying Safety for Cyber-Physical Systems: An educational experiment". *In*: 2016 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA).

[16] Gacek, A., Backes, J., Cofer, D., Slind, K., and M. Whalen. 2014. "Resolute: an assurance case language for architecture models". *In*: Proceedings HILT 2014, 19–28.

[17] Groza, A., and N. Marc. 2014. "Consistency checking of safety arguments in the goal structuring notation standard". *In*: Proceedings of ICCP 2014, 59–66.

[18] Meng, B., et al. 2021. "VERDICT: A Language and Framework for Engineering Cyber Resilient and Safe System". *Systems*, 9(18). https://doi.org/10.3390/systems9010018.

[19] Netkachova, K., Netkachov, O., and R. Bloomfield. 2015. "Tool Support for assurance case building blocks". *In*: Koornneef, F., van Gulijk, C. (eds.) SAFECOMP 2015. LNCS 9338, 62–71. Springer, Cham.

[20] Adelard. "What's new guide to ASCE 5". Accessed May 2022. https://www.adelard.com/partners_files/customer_collateral/MK94v11_What's_new_guide_to_ASCE_5.pdf.

[21] Astah & Change Vision, Inc. "Support for Astah GSN – Help documentation for Astah". Accessed May 2022. https://astah.net/support/astah-gsn/.

[22] rcm2 limited. "Integrated Safety Case Development ISCaDE Safety Requirements, Goals and Hazard Log All in One". Accessed May 2022. http://www.iscade.com/.

[23] Górski, J., Jarzebowicz, A., Miler, J., Witkowicz, M., Czyżnikiewicz, J., and P. Jar. 2012. Supporting assurance by evidence-based argument services. *In*: Ortmeier, F., Daniel, P. (eds.) SAFECOMP 2012. LNCS 7613, 417–426. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-33675-1_39.

[24] Argevide. "Assurance case – Argevide". Accessed May 2022. https://www.argevide.com/assurance-case/.

[25] GessNet. "Medical Device Safety Assurance Case and Risk Management Solutions". Accessed May 2022. https://www.gessnet.com/.

[26] Laibinis, L., Troubitsyna, E., Prokhorova, Y., Iliasov, A., and A. Romanovsky. 2015. From requirements engineering to safety assurance: refinement approach. In: Li, X., Liu, Z., Yi, W. (eds.) SETTA 2015. LNCS 9409, 201–216. Springer, Cham. https://doi.org/10.1007/978-3-319-25942-0_13_31.

[28] Meng, B., et al. 2020. “Towards Developing Formalized Assurance Cases”. *In* 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), p. 1-9.

[29] Beyene, T. A. and C. Carlan. 2021. “CyberGSN: A Semi-formal Language for Specifying Safety Cases”. *In* 2021 51st Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshops (DSN-W), p. 63-66.

[30] Hoa Vu, A., Ole Tippenhauer, N., Chen, B., Nicol, D. M., and Z. Kalbaczyk. 2014. “CyberSAGE: A Tool for Automatic Security Assessment of Cyber-Physical Systems”. *In*: Norman, G., Sanders, W. (eds) Quantitative Evaluation of Systems. QEST 2014. LNCS 8657.

[31] Yu, Y. et al. 2011. “OpenArgue: Supporting Argumentation to Evolve Secure Software Systems”. *In* 2011 IEEE 19th International Requirements Engineering Conference, p. 351-352.

[32] Schmittner, C. and D. J. Pribyl. n. d. “WEFACT Datasheet”. https://www.ait.ac.at/fileadmin//mc/digital_safety_security/downloads/Datasheet_WEFACT_EN.pdf. Accessed May 2022.

[] Bishop P. G. and R. E. Bloomfield. 1995. “The SHIP Safety Case”. *In*: G. Rabe (ed.) SafeComp 1995, Proceedings of the 14th IFAC Conference on Computer Safety, Reliability and Security, Springer, ISBN 3-540-19962-4.

[28] Bourbouh, H., Farrell, M., Mavridou, A., and I. Sljivo. 2020. “Integration and Evaluation of the AdvoCATE, FRET, CoCoSim, and Event-B Tools on the Inspection Rover Case Study”. Report NASA/TM-2020-20205011049.

(This page is intentionally blank.)

Appendix B. Acronyms

AC	Assurance Case
ASCE	Assurance and Safety Case Enviornment
CAE	Claims, Arguments and Evidence
GSN	Goal Structuring Notation
NASA	National Aeronautics and Space Administration
PM	Project Manager
UAS	Unmanned Aerial Systems

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
08-2022	Paper	START DATE	END DATE

4. TITLE AND SUBTITLE
A survey of current tools to develop and manage assurance cases

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
HQ0034-19-D-0001		
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
AX-1-3100		

6. AUTHOR(S)
Roback, Kevin, P.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
730 East Glebe Road
Alexandria, Virginia 22305

8. PERFORMING ORGANIZATION REPORT NUMBER
P-33140
H 2022-000262

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER
Mr. Chris Collins DTE&A		

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Assurance cases are becoming more challenging to develop, as increasing system complexities challenge efforts to develop comprehensible arguments proving their safety. Many tools have been developed to aid practitioners in making and understanding assurance cases. Building off of the work of a previous survey, we produce a current survey of tools that aid with the development and management of assurance cases. We find that there are at least 21 currently available tools, most of which are openly accessible. We find that a number of both open access and commercially available tools have strong functional support for the creation, maintenance and assessment of assurance cases. However, commercial tools tend to have stronger support for auxiliary functions such as report generation and integration with other tools, suggesting that they continue to be used in spite of their cost because of their ease of use. Nonetheless, a small number of open access tools also offer significant support in auxiliary functional areas. Program managers can determine which specific tools are best for their project and team by trying out the range of high performing tools found in this survey.

15. SUBJECT TERMS
Goal structured notation; Assurance case; Safety assurance; Software tools

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
Unclassified	Unclassified	Unclassified	SAR	

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER
John Hong	703-845-2564