Argument-driven Validation of Computer Simulations – A Necessity Rather Than an Option

Teodor Ghetiu and Fiona A.C. Polack
Department of Computer Science
University of York
York, UK
{*teodorg, fiona*}*@cs.york.ac.uk*

James Bown
Department of Computer Science
University of Abertay Dundee
Dundee, UK
j.bown@abertay.ac.uk

***Abstract*—Research based on computer simulations, especially that conducted through agent-based experimentation, is often criticised for not being a reliable source of information – the simulation software can hide errors or flawed designs that inherently bias results. Consequently, the academic community shows both enthusiasm and lack of trust for such approaches. In order to gain confidence is using engineered systems, domains such as Safety Critical Systems employ structured argumentation techniques as means of explicitly relating claims to evidence – in other words, requirements to deliverables. We argue here that structured argumentation should be used in the development and validation process of simulation-driven research. Making use of the Goal Structuring Notation, we provide insights into how more trustworthy outcomes can be obtained through argumentation-driven validation.**

***Keywords*-Software verification and validation; Simulation; Modelling.**

I. Introduction

Computer modelling and simulating have been employed, for more than two decades, in researching academic and non-academic topics. In science, these two activities have been expected to lead towards important achievements, such as unifying theories [1] or relaxing the serious limitations imposed by mathematical models.At the same time, modelling and simulating provided researchers with synthetic, rather than analytic results – the self-explanatory power of results obtained from mathematical models was traded for the ease of constructing computer models and simulations [2]. The opacity of such efforts, where the "consequences flow from the premises, but in a non-obvious manner" [3], can hide software errors or flawed designs that inherently affect results.

In Safety Critical Systems (SCS), the situation is rather different. Due to the unacceptable consequences of SCS failure (e.g., airplane crashes, train accidents), such a system becomes operational only after a *safety case*, documenting its compliance with specific safety requirements, is accepted. Transparency, sound documentation and strong safety arguments are a necessity. While in the past, plain text or tabular methods were used for describing safety arguments, visual notations are currently becoming the norm. For our research purposes, we are using the Goal Structuring Notation (GSN) [4], a well established SCS notation.

In this paper, we introduce the concept of argument-driven validation (ADV) and show that a visual notation for structuring arguments is both beneficial and necessary. Argumentation should be used not only post modelling and simulating, but before and during these activities.

The paper continues with a description of ADV in Section II, followed in Section III by a set of scientific considerations and examples of structured arguments, while Section IV details our conclusions.

II. Argument-driven validation

Verification and validation are two activities very familiar to engineers, at the same time being more or less applied in computer-based scientific research. Sargent, for example, uses them in defining his well known research process [5]. In addition, there is a wealth of verification or validity tests that can be found in the computer science literature [6]. However, a *cohesive understanding of what scientific validation requires, is not captured by the existing efforts* that mainly try to solve pieces of the "puzzle". The situation is more evident when referring to complex systems research, where uncertainties make it difficult to establish a precise baseline for expected results [7]. Validation requires more than the use of separate testing methods – it requires a good coordination of testing and reasoning efforts.

We discuss here about ADV of 'in silico' research, based on SCS techniques. The approach is not aimed at generating formal proofs of validity; formal verification of arguments is still in an incipient phase [8]. Instead, focus is placed on inductive reasoning and the process of constructing structured arguments of validity along the various phases of research and development activities.

Expertise in the SCS domain is vast and the number and criticality of problems it addresses makes it an important source of insight for our aims. Uncertainties cannot be completely removed from the safety assessment of a SCS, hence sound arguments, based on the use of adequate evidence

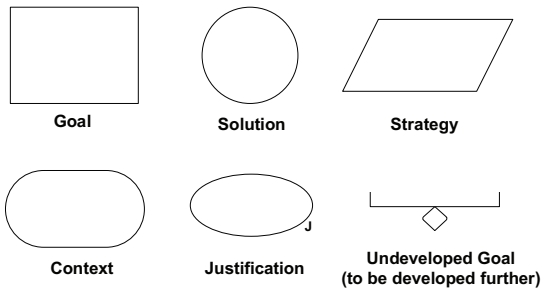

Figure 1. Basic GNS notation

in support of hazard mitigation, are necessary. Structured arguments connect evidence to claims.

Decomposition is a useful features of such arguments: in GSN, for example, claims such as "the system is *safe*" are broken down into sub-claims, this continuing until the argument's structure is sufficiently detailed. The resulting visual argument provides a quick and intuitive view over the main *claims* for a particular product, the *reasoning* used for solving them and the *evidence* they are based on. Each element of an argument is a first-class object, hence it can be challenged: this leads to either the structured argument being extended, or it being rejected. Figure 1 details the basic GSN notation.

There is a long way until establishing a "recipe" for ADV, but this can partially be derived from the features it shares with software testing. Firstly, ADV should be a coordinated, strategic effort of assuring the soundness of each research and development phase. Secondly, ADV can never *prove* a certain research product (e.g model, simulator, results or all of them together) is valid, it can only *claim* that it complies with specified requirements – software testing can never assert a program is completely bug-free [9]. The claim will always be subjective, context dependent and only partially solved. Thirdly, ADV is scalable: is software testing depends on the levels of risk implied by failures [9], structured arguments can similarly sustain a richer or a more reduced degree of detailing, that satisfies stakeholders. Finally, software testing has to consider multiple facets of the same product (e.g., requirements, designs, programs) and equally so must ADV.

One important difference between the two is that, if software testing is a planned, documented activity [9], ADV goes a step further: it facilitates reasoning at a level higher than that of software testing – the level of *relationships or interactions between various testing activities*. Argumentation pulls together all relevant pieces of information, combining them into a data-rich, tree-like structure that is both visually expressive, semantically enhanced and easier to use – it is a more natural representation for software engineers too.

GSN arguments are tree-like structures having goals and strategies as nodes and solutions/contexts/justifications as leafs – the root is called the "top goal", the claim that the argument is aiming to solve. Consequently, the validation process can be "driven" through the proper decomposition of validity goals into sub-goals that are solvable through evidence. The difference to conventional software testing is made by the GSN structure, the way arguments are created and maintained. In more detail, GSN arguments are oriented, multi-level graphs – a goal node can actually be a container for a different sub-argument; in addition, the nodes and leafs are only pointers to evidence located elsewhere, so the whole GSN diagram is a compressed, light-weight means of communicating an argument. Using proper software tools, one can easily perform actions such as browsing, searching through or editing the argument. This also implies that structured arguments can be used in all stages of a (research) project – they can become building-blocks of research and development.

One can develop a set of arguments for different conceptual levels e.g., arguments claiming that the 1) testing plan is efficient and sufficient for the system requirements, or that 2) system requirements are cohesive and compatible with the research purpose, or that 3) test results for each development phase confirm the product's validity, etc. Validity builds on all micro to macro-arguments. Identifying the claims that need to be made and the best ways for solving them, in order to consider a model, simulator or their outputs as valid, is a complex task, addressed also by the CoSMoS project [10].

III. Scientific considerations

In science, opaque assumptions have been known to cause problems. Uncertainty is even higher when computer modelling and simulating methods are used for obtaining "realistic" results. Computer-based research often seeks a "muscle power" advantage over mathematical models: large-scale simulations containing millions of independent, behaviour-rich agents are a reality [11]. This orientation however shouldn't lose track of principles that determine quality. Andrews et al [12] show, for example, that clearly listing all modelling assumptions is a requirement for avoiding conflicts between the research aims and the modelling scope.

Figure 2 reflects an interdependence between domains that are envolved in performing complex systems research. A target domain (biology in this case) is addressed through various computational and mathematical means. This process may however have important weaknesses: essential aspects such as rigurous statistical analysis, may be absent from their findings; claims about the biology of natural systems may not be properly supported by the computational or mathematical means that were employed. These are situations that make ADV necessary: complex research purposes require comprehensive, argumentative ways of proving their

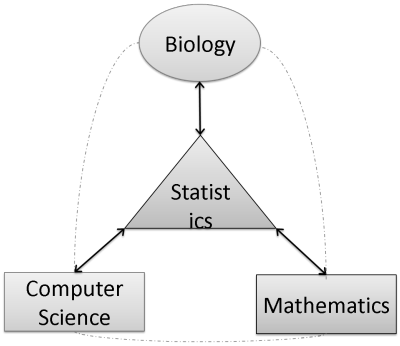

Figure 2. Scientific loop

trustworthiness and structured (GSN) arguments can be used for capturing all this complexity.

GSN has already been used for scientific purposes, albeit only in adressing smaller tasks. An example is [13], where authors construct structured arguments in support of the equivalence of two computer simulations written in C++ and occam-π respectively. It was shown there that what looked as a rapid, straightforward process, required more attention, information and reasoning – all of this being captured in the visual GSN diagram. Validity arguments have different compositions, depending also on the level of precision they require.

Argument patterns can be identified and pattern libraries can be built, so that different goals can be associated with different "recipes". Figure 3 is an example of a high-level argument pattern for claiming validity of simulation results. The GSN diagram shows that, in order to solve the main validity *goal*, it is necessary to define three *contexts*, clarifying the research purpose and requirements and what 'validity' means. The top goal is then decomposed into three sub-goals addressing the validity of: 1) the scientific model, 2) the software implementation of the model and 3)the experimental setup. Apart from the root, each node and leaf is marked symbolically with an 'unsolved' element. When all the elements are solved, the argument is complete.

There is a high degree of flexibility in terms of argument structures for solving a claim. Figure 3 is one partial example. The advantage is that the strengths and weaknesses of the argument are easier to evaluate. One can challenge the whole argument or pick one element at a time and evaluate its appropriatedness. The argument is strong in those areas that are rich in contexts, justifications and evidence, and weaker where these elements are less present. Each arrow in between elements can be similarly challenged. Finally, after all challenges have been addressed and the argument has been extended in order to accommodate necessary changes, it can be said that the initial claim is backed by a stronger argument, or that it has been invalidated.

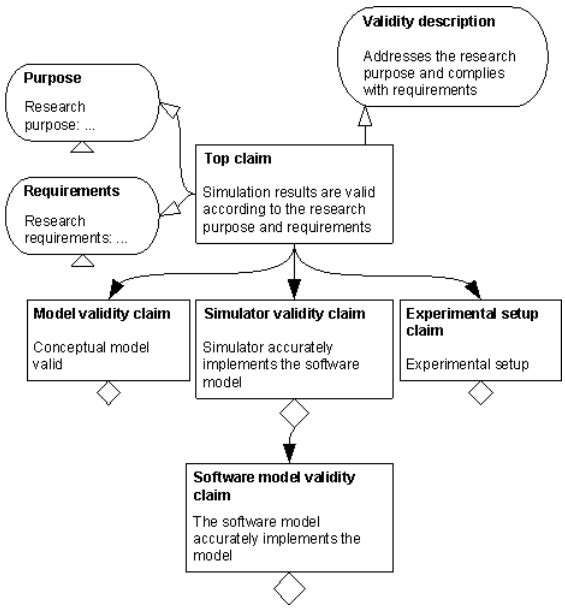

Figure 3. GNS argument for result validity

There are also difficulties with using GSN for ADV. Scale is one issue for which the SCS community hasn't found an efficient solution. There is "no free lunch", an the situation is similar in the case of arguments for science. The complexity of an argument (defined in terms of the number of elements and the multiplicity of relations between them), can seriously impact on its maintainability. The types and efficiency of tools used for developing GSN arguments is also a limiting factor (not many are equipped with modern searching and visualising capabilities). Also, there isn't yet a formal way of expressing uncertainty within a GSN diagram: all goals are equally important, although in reality some are more critical than others. Finally, the syntax itself might need enhancements: GSN doesn't allow contextual elements such as assumptions or justifications to be broken down into sub-elements, as in the case of goals.

A final note is that of the trustworthiness gained through delivering GSN arguments of validity instead of paragraphs of text, to the scientific and non-scientific community. The scientific community can only benefit from undertaking such efforts: this is not a secondary activity, but one that blends with the research and development process itself. The more detailed one's argument is, the more it shows consideration towards the community that will potentially adopt its claims. How detailed an argument must be is up to the researcher to decide – too often, however, the decision misses relevant factors. Figure 4 shows the partially-solved argument through which the authors support this position.

IV. Conclusion

The paper has introduced the concept of *argument-driven validation*, that uses structured arguments as validity build-

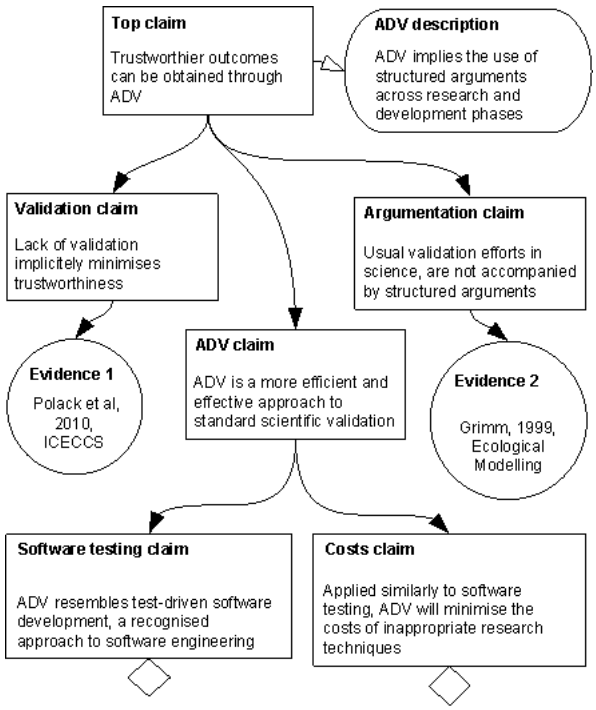

Figure 4. GNS argument for increased trustworthiness through ADV

ing blocks. Domains such as SCS have already certified the value (and weaknesses) of using such techniques. The *challenge of complexity* – studying or engineering systems of higher complexity, that act in ever more complex environments – requires an argumentative understanding of validity, on all levels of interest.

ADV of 'in silico' research shares features and principles with software testing. Validity is a property that stems from the adequate execution of each phase of a project, as assessed through appropriate tests; there are no guarantees though that all hazards have been mitigated, especially when discussing about complex systems functioning in complex environments. Since progress of each research and development phase is guided by reasoning and knowledge, argumentation techniques can provide a stronger methodological foundation.

ADV differs to software testing through the use of structured arguments that enable different facets of validity to be addressed and that may be used as "reasoning building blocks". Structured arguments can capture the validation essence of a project, while seemlessly linking together all the necessary evidence. More than validity can be pursued: argumentation can be applied to any phase of a (research) project, with the aim of assuring other relevant development properties e.g. efficiency, robustness, transparency etc. The collection of arguments thus obtained should provide a sharper view over the quality and validity of the deliverables generated in the process.

ACKNOWLEDGMENT

This work is part of the CoSMoS project, funded by EPSRC grant EP/E053505/1 and a Microsoft Research Europe PhD studentship.

REFERENCES

[1] M. Huston, D. Deangelis, and W. Post, "New computer models unify ecological theory," *BioScience*, vol. 38, no. 10, pp. 682–691, 1988.

[2] J. Bryden and J. Noble, "Computational modelling, explicit mathematical treatments, and scientific explanation," in *Artificial Life X: Proceedings of the Tenth International Conference on Artificial Life*, L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, Eds. MIT Press, 2006, pp. 520–526.

[3] E. A. Di Paolo, J. Noble, and S. Bullock, "Simulation models as opaque thought experiments," in *The Seventh International Conference on Artificial Life*, M. A. Bedau, J. S. McCaskill, N. Packard, and S. Rasmussen, Eds. MIT Press, Cambridge, MA, 2000, pp. 497–506.

[4] T. P. Kelly, "Arguing safety – a systematic approach to managing safety cases," Ph.D. dissertation, Department of Computer Science, University of York, 1999, yCST 99/05.

[5] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of the 37th conference on Winter simulation*, 2005, pp. 130–143.

[6] H. Stanislaw, "Tests of computer simulation validity: what do they measure?" *Simul. Gaming*, vol. 17, no. 2, pp. 173–191, 1986.

[7] M. Read, P. S. Andrews, J. Timmis, and V. Kumar, "A domain model of experimental autoimmune encephalomyelitis," pp. 9–44.

[8] J. Rushby, "Formalism in safety cases," in *Making Systems Safer*, C. Dale and T. Anderson, Eds. Springer London, 2010, ch. 1, pp. 3–17.

[9] W. C. Hetzel and B. Hetzel, *The Complete Guide to Software Testing*. New York, NY, USA: John Wiley & Sons, Inc., 1991.

[10] "CoSMoS project," http://www.cosmos-research.org/, April 2009.

[11] E. Bonnici and P. H. Welch, "Mobile processes, mobile channels and dynamic systems," in *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*. IEEE Press, 2009, pp. 232–239.

[12] P. S. Andrews, F. Polack, A. T. Sampson, J. Timmis, L. Scott, and M. Coles, "Simulating biology: towards understanding what the simulation shows," in *Workshop on Complex Systems Modelling and Simulation*. Luniver Press, 2008, pp. 93–123.

[13] T. Ghetiu, R. D. Alexander, P. Andrews, F. A. C. Polack, and J. Bown, "Equivalence arguments for complex systems simulations – a case-study," in *Workshop on Complex Systems Modelling and Simulation*. Frome, United Kingdom: Luniver Press, 2009, pp. 101–129.