Automating Architecture Alternative Trades

Marshal Blessing

Raytheon

Abstract: This paper describes how advancements in simulation and optimization tools have facilitated architecture trade studies during the initial phases of the system design process. Laying out the architecture for a new system is traditionally done by drawing upon input from stakeholders, experts, and previous work to select a structure that is likely to meet system requirements. While this works well when adapting existing designs to new applications, it does not always provide solid selection criteria, especially when creating distinctly new or innovative systems. This can lead to conforming new systems' architectures to known and trusted design structures, rather than considering novel and potentially superior but untested designs. Selection of an architecture layout without any evaluation of performance can occasionally necessitate radical and costly alterations to the entire system if the architecture proves inadequate during later stages of the development cycle. These issues can be mitigated by creating executable models of the architecture during the initial stages of the design process, but manually creating and testing these models can be time consuming. This paper explains how architecture model creation can be automated in an Isight optimization environment to trade candidate architectures and perform analysis. These techniques can be incorporated into the design process with a minimal amount of manual input or additional time. Ultimately, this has the potential to reduce the cycle time of the design process, and should result in a better quality product. This paper will examine the potential benefits, applications, and challenges of utilizing this new approach.

Keywords: Architecture Modeling, Automation, Design Optimization, Optimization, System Architecture, System Design.

1. Introduction

As modern computing technology becomes increasingly powerful and complex more consideration and planning is required when systems using this technology are being designed. Fortunately, the increased performance of this technology can be put to use through current computer modeling and design tools to assist in the system design process. Most modeling tools focus on system performance and are applied in the later stages of the design process. There are also some software packages that can be used during the initial stages of system design to lay out the architecture of a new system, but these mostly produce static diagrams, which while illustrative do not possess any analytical capability. However, with modern systems, analysis and careful planning are just as important during the initial architecture phases as they are during the final design phases. For this reason, it is important to understand what system architecture is and why it is important. Motivated by the crucial role that architecture can have in a system's design and performance, the creation of executable architecture models has been successfully automated. This work laid the foundation for a new process, wherein multiple candidate architectures for a system can be created and tested as executable models in order to analytically determine which one best meets the system's requirements. This novel approach to architecture design provides

quantitative metrics for architecture performance, considers non-intuitive alternatives, and can be used to fill in holes in incomplete architectures. While some challenges and future work remain, this process has the potential to become a tool which could greatly improve the effectiveness and the efficiency of system architecture design.

2. Background

As the name indicates, a system's architecture serves as the blueprint which will dictate the logical or physical structure of the final system. The IEEE standard defines architecture as: "The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution," (IEEE, 2000). Another definition provided by the Department of Defense Architecture Framework (DoDAF) describes a system's architecture as follows: "An architecture description is a representation of a defined domain, as of a current or future point in time, in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function," (DoDAF Working Group, 2004). To paraphrase, a system's architecture is a general layout of the system's component, usually described through a set of diagrams, tables, and other documents.

Several architecture frameworks are available which serve as templates for the components that make up the architecture as well as their relationship to one another. Some of the well known frameworks include the DoDAF, The Open Group Architecture Framework (TOGAF), and the Zachman framework. While they differ slightly in their structure, each framework serves to help designers plan out and compose the architecture of a new system. Specific courses and certifications are now available to provide system designers with expertise in how to utilize and navigate these different frameworks and architecture in general. Industry is beginning to put special emphasis on this sort of training and recognize "systems architect" as a specific role on design teams.

More and more, systems designers are coming to realize that an effective architecture can benefit the entire design process and can result in a better final product. The architecture lays out the structure of the entire system, and the design choices that go into the architecture can impacts throughout the system's lifecycle. A faulty architecture can result in a system that fails to meet performance specifications or fails to meet the customer's needs. Even if it does meet these requirements, its performance may still be inferior to a better designed system. If problems with the architecture are detected during the system development process, they can be corrected, but this could result in an extensive redesign and delays to the system's development process. Ultimately, this could result in a loss of business for the company. For all these reasons, it is clear that a system's architecture deserves careful consideration.

Unfortunately, defining a new system's architecture is not always a straightforward process. While the notion of "architecture" may be fairly well understood, the architecture of a particular system may not be. There are processes associated with many of the frameworks for building up the components of the architecture, but it still requires some innovative thinking to create these components. Often architectures are designed based on stakeholder input and/or some need analysis, as well as the advice of subject matter experts (SME) and reviews of previous designs. The problem with this approach is that each of these sources of input can be misapplied. Focusing

solely on the customers' wants and needs can lead to an unrealistic or an infeasible design. Occasionally, SMEs can see their area of expertise as the solution to all of the system's design problems, even when a different approach would be more effective. Drawing too heavily upon previous designs can result in an update of the previous system rather than a uniquely new system. Designers are usually more comfortable building off of well known and tested design structures, rather than considering novel and potentially superior but untested designs. Consequently, it can be difficult to get novel designs of quality from the obvious starting points.

3. Executable Architectures

One method to test the performance of a potential new architecture is to create an executable model of that architecture. Architectures are commonly represented by diagrams that show relationships and process flows. These diagrams can be recreated using discrete event modeling tools to facilitate an examination of the architecture in action. This method can be used to verify the process flow of the architecture and check it for completeness. It can also highlight any bottlenecks that result from the architecture's layout. An executable architecture model can be used to check resource allocation within the systems, as well as appropriate scheduling of independent functions. Depending on how much is known about the system and the level of fidelity needed from the model, a discrete event model can provide additional performance metrics for the architecture based on input data describing the system's function. Some current architecture diagram design tools have native simulation capability, but it is usually limited in its analysis capability and requires special consideration when the diagrams are being created. Separate discrete event modeling software packages have been found to be much more flexible in terms of the architecture components that can be represented and the analysis that can be performed. Even so, discrete event modeling is not applicable to all components or products of a system's architecture. It is best suited for components that describe interaction over time, such as an OV-5 Activity Diagram in the DoDAF framework.

While effective, constructing a discrete event model for each architecture diagram from scratch can be a time consuming process. To address this problem, an automation tool has been created which automatically generates executable architecture models based on electronic architecture diagrams. This tool interfaces with common architecture design tools System Architect and Telelogic Rhapsody, and exports the relevant features of individual diagrams to a comma separated value (csv) file. The tool then allows the discrete event modeling software ExtendSim to use the data in the csv file to lay out an executable model of the architecture diagram. ExtendSim assembles this model from a pre-created library of components representing common diagram constituents. After the discrete event model has been automatically generated, some adjustment is still needed to fine tune the model's behavior. Even so, this tool has been shown to significantly reduce model development time and repeated effort in performing this type of architecture analysis.

4. Automating Architecture Trade Studies

By wrapping the automated discrete event model generation tool in the Isight optimization environment, it is now possible to quickly generate multiple executable architecture models and test their performance. The discrete event models are laid out based on the data in the

csv file, so minor changes to that file can result in unique but coherent combinations of the architecture's components. The designer can specify design constraints and performance metrics to maximize or minimize. Isight can then generate and compare a family of architecture models quantitatively to determine which is best. This procedure represents a shift from automating individual model evaluation to automating a portion of the architecture design process.

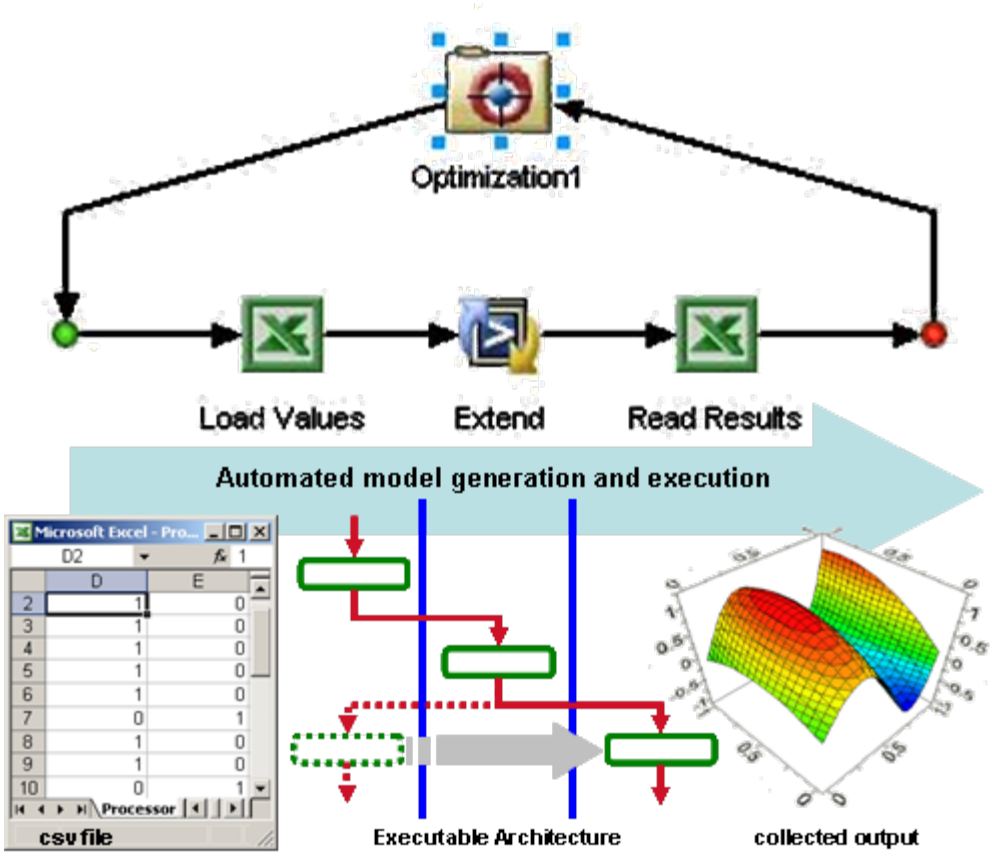

Figure 1. Architecture optimization loop.

This automated procedure begins with an initial architecture diagram to serve as a basis for other potential architecture layouts. The user then generates a csv file describing the initial diagram and identifies aspects of the model or component configurations that could be variable. The csv file is loaded into Isight through the Excel component and the variables are indicated along with constraints on their values. The parameterized spreadsheet can then be used to generate discrete event models using a command call to ExtendSim. Another command executes the generated model, and the results are collected in a separate spreadsheet. The Isight loop can then be used to carry out a design of experiments (DOE) study, a sensitivity study, or a Monte

Carlo analysis; reconfiguring and executing the architecture multiple times to analyze the result. Alternatively, Isight allows the user to specify inputs and outputs to be maximized or minimized as well as their relative weight. In this way, a performance optimization equation is formulated, and Isight can then apply it through an optimization algorithm to try to find the architecture configuration that best meets the user's design criteria. At this point, the architect can inspect the design that was identified as the best, as well as other designs that performed well. They can then move on with the design process or go back and readjust the base architecture or the optimization criteria and run the experiment again, until they are satisfied with the resulting architecture.

5. Example Application

As a notional example, an architect could use this tool when trying to distribute service routines among multiple processors. If the architect knew their system will have two processors connected through a communication bus, then they might want to know how best to distribute a known set of services between the processors in order to minimize traffic over the bus. The distribution of the services can be described as a simple Boolean vector in the csv file. Some additional information may be needed to specify message generation, routing, and transport times between the services. Once rough values for these are provided, Isight can begin creating executable models of the architecture with the services distributed in various ways. As it executes each new architecture, metrics for bus traffic are collected and compared. Depending on the optimization algorithm, Isight will make fewer adjustments with successive simulation runs until it determines that a design cannot be improved. At this point it remains for the architect to accept the optimized architecture or reconfigure the experiment to achieve a different result. They may want to include more performance criteria, such as minimizing message transmission time or balancing the distribution of services between the processors. With each run Isight provides inputs and performance metrics for each configuration tested, which can then be used to go back and improve how the optimization experiment is setup. These experiments are also repeatable and within each experimental run the process of testing alternatives and exploring the design space is greatly simplified.

6. Benefits

The use of this architecture optimization tool provides several benefits in addition to those already listed. One primary benefit is that it can completely explore the design space, and test architecture alternatives that might not have otherwise been considered. In this way, the tool has the potential to discover non-intuitive configurations that provide superior performance. This provides the designers and their customers with assurance that the selected architecture will best meet the system requirements.

Another major benefit is that it evaluates the architectures objectively. Without the executable architecture model, the architecture diagrams could only be observationally evaluated by designers and subject matter experts on the basis of their structure. But there were few quantitative ways to prove that one architecture was better than another. However, this architecture optimization tool utilizes discrete event models to gather performance metrics from each candidate architecture. This allows the architectures to be compared objectively on the basis

of these quantitative metrics. Also, because this data is stored after the optimization experiment, it can be used later for further quantitative analysis during the architecture selection process.

By representing each candidate architecture with an executable model, the architecture optimization tool assists with the verification and validation process. The discrete event model of the selected architecture can be called up at any time and run to verify that its behavior matches what designers had intended. The architecture optimization tool can also produce a large number of candidate architectures with a comparatively small number of inputs. Similarly, it minimizes rework if the initially selected architecture is later found to be inadequate. Data on several other candidate architectures is immediately available, or the optimization can easily be re-run to incorporate new considerations. In this way, the architecture optimization tool has the potential to simplify and speed up the architecture selection process.

7. Future work

Currently the architecture optimization tool is itself in its initial design stages, so there are several avenues for future work needed to mature the tool. So far, the tool has only been applied to activity-type diagrams, OV-5 diagrams in the DoDAF framework. The chronological nature of these diagrams makes them easy to re-create as a discrete event model. Within the DoDAF framework and others, activity diagrams are used as a basis for several other types of diagrams, so by optimizing the activity diagrams it would have a ripple effect of improving the entire architecture. Applying the architecture optimization tool to other diagrams is also being examined.

It has been found that identifying which aspects of the architecture to make variable is often a challenge. Not thinking of the architecture as fixed structure is a shift from the traditional approach to architecture design. It can require a different thought process to look for parts of the architecture that might be flexible or improvable. To assist with this, it would be beneficial to collect a library of variable architectures to serve as a reference to designers attempting to use this tool. At the same time, the more variables there are in the architecture the more processing time it will take to fully analyze all the combinations. Isight's multiple optimization algorithms helps alleviate this problem by intelligently exploring the design space. Distributed processing is also being considered as a means to reduce the time it takes to come up with an optimized solution for a complex architecture. Progress in each of these areas of future work will ultimately provide additional benefits to the tool's application.

8. Conclusion

Architecture is becoming increasingly important in the design of modern, complex systems. Careful consideration for the structure of a system's architecture is justified by the potential impact it can have throughout the rest of the system's design process and life cycle. Unfortunately, traditional approaches to architecture design selection may not always identify the candidate that will best meet the system's performance requirements. Executable architecture models facilitate quantitative evaluation, and can now be automatically generated from architecture diagrams in standard architecture design tools. By automatically generating multiple executable models in the Isight optimization environment, various architecture alternatives can be

evaluated and objectively compared. This architecture optimization tool investigates novel architecture constructions that might not otherwise be considered, and in so doing can find unexpectedly successful designs. It requires a new approach from architecture designers, but it has the potential to expedite the design process, and minimize rework. More importantly, employing the architecture optimization tool provides assurance that the architecture selected will best meet the customer's needs.

9. References

1. Department of Defense Architecture Framework Working Group, "DoD Architecture Framework Version 1.0," vol. I & II, 2004.

2. Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," http://www.pithecanthropus.com/~awg/public_html , 2000.