SoS Explorer: A Tool for System of Systems Architecting

Cihan H. Dagli
David M. Curry

Research Task / Overview

System of systems are important to the functioning of modern society. They define infrastructures such as transportation, energy, and healthcare. They are used to achieve specific needs such as military missions as well as future plans for an intelligent transportation system or smart cities. As important as they are, SoS architecting remains a difficult problem due to the interplay of a large number of variables. Modeling can greatly benefit this effort, however modeling is itself a difficult task. Deciding upon a modelling approach and implementing it along with the requisite models is not trivial and can consume significant time and resources. An effective, general modeling framework for systems-of systems architecting would be a significant benefit for the architect.

Goals & Objectives

The objective of this effort is to provide a structure for SoS architecting to guide model development. This structure would also allow a tool to provide interactive visualization, optimization methods, and negotiation models. Interactive visualization would allow the architect to perform "what-if" analysis via a graphical representation of the SoS architecture. Optimization methods give the architect a set of optimal architectures on which to base the final solution. The negotiation models provide insight into how the SoS is likely to be affected by the authorities such as funding agencies.

Data & Analysis

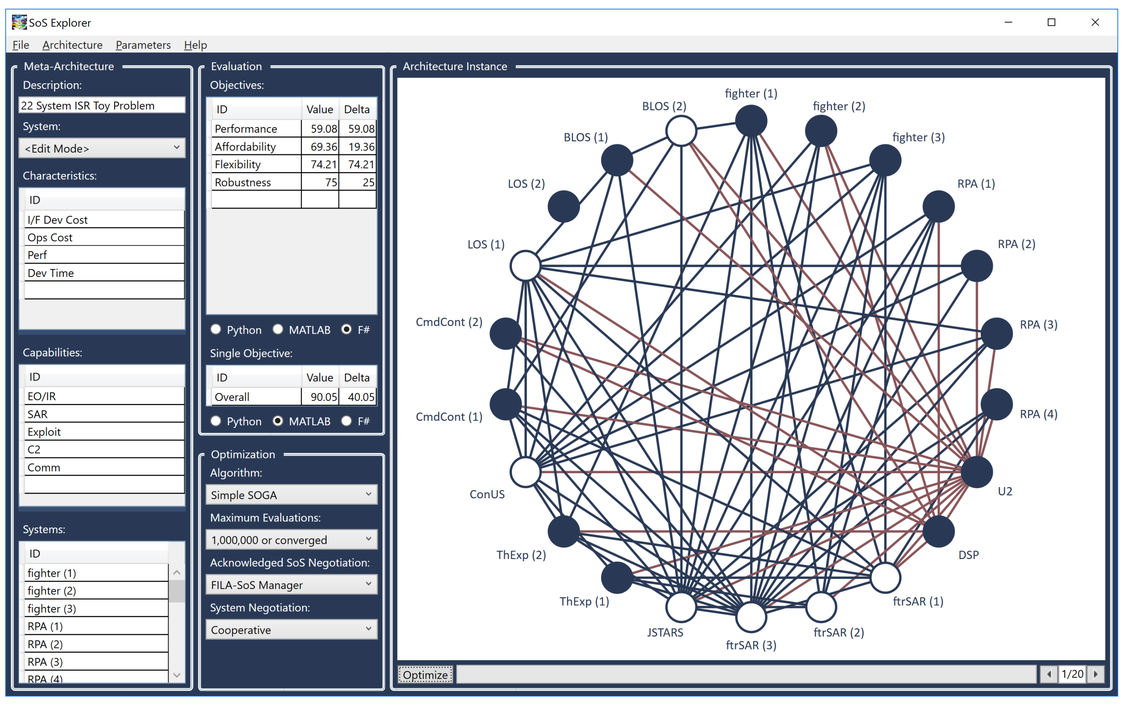
The result of this research is tool called SoS Explorer. The SoS Explorer tool consists of four major components: problem definition, evaluation, optimization, and solutions. The graphical user interface is laid-out in such way as to guide the architect through the steps of defining an architecture using this approach.

The purpose of the problem definition section is to define a meta-architecture. The meta-architecture describes how individual architectures (or architecture instances) are composed. The meta-architecture represents the problem in terms of their characteristics, capabilities, and feasible interfaces.

The evaluation section is where the key performance measures (KPMs) are defined as objectives. The objectives are calculated based upon the characteristics, capabilities, and feasible interfaces of the systems and interfaces selected in a given architecture. The objectives define the modelling required for the SoS. SoS Explorer allows the objectives to be calculated using Python, MATLAB, and F#. An overall objective can also be defined that takes the other objectives as arguments. Its purpose is to enable single-objective optimization. For each objective, there is a "delta" as well as a calculated value. The delta shows the difference in the values between architectures or when an architecture is modified.

The optimization section allows the architect to choose an algorithm with which to find optimal architectures (solutions) and its termination criteria. There are both multiple-objective algorithms and single-objective algorithm from which to choose. Negotiation modelling is planned but not yet implemented.

The solution (architecture instance) section visually represents the architecture solutions. These solutions are maintained as a set and may be paged through, added to, and modified. The architect can interact with the solutions by right-clicking on systems or interfaces and by left-clicking to create new interfaces. The problem and its solutions may then be saved as an Excel-format file.



Methodology

A model should be as simple as possible for the purpose at hand and, at the same time, not artificially restrict the problem being modeled. There are two key components of every SoS: systems and interfaces. Therefore, systems and interfaces will be the basis for the SoS model. Keeping it simple, systems can either participate in the SoS or not. An interface either exists between two systems or it does not. This defines an architecture and can be represented as a chromosome comprised of bit genes for optimization using an evolutionary algorithm. Regarding the individual systems, they are characterized by the following: characteristics, capabilities, and feasible interfaces.

For the system model, characteristics are real valued quantities and can represent items such as the cost of the system, the cost of implementing an interface with another system, performance measures, time to complete, etc. Capabilities are Boolean and represent each system's individual capabilities. These are sometimes referred to as the "little C" capabilities which contribute to the desired overall or "big C" capability required of the SoS. The feasible interfaces are also Boolean and indicate whether it is possible to implement or have an interface between two systems.

The purpose of the system models is to estimate key performance measures (KPMs) for the SoS. The KPMs are used as the objectives when using an optimization method as well as to provide feedback on changes made interactively to an architecture. The only system characteristics, capabilities, and interfaces that need to be considered are those that effect the KPMs used to measure the performance of the SoS.

Future Research

- A negotiation model could allow for competitive, semi-cooperative, and cooperative negotiating between the systems and a managing authority.
- Significant performance gains could be had by optimizing the meta-architecture chromosome by removing genes representing infeasible interfaces and required systems.

Contacts/References

Cihan H. Dagli
Department of Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri 65409-0370
Email: dagli@mst.edu

David M. Curry
Department of Engineering Management and Systems Engineering
Missouri University of Science and Technology
Rolla, Missouri 65409-0370
Email: dmcfh3@mst.edu