EXPERIMENTATION TESTBEDS: USING SORASCS TO RUN AND PROCESS HSCB VIRTUAL EXPERIMENTS

Kathleen M. Carley,[1] Michael W. Bigrigg, David Garlan, Michael Lanham, Yue Lu, Geoff Morgon, Bradley Schmerl
Carnegie Mellon University

ABSTRACT
SORASCS is a service-oriented-plus framework for distributing and composing workflows composed of services and thick clients to support data processing, analysis and experimentation. We describe how SORASCS can be used to support "what-if" reasoning through the design, running, and analysis of the results from a virtual experiment using an HSCB simulation model. Given the experimental specification, SORASCS conducts the experiment using a distributed grid platform, compiles the results, and provides tracking information for the end user. SORASCS enables the user to rapidly move from raw data to analysis to forecasting and back; thus supporting model reuse and sharing. This process is described and an example using a border conflict experiment is shown.

PRIMARY TRACK
MMT

SECONDARY TRACK
Valid Model Use and Validation

DESCRIPTION

Reasoning about the impact of courses of action can involve running virtual experiments using simulation models. Experimentation control is critical. We describe how SORASCS can be used to set up and manage virtual experiments. This process is illustrated using a specific experiment that can be, and has been, run with and without SORASCS to explore the impact of various courses of action in deterring hostilities in a hypothetical border-conflict between India and Pakistan. Our focus is not the conflict or the virtual experiment, but the process of experimental control as it supports a multi-modeling approach.

An important aspect in HSCB analysis is the ability to design virtual experiments for what-if reasoning using simulation systems. Such what-if reasoning requires the analyst to use a simulation model of the domain to be analyzed, instantiated with real-world data that specify the parameters, events, and other data of interest, and to run a virtual experiment specifying conditions and then statistically analyze the results. Since most HSC simulation models have a stochastic component, the model needs to be run a large number of times. To speed processing, this is often done by farming out or distributing the instances of the simulation across multiple processors, tracking performance, and then re-collating the results.

To facilitate model re-use and the development and validation of these simulation models a number of challenges need to be met. These challenges range from auto-instantiation of models

[1] Dr. Carley, due to prior engagements is only available on the 8th. If the presentation is on a different day another member of the team will present.

in novel situations to tracking what model was used with what data to produce what results and so enable re-assessment. Placing the simulation model in an environment for managing workflows can help meet these challenges. Yet, from a purely workflow perspective there are a number of challenges faced in adding a simulation to a workflow that are distinct from adding a data-extraction, analysis, or visualization tool. These work-flow challenges include, setting up the experimental design, establishing support for tracking models, inputs and outputs, handling modeling hierarchies, supporting the various types of interoperability needed for multi-modeling, and supporting distributed processing and data-reintegration. To meet these challenges SORASCS was extended to support the management of virtual experiments.

SORASCS [2] is a service-oriented plus framework for the HSCB community that allows tool providers to integrate HSCB tools as either web services or as thick clients and to run them in a distributed environment. Analysts then construct workflows to automate and manage common sequences of operations. One such workflow is gathering data, analyzing it, then conducting a virtual experiment using a simulation tool as part of the course of action analysis.

Simulation models are an interesting workflow challenge as each model is typically built within a simulation framework and then refined for a specific context. Re-use often requires altering the context, the input, and even processes. How then can such models be integrated into workflows? Our proposed solution is to use a principled specification hierarchy for the incorporation of simulation models.

We find that in the majority of HSCB simulation models there is a hierarchy of increasingly specified models built on top of a more general version of the model. Integrating the simulation at the most detailed level removes the possibility of reuse in different contexts; but integrating at the highest level places all the burden of model design in a new context on the end-user. Potentially different users will want to use models at different levels in this specification hierarchy. Consequently, the SORASCS system is designed to support the hierarchical integration of models and to allow users to build and share workflows at different levels in this hierarchy. Alternative views, management and project, enable different users to examine and use models at different levels in this specification hierarchy.

In addition, in order to support the design and execution of virtual experiments, SORASCS provides the following capabilities:

a. A common integration infrastructure that allows different simulation tools to be used. Simulation tools are integrated into SORASCS as simulation services. SORASCS provides a general, easy method for tool providers to integrate their simulators into SORASCS so that they can be used in virtual experiments.
b. A virtual experiment design frontend that can be customized to different virtual types of virtual experiments. For example, experiments for border conflict analysis or knowledge diffusion within a group can be built that use particular models and simulators. These types of virtual experiments can be further specialized to certain cases (e.g., border conflicts of India and Pakistan, or of Northern and Southern Sudan). Each kind of specialization can be shared.
c. Data services for storing and organizing the results of simulations, as well as for storing the details of each experimental run so that it may be executed again in the case of new data, or to get more results.
d. The ability to link to a distributed platform, e.g., a CONDOR cluster, for processing. This allows multiple simulations to be run simultaneously on a cluster of machines managed by SORASCS, and their results collated in a common location, providing

scalability. Typically, an analyst will want to run numerous such simulations (especially if the simulations are statistical), and with different parameters (so that the effect of different starting conditions and events can be compared). By using a cluster environment, the simulations can make use of a more realistic model.

e. The ability to put a simulation in a workflow and take another models output as input, and to use the same workflow with alternative simulation tools. By supporting multiple types of interoperability SORASCS supports multi-modeling.

Currently, we have demonstrated this process using the Construct simulation system. Construct [1][4], a multi-agent dynamic-network model for assessing the impact of interventions on the flow of ideas, beliefs, and social activity. Construct takes into account how agents learn through interaction and from diverse media predicated on their socio-cultural position in a meta-network. As individuals learn they change what information they know, what they believe and what they will do. The spectrum of experiment specificity that can be saved at any time includes a general Construct experiment [3] and all details on the state of the agents modeled. From a specification hierarchy perspective Construct can be used for a number of different experiments across contexts. Thus we can think of a "deterrence" Construct, a "counter-terrorism" Construct, a "crisis response" Construct and so on, differing by who the agents are, the information being communicated, the beliefs and actions of relevance. At a more detailed level of specification these instances of Construct can be further refined for particular experiments. For example a deterrence Construct can be refined to used data on a border conflict between Pakistan and India during a specific time period. All three levels of specification can be managed and linked to each other in SORASCS.

To run a virtual experiment all levels in the specification hierarchy need to be linked into SROASCS. As such the following activities must be done:

a. The developer provides the simulation model as a service and integrates the simulation model with the SORASCS infrastructure. The developer also provides a specification of the range of parameters that this model accepts. SORASCS provides this information to the experiment designers at more detailed levels of specification so that they can tailor the generic model to their particular domain.
b. An experiment designer further specializes this set of parameters to design a particular kind of experiment (e.g., border conflict). The designer may provide models that set up default parameters for that kind of experiment, as well as specify the subset of parameters from (a) that may be left to the experiment runner to specify.
c. An experiment runner may specify the parameters to run the simulation, including the number of runs and the number of timepoints for the simulation, giving values to the parameters from (b). The experiment is then run by SORASCS.

SORASCS is flexible in the experiment design; at any level, the experiment can be saved and shared. Experimentation parameters can be changed and the experiment submitted for sensitivity analysis, including a change in the model. Using a cluster of computing resources multiple experiments can be run simultaneously with results collected in a consistent manner aiding the

The Border Conflict experiment used 3,000 open source news articles and 24,000 government websites which were processed to create the meta-network of 2,535 entities (agents, organizations, locations, knowledge, tasks, resources, and events) used as real-world input to the simulation. Three vignettes representing actions before, during, and after conflict, with each vignette having a different number of experiments: 2, 8, and 4 respectively for before, during, and after conflict. These 14 experiments were each run 100 times for 300 time periods.

By providing an integration architecture, a flexible user interface for designing experiments, and a means to run experiments in a distributed setting on a grid cluster, the Border Conflict experiment was completed in under a month using real-world data. The experimentation starting from a completed real-world meta-network data, through trial runs, and finally production runs was completed in under two weeks.

ACKNOWLEDGEMENTS

This work was supported in part by the Office of Naval Research - ONR -N000140811223 and by the Air Force Office of Sponsored Research – MURI with GMU Cultural Modeling of the Adversary, 600322. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS). The specific experimental design and the data were originally developed under the AFOSR MURI with GMU. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the Air Force Office of Sponsored Research or the U.S. government.

BIOGRAPHY

Kathleen M. Carley is a Professor of Computation, Organizations and Society and the center director for Computational Analysis of Social and Organizational Systems (CASOS) in the Institute for Software Research, School of Computer Science, at Carnegie Mellon University. Her research combines cognitive science, social networks, agent-based modeling and computer science to address complex social and organizational problems. Specific research areas include dynamic network analysis, computational social and organization theory, adaptation and evolution, text mining, information diffusion and belief dispersion, disease contagion, command control, and disaster response. She and members of her center have developed and deployed multiple tools. She founded the journal Computational and Mathematical Organization Theory, co-edited several books and has over 250 publications in the computational organizations and dynamic network area. In 2001 she received a lifetime achievement award in sociology and computers and is the 2010 winner of the Social Network Simmel award.

REFERENCES

[1] Carley, K.M., Martin M.K. and Hirshman B., 2009, “The Etiology of Social Change,” Topics in Cognitive Science, 1(4):621-650.

[2] Garlan D., Carley K.M., Schmerl B., Bigrigg M. and Celiku O., 2009, Using Service-Oriented Architectures for Socio-Cultural Analysis. In Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE2009), Boston, USA, 1-3 July 2009.

[3] Hirshman B. and Carley, K.M., 2009, Variables, Decisions, and Scripting in Construct. Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report CMU-ISR-09-126

[4] Schreiber C. and Carley K.M., 2004, Construct - A Multi-agent Network Model for the Co-evolution of Agents and Socio-cultural Environments. Construct - A Multi-agent Network Model for the Co-evolution of Agents and Socio-cultural Environments. Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report CMU-ISRI-04-109.