

DATASEM: A Simulation Suite for SoSE Management Research

Richard Turner
Stevens Institute of Technology
Hoboken, NJ, USA
rtturner@stevens.edu

Levent Yilmaz, Jeffrey
Smith, Donghuang Li,
Saicharan Chada, Alice
Smith
Auburn University
Auburn, AL, USA
smithae@auburn.edu

Alexey Tregubov
University of Southern
California
Los Angeles, CA, USA
tregubov@usc.edu

Abstract – The Systems Engineering Research Center (A US DoD University-Affiliated Research Center) has been researching systems engineering and management problems in the evolution of Systems of Systems (SoSs) since 2011. In 2015, an initial Demonstration and Analysis Tool for Adaptive Systems Engineering Management (DATASEM) was developed to investigate how various combinations of organizational structure, work flow, and governance mechanisms affect the visibility, flow, and overall value produced in developing and evolving SoSs. This paper provides an overview of the “as developed” initial system, plans for improving and validating the system, lessons learned and early results.

Keywords: *System of systems, agile, adaptive, management simulation, scheduling, kanban, lean, value-based, software engineering, systems engineering.*

1 INTRODUCTION

Complicated, large systems of systems in rapid or continuous deployment environments, where requirements are not precise and can change or emerge quickly, find traditional approaches inadequate. Our initial research as documented in [1-7] has provided a set of agile and lean governance concepts that represent possible solutions to some of the problems associated with managing and evolving Systems of Systems. Key to this research was work by Reinertsen [8] and Anderson [9]. Initial goals for these mechanisms include:

- Coordinate multiple levels of development activity across multiple system components with diverse and possibly disjoint or isolated development groups
- Support analysis and decision making at every level
- Flexibly schedule work considering value across the system of systems
- Balance work in progress (WIP) across resources with SoS organizational capacity to improve flow
- Make visible to all levels progress toward capability development and deployment

- Establish a basis for continuous improvement in a rapidly changing environment

Given the number of combinations and complex interaction among workflow, organizations, environments, governance mechanisms and strategies, validating and experimenting with these concepts in vivo is difficult at best. The solution was to develop a broad simulation environment to allow in vitro experimentation with such mechanisms, singly and in concert, operating across a range of organizational structures and handling different kinds, durations, complexity and volumes of work flow. We believe establishing statistically significant evidence across various combinations of mechanisms, organizations and work flow, as well as providing a suitable simulation “sandbox” for adopters to perform their own experiments, will provide a higher level of confidence than piloting, is low risk, and provides more value for money to adopters.

I. DEMONSTRATION AND ANALYSIS TOOL FOR ADAPTIVE SYSTEMS ENGINEERING MANAGEMENT (DATASEM)

A. Concept Development

The initial DATASEM concept was to create a means of modeling organizations, workflows and governance mechanisms and then simulating how they interacted with each other. The decision to use an agent-based simulation was made based on the desire for flexibility describing people responsible for doing managerial, technical and analysis work, making decisions about value, services, accepting or assigning work, negotiating, and other human activities. Although it required additional overhead, and had more power and functionality than we initially needed in the simulation, Repast Symphony was the open source product chosen because of its popularity and usage in academia, and previous successful experience by members of the team using it. Somewhat later, it was discovered that Repast lacked a web-based interface, which led to unexpected work and limited the capabilities of the initial web-based version delivered. Figure 1 provides an overview of the general flow of the DATASEM suite.

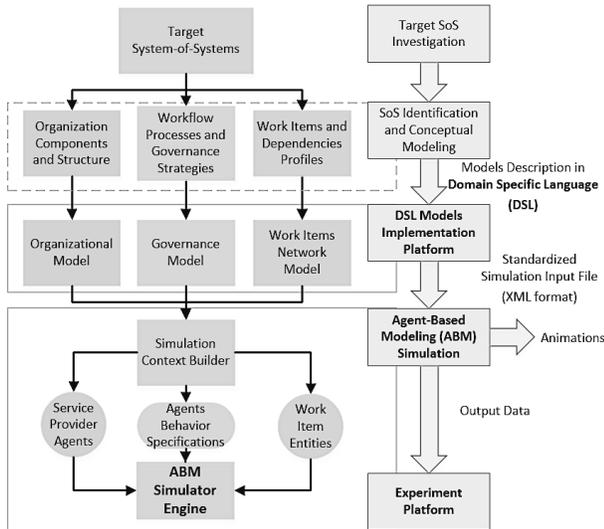


Fig. 1. Overall DATASEM Flow

B. Architecture

Two versions of the DATASEM Software have been developed. One is a web-based implementation and the second is a standalone version. Both are represented in the overall architecture of the initial DATASEM Suite, is shown in Figure 2. The standalone version of the DATASEM can also be installed in developers' mode in Eclipse IDE. This installation includes two projects (Repast simulation engine and Exttext/Extend Modeler) in Eclipse IDE. From Eclipse IDE users can Edit and compile DSL as well as change its grammar and change Java implementation of the agents and governance mechanisms.

The Web application of the DATASEM consists of three major components (groups of modules).

DES Framework web application – Provides UI and connects all other external components and includes the following:

1. Web front end (UI) and corresponding back end modules for UI
2. Database of DSL models
3. Database of experiments
4. Database of simulation results
5. Experiment builders—these include adaptors for DSL code compiler (such as Exttext Modeler)
6. Repast adapters: modules that execute Repast simulation engine and collect/convert simulation results (orange modules on the diagram).

Repast simulation engine—a simulation engine that implements governance mechanisms in Repast Symphony simulation framework. This module is an independent application, and can be used as a standalone application.

Exttext Modeller – a compiler for DSL code. DSL code is compiled into xml file of the experiment scenario (Standardized Simulation Input file on Diagram 1). XML files then stored in database and be then executed in Repast without need to recompile them again.

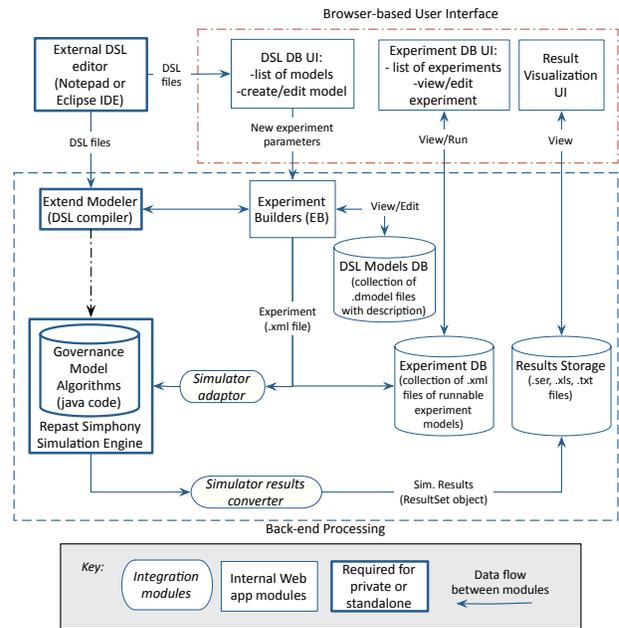


Fig. 2. DATASEM Architecture

The standalone version of the DATASEM application consists of Repast simulation engine and Exttext Modeler packed together in package. The standalone version of the DATASEM does not have an embedded DSL editor, so user's can either set up Eclipse IDE with Exttext/Extend Modeler plugins or use any text editor.

Currently the user interface is limited. Several potential designs were developed, but the continuing evolution of the models and the implementation of the DSL tool left the user interface essentially a DSL-based editor. Outputs are also minimal, although full .xls and .csv files of the simulation results provide for external tools to be added. The user interface and output formats are near the top of the list for next steps.

II. THE DSL

Early in the design process, the use of a DSL was proposed to provide both a semi-formal and therefore verifiable description of an experiment and as a means of extending the simulation and ensuring common definitions and functionality. As developed, the DSL has come to be an important means of both defining the mechanisms more fully and in identifying the significant interactions between the three models and the dependencies among them.

Originally planned as three separate DSLs, one for each aspect – work flow, organization and governance – the DSL actually evolved into a single language that also captured the intentions and mechanisms for the experiment. This makes it a complete representation of the experiment in a single artifact. The DSL took much more time to develop than initially thought, and is still significantly evolving in syntax, semantics, and implementation within the simulation. As expected, it has had a significant impact in how the team approached the software development, but the implementation has also fed back information on the structure of the DSL. Table 1 shows the basic format of a complete DSL experiment

TABLE I. DSL MODEL FORMAT

Section (Subsection)	Description
Model	<i>Name of the model (reserved for later use).</i>
User Libraries	<i>Defines basic model components that are referenced later in creating the experiment.</i>
<i>Service Provider Types</i>	This describes the general Service Providers that can be defined in the model. These are generally used to identify the type of organization (for example a prime developer team, a development contractor, or a specialty service organization). These can be organized in some form of hierarchy, but all providers of the same type should be assigned the same hierarchical level. Service Providers are essentially the organizational building blocks. They can be complex or simple. In some cases, such as an expert that you want to model as a service, an a service provider could be a single individual.
<i>Work Item Types</i>	This describes the kinds of work addressed by the various services. For example, a development task, an analysis task or a problem resolution task could all be types of work items. These definitions also describe hierarchical relationships such as decomposition and associated value inheritance. For example, a 'Capability' work item defined as the highest level, could be decomposed into 'Requirements' work items (each requirement might inherit some percentage of the Capability's value), which might be decomposed into Component or Feature work items (again perhaps inheriting some value), and so forth down to the smallest task modeled. The level, distribution and weight of such relationships are described in attributes associated with each Work Item Type.
<i>Services</i>	Each service, for example a development activity, required for completing the work items or governance mechanisms are named and described. Used to characterize skills for organizational resources.
<i>Governance Strategies</i>	Each strategy is identified by type (currently pull or push), the specific mechanisms implemented, and the attributes associated with each of the mechanisms.
Experiment Model	<i>Defines in detail the specific pieces of the model and how they interact with each other. It also provides information for how the experiment is to be run and what outputs are desired.</i>
<i>Variables</i>	These are used as references within the Organizational and Work Item models to provide constants or multiple types of stochastic values/distributions.
<i>Organizational Model</i>	The Organizational Model defines each of the specific Service Providers in terms of a name, Service Provider Type, Governance Strategy (and any governance attributes for this organizational component that are different from the default definition), internal resources (staff or other assets) and other service providers that may provide services to them. Staff/asset resources are characterized by a number of attributes, including Services they can perform and level of expertise as defined by their efficiency working on those services. It is possible to use stochastic distributions to generate attributes.
<i>Work Item Network Models</i>	This section describes the specific work that needs to be accomplished. Work Items are described to the level of detail required. Work items are defined by work item type, service(s) required and effort, value (if assigned or inherited) probability and degree of impact on other tasks, prerequisites (if appropriate), arrival time (if appropriate). Some

	work items are aggregations of lower level activities, but may require additional work to "analyze" at the higher level to determine how the decomposition will happen (such as described earlier in the work item type and hierarchy) . It may be that this breakdown is determined by the model with a specific structure, or it may be through stochastic guidance provided to generate sublevels of work.
<i>Experiment Parameters</i>	These are primarily associated with the simulation run and the user output expected. For example, the mode of the experiment (batch vs, single run) number of times to run the simulation (Monte Carlo), work uncertainty, rework probability distribution, change propagation characteristics, learning factor, graphic specifications (for the online version), multi-tasking penalty.

The full description of the language is captured in the evolving software and in the *DATASEM Domain Specific Language Reference*. [10]

III. VALIDATING DATASEM

A key issue in the success of DATASEM as a tool is the ability to validate that the results are indeed aligned with reality. Significant consideration has been given to this problem

In parallel with the development, Dr. Forrest Shull of the Software Engineering Institute of Carnegie Mellon University was designing an approach to this problem. In [11] he states

Evidence does exist concerning the utility of Kanban and KSS principles in relevant environments, but more work needs to be done to quantify the expected effects and the contexts in which they can be expected to hold. The vast majority of the empirical evidence found relies on anecdotal reports or on qualitative data drawn from interviews and surveys with developers. While important, such qualitative evidence may represent the subjective views of the developers more than the underlying reality. The minority of studies that were identified that do collect quantitative data from real environments do corroborate the qualitative evidence, but represent only a few development contexts with no claim of wider applicability.

In his framework, he articulates four goals of empirical study in this domain and proposes a specific study design capable of satisfying each goal. The goals build upon each other in a progression to show what could be accomplished using the constructed simulation tool:

1. Validate that the simulation capabilities and the Domain Specific Language are capable of capturing a given system engineering environment with sufficient accuracy.
2. Once validated for some environment and scenarios, the simulator can then be used to study any changes involved in moving from a more traditional system engineering approach to one based on adaptive mechanisms. In particular, the metric of product value

delivered over time was identified as a key way to capture the effect of value-based pull scheduling.

3. Further refinements in the understanding of adaptive principles can be achieved by altering some of the key parameters in the scenario while keeping the vast majority the same, and watching the impact on product value delivered over time and other key metrics. In short, this would allow an understanding of the contexts under which adaptive principles do or do not provide value.
4. For a given scenario, an analyst could do a deeper dive into very specific values produced by the simulation (e.g. resource usage, Work Item delay) to understand where bottlenecks exist and where the system could be fine-tuned to deliver more value.

IV. DISCUSSION

This section describes lessons learned, evaluation of status, and potential for DATASEM-related work in the future.

A. Significant Lessons learned during the DATASEM Development

Given the work accomplished in previous research, there was considerable overconfidence and a general underestimation of the development difficulty. While some was due to the normal ramp up of new researchers and considering new concepts (such as creating a DSL) that would make the tool more easily extensible, there were some humbling but enlightening lessons learned.

1) Impact of the interaction of governance strategies, organizational models and work flow definition in Building experiments

When initially considering the simulation suite, we believed that the three activities we were interested in modeling were essentially independent and that we could easily mix and match different types. As we actually began to develop the DSL and the implementation of the various mechanisms, it became clear this was not the case.

For example, the technical process generally defines how work is partitioned from higher levels of abstraction to lower implementation levels. This drives both the batch size in terms of scheduling, and the number of resources applied to work items. In the case of a Scrum-like technical management process, there is a significant effort to fit work into the sprint length. In kanban systems, some attempt is usually made to normalize the size of work to some extent. On the other hand, in a traditional development with significant sub-contracting or more specialty teams, work may be packaged in larger batches for ease of contracting. The variation in batch sizes and abstractions makes value determination much more difficult and comparisons less convincing.

The technical process also impacts how a work item's status towards completion is determined. Many technical processes are iterative and so may cycle through a number of intermediate versions of a task. The indeterminate number of cycles are more difficult to represent in a work item network. They may also complicate value determination.

The same governance strategy may require very different mechanisms depending on the organizational definition. Contractual boundaries require more ceremony, and in the case of organizations who adopt some form of bidding process for work, the idea of pull scheduling takes on a completely different flavor than that associated with a software development team or of an implementation of systems engineering as a service.

The experimental design and output specification also had interactions with the models and the way they needed to be specified in order to gather comparable data. This was particularly true if there were significant differences in the resulting work item network.

We discovered that the order in which decisions on organization and governance structure are made can affect the ability to create a reasonable work item network. This changed our original approach to user interaction with the system and exacerbated a number of design activities, including those in the following sections.

2) Difficulty in rigorous definition of governance strategies and mechanisms

One of the most frustrating realizations was the difficulty in concretely describing the various governance strategies and mechanisms. The diversity of researcher backgrounds, the unexpected vagueness of the concepts, and the distribution of the team in three areas of the country added to an already challenging situation. The team believed that the material developed in the previous research was sufficiently well-defined to create the simulation. Unfortunately, as the scope of DATASEM expanded from a specific governance strategy to a more general sandbox for experimentation, we realized that our internal definitions and understanding were not easily captured in more general ways.

It was not until the task's last few months that it became obvious the team was in many ways reenacting the story of the blind men and the elephant¹. Each of us had established a mental model to work with, but we were all approaching different parts of the problem and proceeding from different points of view. These disconnects in understanding each other's mental models became much clearer as we attempted to build DSL examples of the KSSN concept. Flow control mechanisms such as pull scheduling, classes of service and limits on work in progress were not implemented in a way that represented the earlier work. What had been implemented, however, represented significant mechanisms that had not been explored earlier. One example is the use of the Contract Net Protocol as a basis for pull scheduling. While it does not capture some of the flow control concepts

¹ In this ancient parable, a group of blind men encounters an elephant. To describe it, each touches one part of the animal. The man who grasped the tail said, "It is much like a rope." The man who touched the side said, "No, it is more like a wall!" A third man, having grasped the trunk replied, "You are both wrong. It is obviously much more like a snake." "No, not at all," said a fourth, who had found the leg. "It is most surely like a tree." Unable to resolve their differences, they went off—none the wiser about the elephant, but each convinced they understood it better than the others.

of kanban, it does provide a terrific foundation for modeling negotiation.

Because the discussions around interpretations happened late in the development process, we were not able to adjust the DSL or implementations to do a full model of KSSN for this release. We have, however, gained a great deal of insight into the issues around these mechanisms, and are seeking ways to unify our differing visions to create a broader, more useful DATASEM implementation.

3) *Tension between elegance, extendibility, and practicality of the DSL*

The initial DSL was very elegant, but provided little concrete concepts to work with. This was thought to be reasonable given that we wanted an extensible model and elegance generally allows evolution to be relatively painless. However, as we learned more about the intricacies of the ideas we were going to investigate, more and more of the detail was ending up in the implementing java code than in the DSL. We felt that the need for systems engineers to be able to understand what they were modeling needed more explicit expression in the DSL.

We learned fairly quickly, though, that implementing more detail in the DSL could not be done without looking back at the whole language. We are aware of a number of shortcomings in the DSL in terms of consistency across definitions and in the sense of structure and design of the language. Reviewing and revising the DSL in light of this first round of development is a high priority for the next phase.

4) *Complexity of interactions between organizations*

One of the uses identified for DATASEM is investigating the human and managerial aspects of systems of systems. The use of an agent-based simulation engine was driven primarily by this desire to model the human components of the system. Developing the DSL proved hard even for the mechanical complexity of the governance, organizational and work flow representations. We readily identified places where behavior modeling would be valuable, but we were not able to address it and still deliver something useful within the period of performance. Areas where we believe that behavior modeling can be added include:

- Negotiation of services and contracts
- Interface of pull and push governance mechanisms
- Value determination and agreement
- Conflicts due to differing value systems and stakeholders between the system of systems level and the constituent system level
- Stakeholder/constituent behavior

5) *Limitations and complications of agent-based simulation*

In retrospect, Repast provided much more sophisticated simulation capabilities at the cost of not providing more useful capabilities (like a web interface) for this stage in the modeling and simulation work. The availability of all the

agent capabilities was important to make sure we could include the human aspect. However, it forced us to think more in terms of agents than of mechanisms, which simplified the concept into service providers and service requestors, but complicated some of the design work. It also affected defining the mechanisms.

B. *The Impact of the DATASEM Project and Related Research*

Initially motivated by the ineffectiveness of integrated master schedules in rapidly changing operational environments and the success of Kanban approaches for the knowledge work of software development, the project has expanded to investigating kanban as well as other adaptive governance mechanisms applicability. The initial Kanban scheduling system (KSS) networks described in [4] sought a way to prioritize engineering tasks based on SoS or complex system capability priorities and task interdependencies by selecting value-adding features first, reducing wait time for scarce engineering specialties, and minimizing time wasted on context switching by overloaded resources. The KSS network concept provides two valuable side effects. First, the implementation of the network supports critical conversations about schedule and value decisions by the appropriate people at the right time and nearest the actual implementation. Second, the network significantly improves executive and systems engineering visibility into the status of multiple independent development organizations.

V. CONCLUSIONS

The development of DATASEM and the DSL provide a flexible, available, and inexpensive way to research how governance, organization, and work characteristics impact flow through an organization. New concepts can be mixed and matched with traditional concepts. Various types, sources, and cadences of workflow can be combined or assessed independently. Research can be conducted on the predictive nature of more subtle measurements for identifying trends, on improving the accuracy of status, and on understanding the impact of assigning, monitoring and using value, work load and other characteristics.

The work provided insight into the definition of suitable adaptive mechanisms and the simulation of their implementation. While still in process, it is clear that what we thought were separable pieces of a management system – the work, the organization, and the governance model – when viewed through the lens of adaptive management processes become highly interdependent. This has significant implications and requires us to rethink in a more holistic way how the mechanism definitions, DSL and simulations will need to be refactored to better address this interaction as well as capture the relationships in further experimentation.

REFERENCES

- [1] Turner, R.; "A Lean Approach to Scheduling Systems Engineering Resources," CrossTalk, May/June, 2013.
- [2] Turner, R.; Madachy, R.; Lane, J.A.; Ingold, D.; Levine, L.; "Agile-Lean Software Engineering (ALSE): Evaluating Kanban in Systems Engineering," A013 - Final Technical Report SERC-2013-TR-022-2, March 6, 2013.

- [3] Lane, J. A. and Turner, R. "Improving Development Visibility and Flow in Large Operational Organizations," *4th International Conference on Lean Enterprise Software and Systems*, Galway, Ireland, December 1-4, 2013, Proceedings, [Lecture Notes in Business Information Processing](#), Vol. 167, pp 65-80, Springer-Verlag, Heidelberg, 2013.
- [4] Turner, R., "Value-based scheduling in system of systems evolution," *System of Systems Engineering (SOSE), 2014 9th International Conference on*, pp.301,306, 9-13 June 2014.
- [5] Turner, R.; Yilmaz, L.; Smith, J.; Donghuang Li; Chada, S.; Smith, A.; Tregubov, A., "Modeling an organizational view of the SoS towards managing its evolution," *10th System of Systems Engineering Conference (SoSE), 2015*, pp.480,485, 17-20 May 2015.
- [6] Tregubov, Alexey and Jo Ann Lane, "Simulation of Kanban-based Scheduling for Systems of Systems: Initial Results," *Procedia Computer Science*, Volume 44, 2015, Pages 224-233.
- [7] Boehm, B; Koolmanojwong, S.; Lane, J; Turner, R.; "Principles for Successful Systems Engineering," *Conference on Systems Engineering Research*, Mar 19-22, 2012, *Procedia Computer Science*, Elsevier, Vol. 8, 2012
- [8] Reinertsen, Donald G. (2010). *The Principles of Product Development Flow*. Redondo Beach, CA: Celeritas Publishing.
- [9] Anderson, David, et al. "Studying Lean-Kanban Approach Using Software Process Simulation." In *Agile Processes in Software Engineering and Extreme Programming*, Volume 77 of the series *Lecture Notes in Business Information Processing*, Sillitti, Alberto; Hazzan, Orit; Bache, Emily; Albaladejo, Xavier (eds.) pp 12-26. Springer, 2011.
- [10] *DATASEM Domain Specific Language Reference*, SERC 2015-TR-111B, Dec 2015.
- [11] Shull, Forrest, "A Framework of Empirical Studies of Kanban-Based Scheduling for Software-Intensive Systems," *CMU/SEI-2015-TH-010*, December 2015.
- [12] Ramsey, Marc and R. Levitt, "A Computational Framework for Experimentation with Edge Organizations," *10th International Command and Control Research and Technology Symposium*, 2005.
- [13] Levitt, R.E., "Computational Modeling of Organizations Comes of Age," *Journal of Computational and Mathematical Organizational Theory*, 10(2) (2004), pp.127-145.