Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L.Yilmaz, S.Buckley, and J. A. Miller, eds.

AGENT-SUPPORTED SIMULATION FOR COHERENCE-DRIVEN WORKFLOW DISCOVERY AND EVALUATION

Okan Topçu

Department of Computer Engineering
Naval Science and Engineering Institute
Naval Academy
Istanbul, TURKEY

Levent Yilmaz

Computer Science and Software Engineering
Samuel Ginn College of Engineering
Auburn University
Auburn, AL, USA

ABSTRACT

This article proposes a generic agent-supported symbiotic simulation architecture that generates and evaluates competing coherence-driven workflows using genetic programming. Workflows are examined by an agent-supported multi-simulation environment that allows inducing variation and assessing the outcome of the candidate workflows under a variety of environmental scenarios. The evaluation strategy builds on a coherence-driven selection mechanism that views assessment as a constraint satisfaction problem. The proposed system is also based on an introspective architecture that facilitates monitoring the activities of competing agent simulations as well as the status of the environment to determine the success of the candidate workflow with respect to given or emergent goals.

1 INTRODUCTION

Solutions to strategy-based problems observed in diverse applications such as decision support, command and control (C2), and business process management are often guided by well-defined workflows (Georgakopoulos, Hornick and Sheth 1995; Leymann and Altenhuber 1994; Jablonski and Bussler 1996). Each workflow is comprised of functionally or temporally related tasks supporting a common goal. For instance, C2 activities involve the submission of battle orders that conform to the (operation) plan to the subordinate commanders for a specific mission, and receiving the reports that support control function. Each order, plan, and report contains (military) tasks, which create a workflow. In an uncertain environment such as a battlefield, crisis, or a disaster, it is important to evaluate and to understand the facilitation relations between the tasks and the goals to improve the resiliency of a workflow.

The domain of exploratory (Davis 2000) and generative modeling (Mitchell and Yılmaz 2008) focuses on conducting computational experiments in the presence of deep uncertainty and ambiguity in decision-making. Seeking strategies that perform well across a large number of plausible, yet unknown future states is especially critical. Similarly, in strategic planning, planners seek to discern robust workflows and mechanisms that are capable of carrying out targeted mission scenarios under both foreseeable and unforeseeable disruptions caused by not only the environment, but also the opponents. We conjecture that simulation systems that allow us to devise a generative form of modeling derived from synergistic integration of creative evolutionary dynamics (Bentley and Come 2001) and creative cognition (Finke, Ward, and Smith 1996) can improve our ability to discern resilient workflows.

The ability to instantiate, generate, transform, execute, and if necessary, evolve multiple models of interacting workflow mechanisms, in parallel, all of which take similar but slightly different perspectives (e.g., parallax view) on the same referent system, opens the door to the automatic generation and selection (by falsification) of many somewhat different hypothetical, including non-intuitive workflows. Such an exponential increase in model and hypothesis throughput would promote creative discovery of strategies and increase opportunities for creative leaps.

To this end, to support workflow discovery, we propose a coherence-driven realization strategy of a generic autonomic introspective simulation system architecture. The strategy enables us to evaluate mission objectives, to assess plan tasks, and to select the most coherent workflow among the competing workflows. The system evaluates competing workflows that aim to achieve specific goals by examining to what extent each task (activity) in the workflow contributes to a specific goal or subgoal and which tasks are more relevant for specific goals. As a result, the system recommends a workflow that contains relevant tasks, the successful completion of which would contribute to the achievement of the goals more than others in order to improve resiliency in shifting, ill-defined, and uncertain environments.

The rest of the paper is organized as follows. first, we give a background to provide a context for the proposed system. Section 2 presents the conceptual basis for the workflow discovery and evaluation system. In sections 3 and 4, we examine concrete implementation strategies for workflow and evaluation respectively. Coherence theory (Thagard and Millgram 1995) is utilized to provide an implementation strategy. In section 5, we conclude by summarizing our findings and discussing the potential avenues of future research.

1.1 Background

An operation plan refers to future or anticipated operations under the assumptions about the situation at the time of execution (SISO 2012). An operation plan transforms into an operation order when the conditions of execution occur and time of execution is specified. A well-designed plan includes a mission that presents the intended outcome (objectives) from a sequence of tasks (workflow); commander's intent (in case of a command and control domain), and the situation of the environment and the opponents as well as other factors (SISO 2012). Objectives can be interpreted as a goal system that specifies the public goals (Bjornberg 2009) that derive the plan, where the workflow aims to satisfy those goals and subgoals. Sequence of tasks specifies a workflow (schema) that will be carried out by multiple units. A task is a specific activity to be performed at a specific time, at a specific location and by a specified unit for a specific reason.

Each workflow contains one or more tasks to be carried out by multiple units (e.g. ships in terms of a naval operation or agents in terms of an agent-based simulation (Macal ve North 2010)). In a multi-agent environment, each agent has a particular set of goals and actions. Agents plan and coordinate their behaviors motivated by their (private) goals. Thus, a specific part of the overall workflow, which a particular single agent will carry out, specifies an executional plan for that agent that forms its behavioral specification (e.g. its rule set) motivated by goals. Plans cannot be static as they foresee the future in dynamic environment conditions that evolve with time and other conditions.

In this respect, the relation between the activities in the workflow domain and the execution plan in the software agent domain is depicted in Figure 1. Each workflow is executed by an agent in an agent-directed multi simulation, and the relations among the goals and tasks in the workflow are evaluated in terms of coherence satisfaction, which is based on the deliberative coherence theory (Thagard and Millgram 1995). The coherence theory helps us understand which tasks and goals in a workflow fit each other and which ones do not. Fitting together emphasizes the coherence relation such as facilitation (e.g. a task facilitates to achieve a goal) and is represented as a positive constraint. If they are incoherent (e.g. when two goals are in conflict), then the relation is represented as a negative constraint.

The proposed system is also based on an introspective architecture that facilitates monitoring the activities of competing agent simulations as well as the status of the environment to determine the success of the candidate workflow with respect to given or emergent goals. The optimal/robust workflow that will be discovered will be cast as an emergent property, while evaluation and discovery serve as the selection and variation components.

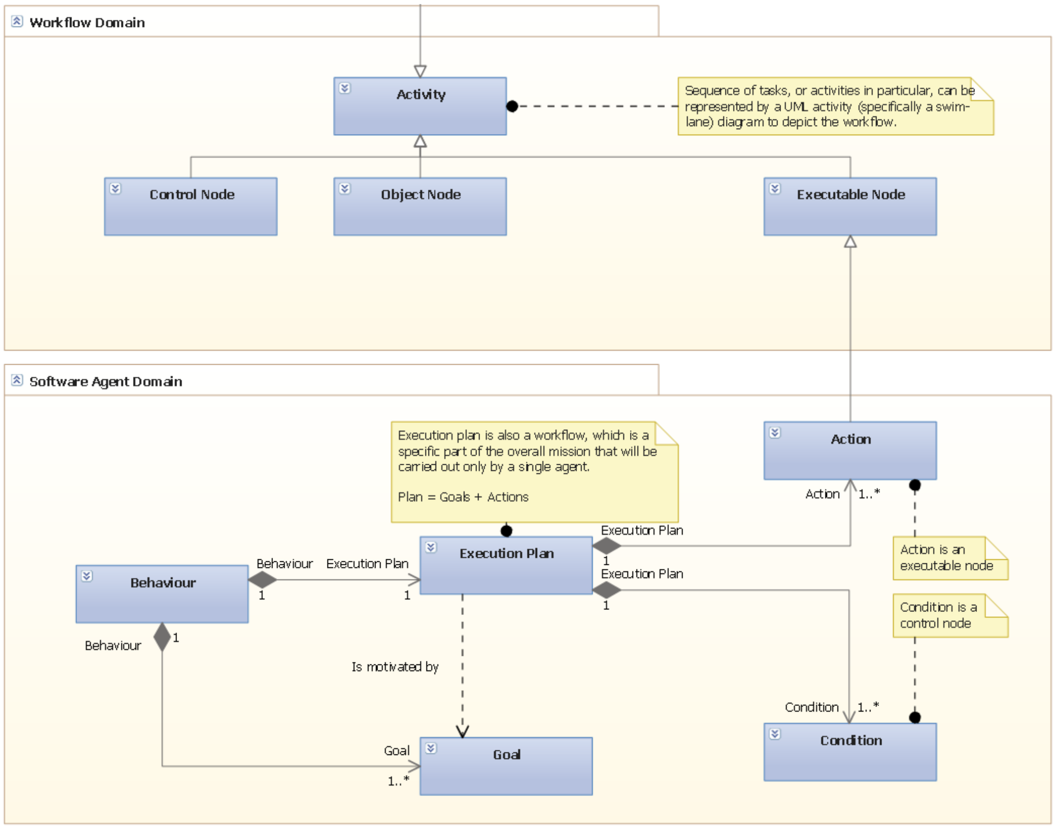

Figure 1: Concepts and Their Relations Among Domains.

2 CONCEPTUAL BASIS AND REFERENCE ARCHITECTURE FOR WORKFLOW DISCOVERY AND EVALUATION

A workflow is modeled using graph-based formalisms such as Petri-nets and directed graphs; or using a general purpose visual modeling language such as Unified Modeling Language (UML) activity diagrams (OMG 2013); or using a domain-specific language such as Business Process Model and Notation (BPMN) (OMG 2011). In its simplest form, using a graph-based formalism, a workflow can be represented as a directed graph (digraph) whose nodes (vertices) are the tasks with edges denoting the flow of control between tasks (Narendra 2004).

As depicted in Figure 2, UML activities are also specified as a graph of nodes interconnected by edges (OMG 2013). Each node corresponds to an activity node. There are three kinds of activity nodes: Control nodes, object nodes, and executable nodes (see Figure 1). The directed arc between activity nodes is called an activity edge. Two activity edges are defined in UML specification: a control flow and a data flow. An instance of the workflow is a scenario that specifies a course of action in execution. Agents perform workflows as part of meeting their goals and can collaborate to perform a workflow (Narendra 2003).

In this respect, the proposed workflow discovery and evaluation system is based on the idea that the primitives of a model specification (i.e., condition-action pairs and their interdependencies) form, evolve, and self-organize to generate plausible competing workflows. Thus, the workflow evaluation process selects the most coherent one based on the Deliberative Coherence Theory (Thagard ve Millgram 1995) and uses it as a behavioral specification of an agent in the simulation subsystem. The coherent workflow contains the relevant tasks, and the successful completion of those tasks contribute to the achievement of the goals more than others in order to improve resiliency.

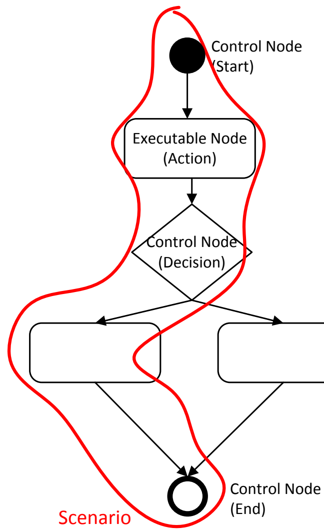

Figure 2: A Workflow and a Workflow Instance.

The Workflow Discovery and Evaluation System (WDES) is an autonomic introspective simulation system (Yilmaz and Mitchell 2009) that involves discovery of plausible workflows and exploring the use, further advancement, and practical application of the Symbiotic Adaptive Multi-simulation (SAMS) methodology (Mitchell and Yılmaz 2008) to engineer adaptive decision-making support capability for systems in shifting, ill-defined, and uncertain environments. The system is mainly consisted of three subsystems in a layered architecture, where each subsystem forms a layer for the sake of separation of concerns. The subsystems, as depicted in Figure 3, are Workflow Discovery Subsystem (WDSS), Workflow Evaluation Subsystem (WESS) and an Agent-Supported Multi Simulation (ASMS).

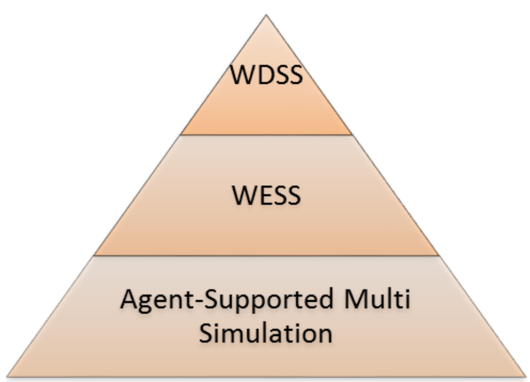

Figure 3: Layers in Workflow Discovery and Evaluation System.

Our approach combines both top-down and bottom-up approaches to realize the creative problem solving strategy introduced in (Yilmaz and Oren 2009). Using a top-down approach, a strategic plan gives the models a context, which provides a strategic direction and motivation in terms of mission objectives and a sequence of tasks, generally in the form of a scenario, in addition to the current opponent and environment situation. On the other hand, the bottom-up approach promotes the creativity by employing evolutionary techniques, where the system discovers plausible models for workflows.

WDES instantiates, executes, and evolves multiple workflows in parallel to discover a plausible workflow. The stages of the process form a closed-loop as presented in Figure 4. Prior to entering into the loop, initialization (instantiation) of the first set of models (i.e. the population) is required. WDSS is responsible for the evolution and selection stages, while WESS realizes the evaluation stage with the help of execution of workflows in an agent-supported simulation.

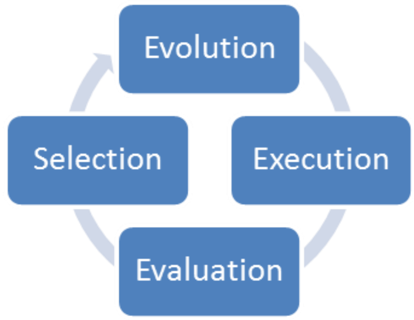

Figure 4: WDES Stages.

The reference system architecture with data flows are depicted in Figure 5. WDSS fundamentally creates competing workflows (i.e. candidate workflows) by using evolutionary algorithms, where each candidate workflow is run in an agent-supported simulation in order to assess the outcome of the candidate workflow by the help of WESS.

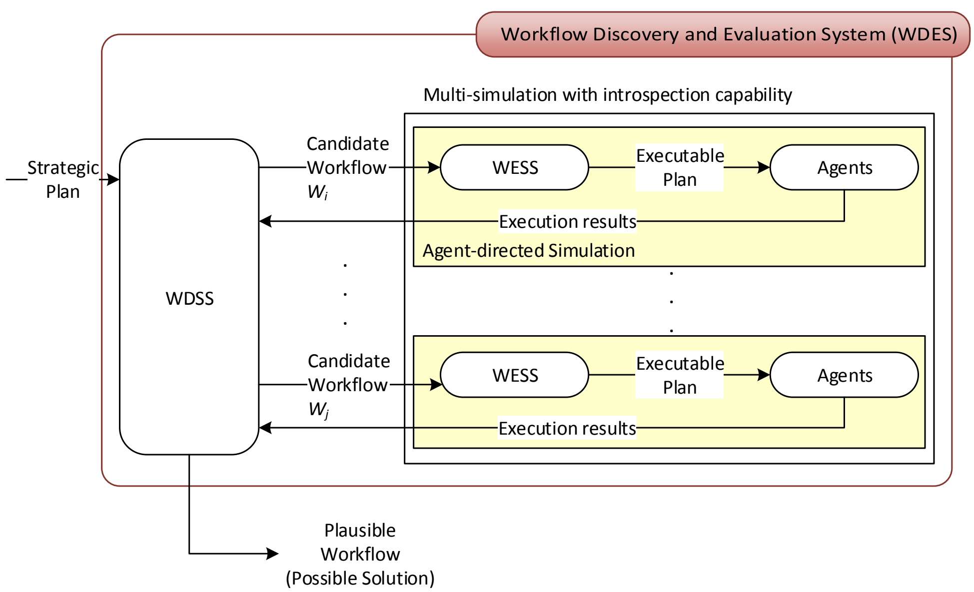

Figure 5: Workflow Discovery and Evaluation System.

WESS is an introspection system that constantly monitors the activity of the agent and the status of the environment to determine how successful the outcome of the candidate workflow is, with respect to given and selected goals. All execution results are interpreted by WDSS in order to evolve the existing workflows and to create a new set of candidate workflows. This discovery and evaluation cycle continues until a plausible workflow is found.

3 WORKFLOW DISCOVERY SUBSYSTEM

The Workflow Discovery Subsystem (WDSS) monitors the activity of agents to discern behavioral specification primitives that co-occur in simulation experiments that yield successful outcomes with respect to selected emergent goals or performance metrics. Given a strategic plan, WDSS discovers a plausible workflow by using evolutionary algorithms, specifically, the genetic programming method (Koza 1992).

Workflow discovery employs (tree-based) genetic programming (GP) to explore the possible candidate workflows. GP is an evolutionary computation (EC) algorithm, which operates on chromosomes in the form of programs or structures represented as a parse tree (Koza 1992). The main reason for using GP as the EC algorithm in this work is its ability to operate on parse trees, into which a workflow graph representation can be transformed. The strategic plan contains a sequence of tasks (workflow) in a UML activity diagram representation as well as opponent and environment information. Abstract Syntax Tree (AST) transformer translates the workflow (e.g. represented as an activity diagram) to an AST, called workflow AST, which conforms to the GP-AST metamodel. The evolution component takes the AST as input seed and generates a population of candidate workflows using the primary genetic operations (i.e. crossover and mutation). The selection component evaluates the candidates to determine the fitness value of each candidate workflow and then breeds the evolution component with the fitter candidate(s). The evaluation process continues until the desired solution (a plausible workflow) is found or termination conditions (e.g. maximum number of generations) met (Poli, Langdon, and McPhee 2008). Then the plausible workflow in AST back-transformed to an activity diagram and suggested as the possible solution.

The selection process utilizes the fitness values derived from the performance results of the simulation of each candidate workflow. The results include the system coherence (i.e. the total coherence of the tasks and goals) for each workflow under the condition of different environmental parameters and shifting and competing goals. The evaluation of the candidate workflows as well as the judgments about the tasks that contribute to successful outcomes in a workflow is viewed as a coherence problem, where each task in the workflow facilitates the achievement of another task or goal.

AST transformation can be performed automatically in the presence of a domain specific language. For example, for the C2 domain, a major domain specific language is the Coalition Battle Management Language (C-BML) (SISO 2012), which is currently under development by SISO. C-BML contains required representation elements for expressing the sequence of tasks in an operation plan/order as functionally or temporally related. Thus, the sequence of tasks can be represented in terms of an activity diagram (a workflow). Consequently, AST transformer takes the workflow represented by an activity diagram and then creates an abstract syntax tree for the GP-based evolution phase. The structure of the AST may include a primitive set, on which the genetic operations operate. Such a set may include functions such as inclusion and precedence relations that operate on the terminal set formed by the tasks in a workflow.

4 WORKFLOW EVALUATION AND AGENT-SUPPORTED SIMULATION

Each candidate workflow created by the workflow discovery subsystem is simulated by an agent. Each agent has a workflow evaluation module, where evaluation is based on the adaptive decision making capability of the agent. An agent that has an adaptive decision making capability is proposed in (Topçu 2014) and is called as deliberative coherence driven agent (DeCoAgent). DeCoAgent employs the deliberative coherence theory (Thagard and Millgram 1995) to compute the coherence, which is interpreted as a constraint satisfaction problem. In this respect, Thagard's deliberative coherence provides a mechanism to evaluate the agent's goals with the help of a connectionist network model. The connectionist model depicts a connected network of goals, tasks, and the relations among them in order to enable the calculation of the activations, link weights, and the total coherence of the model. DeCoAgent has an adaptive decision making module to (1) build a connectionist model of tasks, goals, and inhibition and facilitation relations among them, (2) calculate the total coherence of the connectionist model at every decision cycle, and (3) learn new goals, the priority of a goal, and the strengths of the relations (i.e. the connectionist model parameters) by observing its behavior and monitoring the environment.

Agents are interactive, situated, context-aware, activity-aware, adaptive, and goal-oriented. Some tasks cannot be accomplished only by one agent therefore tasks need collaboration. On the other hand, sometimes agents compete for the same tasks. Both requirements dictate that agents must be interactive, so that they can communicate in order to cooperate or compete when they execute the same task. Each agent's state consists of a list of tasks and related goals assigned to this agent as well as some other domain-specific information. For example, the state of an unmanned surface vehicle agent consists of its current posture (i.e. position and heading).

4.1 Evaluation Process

The workflow evaluation subsystem is employed to (1) transform the workflow AST to the connectionist model that includes tasks, goals, and their facilitation structure, (2) calculate the total coherence of the facilitation model, and (3) monitor the simulation execution. Simulation runs until the activation of nodes in the facilitation model becomes stable. Then, the simulation result, containing the workflow coherence, is reported back to the WDSS.

In the proposed approach, the system presents three perspectives of analysis by separating the concerns, which focus on the competing workflows, or tasks in a workflow, or a plan goal. First, by bringing forth a goal in the analysis, the system helps the user understand which tasks are essential for that objective. In many applications, goals are shifting, ill-defined, and competing. Moreover, the tasks that an agent perform depend on the context. According to the context, the priority of goals that the agent pursues may change. When a goal priority is changed, then the actions to be performed is expected to change. For example, in an exam week, a student's goal "to pass the exams" has higher priority than "to socialize with friends". Thus, the student will choose to study instead of socializing. As a consequence, the proposed system provides a technique to evaluate which tasks are chosen (making a decision for the course of action) in the presence of goals. This dictates that the agents in the system must be context-aware agents. Context-awareness can be based on the activity recognition, which is critical in an activity-driven simulation (Muzy, et al. 2013). Specifically, the evaluation of objectives helps us to analyze the following questions:

- What extent does a goal contribute to the achievement of the other goal?
- For a specific goal, which tasks are most relevant?
- What is the course of action when goal priorities are changed?
- What is the course of action when a new goal is introduced?

Second, it allows the user perform extensive analyses for each specific task in a workflow to determine its relationship both with the overall (plan) goals and with other tasks in the workflow. Thus, it helps the user understand the importance and impact of a task in the workflow. Specifically, the proposed system will help to highlight the answers for the following questions in the evaluation of tasks:

- What are the most relevant tasks in the workflow?
- Which pair of tasks most fit together?

Last, the proposed system helps to understand by evaluating of the competing plans

- If there are more than one competing/candidate workflows for the same goals, which one is to be selected?

4.2 Hypothetical Example

The following scenario is a symbolic and an illustrative example that demonstrates how to conduct the evaluation of competing workflows. The example presents two generated competing workflows for a given mission. Both workflows share the same goal system related to the mission, but they differ in tasks.

$Goals = \{g_1, g_2, g_3, g_4, g_5\}$
$wf_1 = \{t_1, t_2, t_3, t_4\}$
$wf_2 = \{t_5, t_6, t_7\}$

Each workflow is run by a DeCoAgent, which is available as open source (DeCoAgent 2014). First, the DeCoAgent will construct the related connectionist model using the task and goal relations and then will adapt its connectionist model as the simulation progress. After a specific iteration, the activations and the link weights in the connectionist model will be computed according to (Thagard and Millgram 1995) and the topology of the model will be settled. The connectionist model and the computed values for each agent is given respectively in Figure 6. Here: (i) g3 and g4 facilitates g1, (ii) g5 facilitates g2, (iii) g1 and g2 are top goals and conflict with each other. The goal relations are the same for each workflow. In the first workflow (Figure 6a): (i) t1 and t2 together facilitates g2, (ii) t3 and t4 together facilitates g4 and g5. And in the second workflow (Figure 6b): (i) t5 facilitates g4 and g5, (ii) t6 and t7 together facilitates g5.

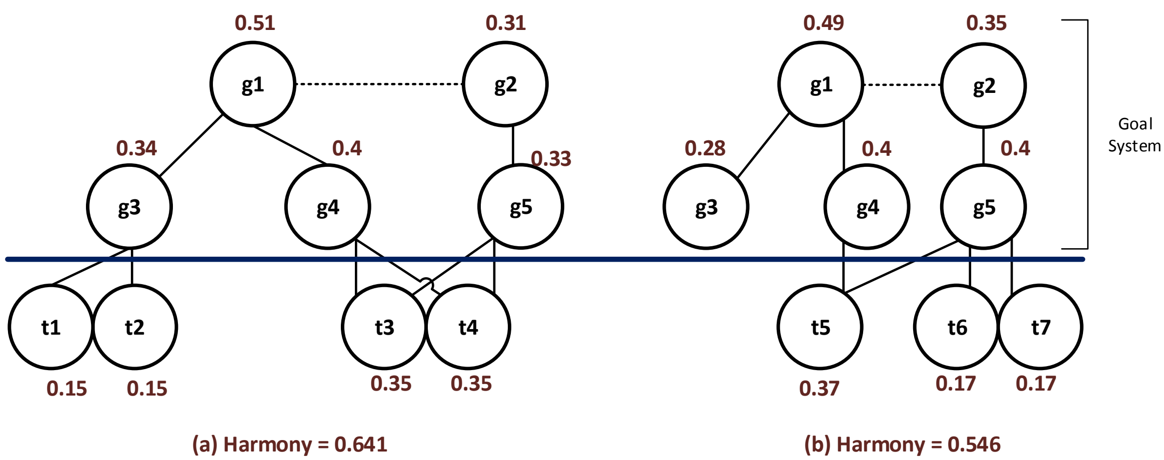

Figure 6: The Connectionist Model.

In the evaluation phase, the agent computes the harmony (i.e. the goodness of fit – the overall coherence of all tasks and goals in a workflow) of its network model and then reports it to the selection module in discovery system for the next generation of workflows. Figure 6 presents the harmony values for each connectionist model. Harmony values are computed using the formula in (Thagard 1989) and is normalized by dividing it to the number of constraints as suggested in (Thagard and Verbeurgt 1998). According to the computed values, the first workflow is more advantageous than the second one as the harmony among its goals and tasks are stronger. Here, the harmony of the tasks and goals are not the only criteria, but also some other criteria such as the successful completion of the workflow is used for the selection.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a coherence-based strategy discovery process, which is an autonomic introspective simulation system (Yilmaz and Mitchell 2009) that involves discovery of plausible workflows and exploring the use, further advancement, and practical application of the Symbiotic Adaptive Multi-simulation (SAMS) technology (Mitchell and Yılmaz 2008) to engineer adaptive decision-making support capability for systems in shifting, ill-defined, and uncertain environments.

The workflow evaluation subsystem enables us to evaluate the (competing/candidate) workflows that aim to achieve specific goals to understand the extent each task (activity) in the workflow contributes to a specific goal or subgoal and which tasks are more relevant for specific goals by analyzing a connection model. As a future work, the types and patterns of workflow designs that are robust across a large number of scenarios can be examined. The proposed strategy has potential application areas in workflow, C2, and business management disciplines. Such capability can be extended to complex domains and applications such as disaster management, where the discovery of tactics and strategies under uncertainty and evolving environmental conditions will increasingly become critical.

ACKNOWLEDGMENTS AND DISCLAIMER

This work is partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) as part of 2219 Research Fellowship Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any affiliated organization or government.

REFERENCES

Bentley, P., and D. W. Corne. 2002. *Creative Evolutionary Systems.* Elsevier.

Bjornberg, K. E. 2009. "What Relations Can Hold Among Goals, And Why Does It Matter?" *CRITICA, Revista Hispanoamericana de Filosofia* (Instituto de Investigaciones Filosóficas of the National Autonomous University of Mexico) 41 (121): 47-66.

Davis, P.K. 2000. "Exploratory Analysis Enabled by Multiresolution, Multiperspective Modeling." In *Proceedings of the 2000 Winter Simulation Conference.* Edited by J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, 127-134. Orlando, FL: ACM.

DeCoAgent. 2014. "DeCoAgent Library." Last modified July 06, 2014. http://sourceforge.net/projects/decoagentlibrary/.

Finke, R. A., T. B. Ward, and S. M. Smith. 1996. *Creative Cognition: Theory, Research, and Applications.* A Bradford Book.

Georgakopoulos, D., M. Hornick, and A. Sheth. 1995. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure." *Distributed and Parallel Databases* (Kluwer Academic Publishers). 3: 119-153.

Jablonski, S., and C. Bussler. 1996. *Workflow Management: Modeling Concepts, Architecture, and Implementation.* International Thomson Computer Press.

Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers By Means of Natural Selection.* MIT Press.

Leymann, F., and W. Altenhuber. 1994. "Managing Business Processes as an Information Resource." *IBM Systems Journal* 33: 326-348.

Macal, C. M., and M. J. North. 2010. "Tutorial on Agent-based Modelling and Simulation." *Journal of Simulation* (Palgrave Macmillan) 4: 151-162. doi:10.1057/jos.2010.3.

Mitchell, B., and L. Yılmaz. 2008. "Symbiotic Adaptive Multisimulation: An Autonomic Simulation Framework for Real-Time Decision Support under Uncertainty." *ACM Transactions on Modeling and Computer Simulation* 19 (1): 2:1-2:31.

Muzy, A., F. Varenne, B. P. Zeigler, J. Caux, P. Coquillard, L. Touraille, D. Prunetti, C. Philippe, O. Michel, and D. RC Hill. 2013. "Refounding Activity Concept? Towards a Federative Paradigm for Modeling and Simulation." *Simulation Transactions of the Society for Modeling and Simulation International* (SAGE) 89 (2): 156-177. doi:10.1177/0037549712457852.

Narendra, N. C. 2003. "AdaptAgent: Integrated Architecture for Adaptive Workflow and Agents." *South African Computer Journal* 31:9-17.

Narendra, N. C. 2004. "Flexible Support and Management of Adaptive Workflow Processes." *Information Systems Frontiers* (Springer) 6 (3): 247-262. doi:10.1023/B:ISFI.0000037879.05648.ca.

OMG. 2011. *Business Process Model and Notation (BPMN).* Object Management Group.

OMG. 2013. *OMG Unified Modeling Language (OMG UML) Version 2.5 - Beta 2.* Object Management Group.

Poli, R., W. B. Langdon, and N. F. McPhee. 2008. *A Field Guide to Genetic Programming.* Published via http://lulu.com. Accessed May 09, 2013. http://www.gp-field-guide.org.uk/.

SISO. 2012. *Guide for: Coalition Battle Management Language (C-BML) Phase 1.* Simulation Interoperability Standards Organization (SISO).

Thagard, P. 1989. "Explanatory Coherence." *Behavioral and Brain Sciences* 12 (3): 435-502.

Thagard, P., and E. Millgram. 1995. "Inference to the Best Plan: A Coherence Theory of Decision." In *Goal-driven Learning*, edited by Ashwin Ram and David B. Leake, 439-454. Cambridge, MA: MIT Press.

Thagard, P., and K. Verbeurgt. 1998. "Coherence as Constraint Satisfaction." *Cognitive Science* 22 (1): 1-24.

Topçu, O. July 2014. "Adaptive Decision Making in Agent-based Simulation." *Journal of Simulation: Transactions of the Society for Modeling and Simulation International* (SAGE) 90 (7): 815-832. doi:10.1177/0037549714536930.

Yilmaz, L., and B. Mitchell. 2009. "Autonomic Introspective Simulation Systems." In *Agent-directed Simulation and Systems Engineering*, by L. Yilmaz and T. Oren, 37-72. Weinheim: Wiley-VCH Verlag GmbH & co. KGaA.

Yilmaz, L., and T. Oren. 2009. "Rethinking M&S to Enhance Creativity and Computational Discovery." In *Proceeding of the 2009 Summer Computer Simulation Conference.* 66-74. Istanbul, Turkey.

AUTHOR BIOGRAPHIES

OKAN TOPÇU is an Assistant Professor in the department of Computer Engineering with a joint appointment to Naval Sciences and Engineering Institute at Naval Academy, Istanbul, Turkey. Professor Topçu received his Ph.D. and M.S. degrees from the Middle East Technical University, Ankara, Turkey in 2007 and 1999 respectively. His current research interests include distributed simulation, agent-based simulation, model driven engineering, software architectures, and creative evolutionary systems. He is currently serving as department head of computer engineering. His email address is okantopcu@gmail.com.

LEVENT YILMAZ is Professor of Computer Science and Software Engineering with a joint appointment at the department of Industrial and Systems Engineering at Auburn University. He holds M.S. and Ph.D. degrees in Computer Science from Virginia Tech. He is the founding organizer and General Chair of the Annual Agent-Directed Simulation Symposium series and is currently serving as the Editor-in-Chief of the Simulation: Transactions of the SCS. His email address is yilmaz@auburn.edu.