Cognitive Agents for Adaptive Training in Cyber Operations

Randolph M. Jones[✉], Ryan O'Grady, Fernando Maymi, and Alex Nickels

Soar Technology, 3600 Green Court, Suite 600, Ann Arbor, MI 48105, USA
{rjones,alex.nickels}@soartech.com, ryanjo@gmail.com,
fernando@maymi.net

Abstract. To support training for offensive and defensive cyber operations, we focus on giving the trainee a realistic ecosystem to train in. This ecosystem includes models of attackers, defenders, and users.

The high-level goals for adaptation in this ecosystem are of two types: realism in behavior and tailoring of training. In terms of realism, real-world cyber operations are highly adaptive. Attackers constantly innovate new attack techniques and adapt existing techniques to take advantage of emerging vulnerabilities. Defenders must adapt to ever-changing attack tactics and vulnerabilities. Users continuously adapt to rapidly changing technology. A realistic training ecosystem requires those adaptations to be reflected in the models of the synthetic actors. In terms of tailoring, training systems often require ecosystem actors to step outside of what would "realistically" happen and instead create artifices to focus the trainee's experience on particular learning objectives.

In support of these high-level adaptation goals, the CyCog (CYber COGnitive) framework currently supports three types of adaptivity. These include adaptation of tactics and techniques (for example, innovating a new attack or defense), adaptation of level of sophistication (for example, to make an attacker more or less aggressive, or to limit or expand a defender's awareness to focus training), and adaptation of personality parameters (for example, to tune the preferences of various types of users in the ecosystem).

To maintain maximum training flexibility, we use a mixed-autonomy approach that allows all forms of adaptation to be controlled on a spectrum from automated tuning to manual manipulation by human instructors.

Keywords: Cyber operations · Training · Cognitive modeling · Adaptive behavior · Adaptive instruction

1 Introduction

We have developed the CYber COGnitve (CyCog) framework to, among other reasons, add realistic decision-making models to the "ecosystem" used in cyber-operations training environments. The core of CyCog is a set of cognitive models that can play the roles of cyber attackers, defenders, or users, to various levels of fidelity. In addition to the cognitive models themselves, CyCog incorporates a generic framework for simulating applications and networks, as well as for integrating with real applications and

R. A. Sottilare and J. Schwarz (Eds.): HCII 2019, LNCS 11597, pp. 505–520, 2019.
https://doi.org/10.1007/978-3-030-22341-0_40

networks. Specialized components of CyCog focus on how best to represent the common knowledge and situation understanding shared by cyber operators.

One of the goals of the CyCog project is to provide training systems with a rich ecosystem of realistic actors, so defensive operators can train against threats as they appear “in the wild”, rather than simply against individual tactics and techniques in a classroom or lab setting. The use of cognitive models provides two primary enhancements to the training experience. First, our cognitive models have goals, and the tactics, techniques, and procedures (TTPs) they use are consistent with those goals. This provides opportunities for diagnosis, attribution, prediction, and preemption that would otherwise be absent from a training experience. Goals can be long term, allowing CyCog to model Advanced Persistent Threats (APTs), as well as less persistent types of threats, such as “script kiddies”. Second, our cognitive models include knowledge, situation-understanding capabilities, and learning mechanisms that allow them to adapt in their responses to defensive measures. Adaptive attackers provide the opportunity for a much richer and more realistic experience in training than current textbook and classroom activities provide, and they provide a more cost-effective training option than exercises that employ human experts as role players. We have also integrated the CyCog agents with a form of *Dynamic Tailoring*, which adapts agent behaviors to support training goals. This paper describes the CyCog framework and its application to training, with a focus on forms of adaptation that we are incorporating into the cognitive models and training systems.

2 Cyber Operations Training

Cyber operations present a persistent and evolving threat to military and civilian information systems. Both the Department of Defense [24] and the Office of the Director of National Intelligence [25] have ranked cyber warfare as our top national security concern. Department of Homeland Security Secretary Nielsen has stated that “cyber threats collectively now exceed the danger of physical attacks” [22]. In addition to threats to our military forces, cyber attacks pose domestic infrastructure and economic threats [27]. Attackers continually evolve their tactics, techniques, and procedures (TTPs) to exploit emerging vulnerabilities so they can exfiltrate, manipulate, or deny information. Would-be cyber attackers are constantly changing their attack vectors to take advantage of security lapses by human resources and the latest vulnerabilities in information technology. These activities are guided by cognitive behavior that includes a variety of types of goals and expertise: script kiddies, ideological activists, investigators, financial criminals, intelligence agents, or cyber warfighters [15]. At the human, cognitive level, offense reacts and adapts to actions of defenders [26] and users [1] that are also cognitively driven. To counter these adversarial actions, cyber-security personnel must rapidly adapt to develop and refine their defensive skills.

A common way to support this adaptation by defensive personnel is through training exercises within realistic environments. To be effective, these events require intelligent, adaptive opposition forces (OPFOR), which currently requires the use of human role players. Unfortunately, using human adversaries is not feasible to support the scale and frequency of exercises needed to maintain a highly skilled defense, because these skilled OPFOR are scarce and expensive resources. In addition, a maximally effective training paradigm adjusts the training experience in response to the level of performance of the trainee. Using intelligent systems to provide role players introduces consistency and cost-effective automation across the training experiences.

Building effective training systems for cyber operations presents a suite of unique problems:

- Offensive and defensive activity is highly interactive and dynamic.
- Cyberspace environments are defined by both their structure and the activity of all actors operating within them. To be realistic, training environments must provide an *ecosystem* of users (whose behavior can is variable and unpredictable), as well as attackers and defenders.
- User behavior (willingly or not) can either assist or hinder the efforts of both attackers and defenders. For example, vigilant users can provide valuable intelligence, while careless users often create vulnerabilities that can be exploited.
- Cyber attackers and defenders themselves are extremely adaptive and creative. In order to meet their objectives, they will change tactics or tools based on opportunities detected in a computer network or responses initiated by adversaries or users.

These problems underscore the fact that cyber operations are a domain in which rapid adaptivity is the coin of the realm. Effective training systems and role players must exemplify this variability in their basic structure and their delivery of training experiences. We have developed a set of cognitive models for cyber-operations training, together with a general framework in which they operate. The CyCog framework provides the shared infrastructure for building knowledge-based agents that can play various roles within the cyber-operations ecosystem. Individual attacker, defender, and user models instantiate CyCog with specific knowledge bases, modes of operation, and types of adaptivity. This paper describes three particular forms of adaptivity that we have so far emphasized in developing CyCog models that support cyber-operations training.

3 Intelligent Systems for Cyber Training

Our team began development of a cyber attacker model in 2013. Subsequently, we pursued projects that demanded the development of additional models of cyber defenders and network users. Combining the models and performing several major redesigns and refactors eventually produced the generalized CyCog framework. CyCog-A (the attacker instantiation of CyCog) and CyCog-D (the defender instantiation) are implemented in the Soar cognitive architecture [17], with associated tools implemented in Java for a high degree of portability. CyCog agents emulate human role players by modeling their decision-making processes using cognitive-systems design patterns we have developed over the course of more than two decades of applied research and development (see, for example, [8, 9, 11, 29, 30, 32]). This allows CyCog agents to provide not just static representations of cyber operations, but generative behavioral models that execute and interact with a network in real time.

One advantage of using the Soar architecture over some other approaches to cognitive modeling [14] is that it incorporates decades of experience in cognitive research into an architecture that is reusable across new cognitive models. Soar's models integrate varieties of knowledge, learning, and reasoning strategies, including semantic, episodic, and procedural memory, reinforcement learning, spatial reasoning, and activation dynamics, as well as cognitive models that depend on situation understanding and interaction with complex environments [18]. Soar provides a reusable software architecture that instantiates a unified theory of cognition, which gives CyCog firm practical and theoretical grounding.

In the CyCog-A model (Fig. 1), knowledge is structured as composable hierarchical modules representing TTPs that can combine in multiple ways to support different situations. Combining goals and subgoals for a particular attack generates a flexible, situation-dependent attack tree structure. As the agent attempts to achieve the goals of its mission, it communicates its intent to an abstraction layer, which is responsible for translating intent into real-world actions. This enables the agent to reason over task-level goals, such as "list open ports on 192.168.1.5", without requiring the agent to possess system-level knowledge of how to do so. To accomplish this, the abstraction layer determines which of its available resources are appropriate and translates the agent's intent into commands for those resources. The abstraction layer supports interaction between CyCog-A and custom-built or commercial off-the-shelf (COTS) tools to perform specific offensive actions, such as port scanning, password cracking, or phishing. The modular nature of the toolkit facilitates the addition and removal of tools as needed, and also provides a robust framework for configurability and portability. Finally, a human supervisor can command and control the agent via a CyCog Command and Control (C4) Server, which uses an abstract communication model that currently implements HTTP and IRC support.

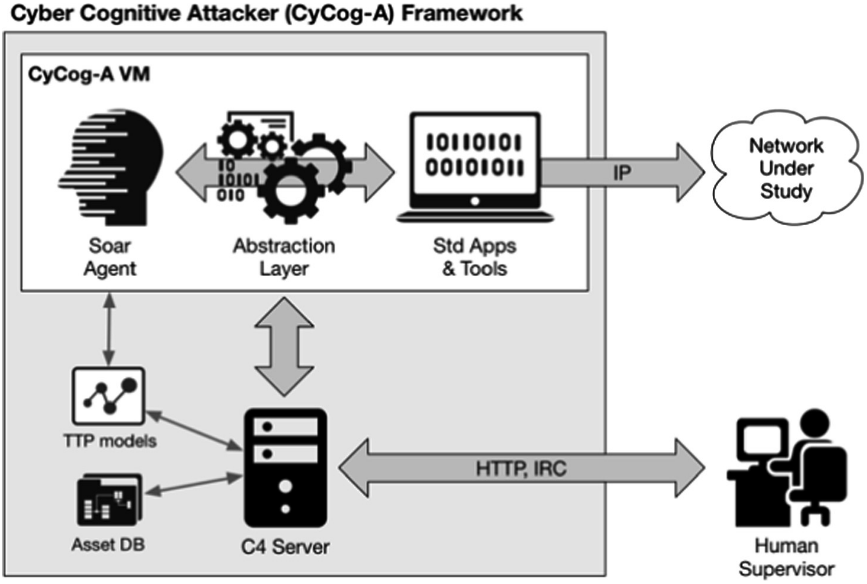

Fig. 1. CyCog-A framework

CyCog agents use a key combination of *situational understanding* [14] and *least-commitment reasoning* [33] to generate human-realistic, situation-responsive behaviors. Once a scenario has been configured, CyCog agents use situation-understanding knowledge to engage with the network and begin building and maintaining a model of the environment. This model drives the agents in the pursuit of their scenario goals and the employment of tactics to achieve those goals. This, in turn, allows them to adapt robustly in response to changes in the environment, including actions and responses by other role players in the cyber ecosystem. Least-commitment reasoning (LCR) is a process for "generating partially ordered, partially specified sequences of actions whose execution will achieve an agent's goal" [33]. CyCog also implements the situation-understanding model in an external and sharable knowledge representation, which allows multiple CyCog agents to work together. This shared knowledge representation is stored in an *Asset Database*, which is currently implemented in a software database we call the Cyber Data Repository (CyDaR). This datastore enables storing, correlating, and retrieving data from all three layers of cyberspace: physical, logical and persona [6]. As other actors and role players take actions that change elements of the ecosystem, CyCog agents may update their shared situation-understanding model and revise their goals and tactics in response to the changes.

4 Adaptivity in Cyber Training

As described above, to support training for offensive and defensive cyber operations, we focus on giving the trainee a realistic ecosystem in which to train. This ecosystem includes models of attackers, defenders, and users. We have also stressed the adaptive nature of cyber operations in general, as well as the need for adaptivity in effective training.

The high-level goals for adaptation in the training ecosystem include *realism in behavior* and *tailoring of training*. In terms of realism, real-world cyber operations are highly adaptive. Attackers constantly innovate new attack techniques, while defenders must adapt to ever-changing attacker TTPs, and users continuously adapt to rapidly changing technology. A realistic training ecosystem requires those adaptations to be reflected in the models of all of the synthetic actors. Simultaneously, we must balance realism and trainee proficiency in order to challenge learners without overwhelming them. Thus, any effective training environment must be tailorable in order to adapt to an individual's (or team's) proficiencies, as well as to other pedagogical goals (such as targeted lesson plans).

In support of these high-level adaptation goals, the CyCog framework currently supports three types of adaptation. These include adaptation of tactics and techniques (for example, innovating a new attack or defense), adaptation of level of sophistication (for example, to make an attacker more or less aggressive, or to limit or expand a defender's awareness to focus training), and adaptation of personality parameters (for example, to tune the preferences of various types of users in the ecosystem).

To maintain maximum training flexibility, we use a mixed-autonomy approach that allows all forms of adaptation to be controlled on a spectrum from automated tuning to manual manipulation by human instructors

4.1 Adaptivity of Tactics and Techniques

One of the key characteristics of cyber attackers (as well as effective defenders) is that they are adaptive: they perceive and react to changes in their environment and they learn to exploit the tendencies of their adversaries. While there are varying degrees of adaptivity, any viable autonomous agent must execute a four-phased loop: (1) sense the environment, (2) learn what is different and/or interesting, (3) decide how to best achieve its next set of goals, and (4) act on the environment in pursuit of those goals [20].

CyCog attacker models incorporate TTP-level adaptivity by combining a robustly updated representation of situational understanding (stored in the CyDaR database) with least-commitment reasoning over a modular, generative set of goals and subgoals that fluidly implement a variety of TTPs for a spectrum of cyber-operational goals. CyDaR represents situational understanding by integrating an ontology of cyber-operational concepts with an application programming interface (API) for database queries and updates. When multiple agents are coordinating, they can use the shared database to drive adaptivity in their situation understanding and employment of tactics. For example, one agent may remember on which hosts it implanted remote access toolkits (RAT), while another agent notices when they stop responding to commands, reasoning that the defender may have detected and contained the compromise.

Situation understanding is only the first half of the story for the adaptivity of tactics and techniques. Least-commitment reasoning (LCR) is an approach to intelligent systems that allows the system to "satisfice", or "make the best decision in a reasonable amount of time". Least-commitment strategies differ from more traditional AI planning and rule-based systems along a number of dimensions:

- Traditional AI relies on "weak methods" (logical formalism with little knowledge), while LCR relies on significant amounts of domain-specific knowledge.
- Traditional AI downplays the role of dynamic situational understanding, while dynamic consideration of the environment is a key part of LCR.
- Traditional AI relies on computationally expensive search techniques to generate optimal plans and to precompute contingencies, while LCR trades off optimality for "reasonability" and reconsiders contingencies in rapid fashion at run time.
- Traditional AI expends significant time and effort identifying a "best course of action", making it very expensive to change course, while LCR continuously makes inexpensive decisions about whether to adjust or switch the current course of action.
- Traditional AI has difficulty anticipating actions that cannot be predicted ahead of time, while LCR reduces the need for anticipatory planning.

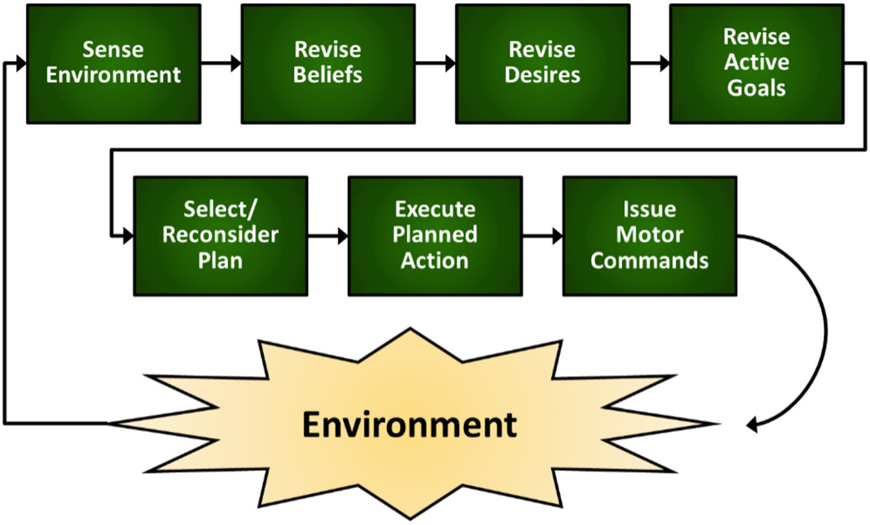

Fig. 2. Abstract representation of the least-commitment decision cycle for a system that integrates situation understanding, reasoning, planning, and action.

An LCR system implements intelligent decision making by engaging in a continuously running "decision loop". This loop must run many times per second, possibly hundreds or thousands, depending on the application, in order to deliver responsive behavior. Traditionally, when mapped onto human cognition, it is assumed that this cycle should run about 20 times per second [21]. In each 50 ms time slice, the LCR system must make incremental changes to its internal state, based on the most recent sensed information from the environment. The LCR's internal state consists of several functional elements including situational understanding (represented by beliefs), goal management (consisting of desires, which are things the agent would like to be true, and goals, which are things the agent has decided to commit resources to achieving), and action management (consisting of plans, which are short-term courses of action, and individuals actions that are consistent with the current plans). Figure 2 provides an abstract illustration of this cycle [12].

Because each individual decision is rapid and incremental, the LCR decision cycle composes individual decisions into a course of goal-driven actions that can adapt in small or large ways, depending on changes to the situation. The key to making rapid, but good quality, decisions is to replace the resource-intensive search process of traditional AI with an efficient, knowledge-intensive pattern-matching process. Cognitive systems that support LCR provide fast methods for accessing large knowledge bases and bringing the knowledge to bear on the decision-making process in rapid fashion [13].

The Soar Cognitive Architecture

The Soar cognitive architecture provides the implementation substrate for CyCog Attacker and Defender agents, as it is one such mature engine for developing LCR cognitive systems. Soar includes an extremely efficient pattern-matching engine for accessing knowledge, and it has been demonstrated to execute the LCR decision cycle orders of magnitude faster than human-scale decision times [7, 16, 19]. Every software architecture that supports LCR implements some variation of the LCR decision cycle [12]. Figure 3 illustrates Soar's implementation of this cycle.

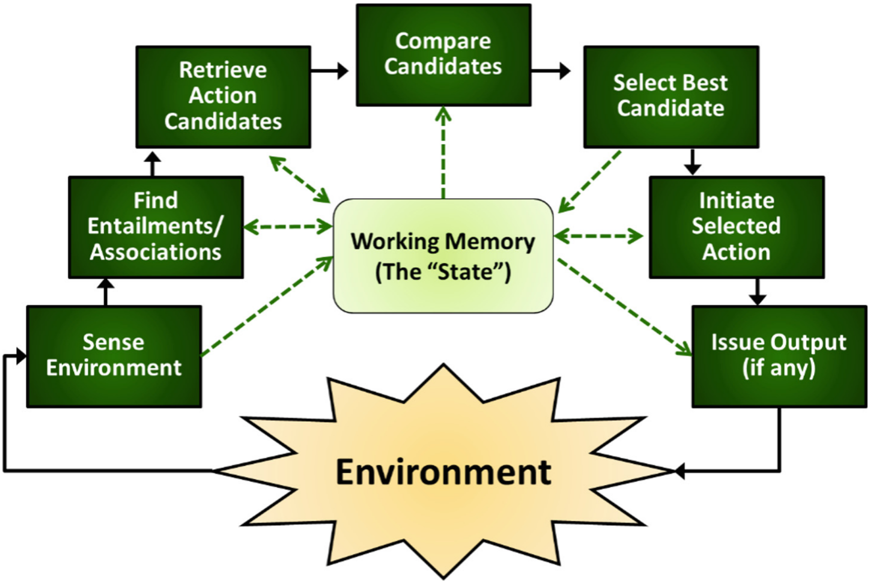

Fig. 3. Instantiation of the least-commitment decision cycle in Soar.

Soar implements a state-based approach to LCR, but this is not to be confused with the way the term "state" is used for finite-state machines (FSMs). In FSMs, each state must contain a small number of situational variables, and the number of states in an FSM explodes combinatorially with the complexity of the problem domain. A Soar "state" instead contains some arbitrary number of features (possibly thousands) that describe the system's current situational awareness and current goals. These features serve as input to the pattern-matching engine, which retrieves knowledge to create new beliefs, create new goals, or suggest new actions to take in pursuit of existing goals. During each instance of the decision cycle, a Soar agent retrieves knowledge to compute inferences (entailments and associations) from the current state, to retrieve candidates for the next "action decision" (which may be an action to create a new belief, to create a new goal, or to send output to the environment), to select a single next action, and then to execute that action.

In addition to providing efficient native support for LCR, Soar contains a number of subsystems to support psychological relevance of implemented cognitive systems. Primary among these are learning and memory subsystems that support reinforcement learning, semantic memory and learning, and episodic memory and learning. For CyCog, episodic memory can assist in discovering unknown side effects of individual decisions. Semantic memory can help in generalizing side effects into abstract patterns. Reinforcement learning can assist in making probabilistic evaluations of likely outcomes of competing combinations of offensive or defensive decisions. Each of these three mechanisms support some form of adaptivity or learning. Some have already been exploited within CyCog, and others will lead to extended forms of adaptivity and learning in the future.

Cognitive Adaptivity and Learning

Our initial CyCog agents incorporate fairly simple logic for choosing offensive and defensive actions, and include a basic understanding of how to compromise or defend computer systems. The initial agents provide a conceptual structure in which to organize this knowledge, based on a cognitive analysis of deception, sensemaking, and relevant features of uncertainty and complexity. The initial agents also implement an extensible, first-principles knowledge representation. We are building from this base representation and additional analyses to implement agent extensions that naturally synthesize complex offensive and defensive strategies.

Our work on the generalized CyCog framework has also focused in part on building a relatively thorough (albeit scaled-down) body of cyber-relevant knowledge, and developing a grammatical model of cyber operations to facilitate the continual updating of agent knowledge with new goals, subgoals, and TTPs. Our companion analysis of cyber operations and associated workflows suggests that we can capture much of the structure of cyber strategy as a formal model. The key focus is to develop an ontological model that relates abstractions about tactical approaches and constraints that govern how they may be combined, to types and permissible sequences of specific tactics. One key to this approach is the modularization of knowledge units that can then be recombined at run-time to generate novel attacks from first principles, reasoning as appropriate about possible moves at multiple levels of abstraction.

This approach also forms the foundation for long-term tactical adaptivity through learning mechanisms that minimize the need for model engineers to update the agents manually through time. Although most work on CyCog to date has emphasized expert models (as opposed to learning models), we have also performed initial investigations into four types of learning within the cyber-operations domain:

- *Inferential learning* is the result of Soar's goal-driven reasoning process, and occurs when this process derives a new high-level insight from previously known low-level steps. In the Soar architecture, this capability is called "chunking." It is analogous to the learning that a trainee experiences on working through a proof: the trainee "knows" all the pieces, but inferential learning leads her to recognize their implications. In our initial agents, inferential learning takes place when the agents dynamically compile multiple steps (possibly modules taken from multiple TTPs) to generate a single, coherent attack.
- *Observational learning* results from agent interaction with the external world. For example, our attacker agent can deploy a distributed denial of service attack for the goal of disrupting a service *source*, and it may learn from experience that the same action can also accomplish the goal of disrupting a service *channel* when the defensive response includes blocking traffic from the attack addresses.

Architecturally, Soar supports this kind of reasoning through a combination of its semantic and episodic memories.

- *Abductive learning* is reasoning from observations to the best explanation. This mechanism may allow the agents to infer causal explanations to observed adversary responses, and then use these causal models to innovate new attacks or defenses. The inference of causal models is an essential example of adaptivity among human cyber operators, and it serves as an essential part of the "dance" between attackers and defenders.
- The development of a formal knowledge representation also opens the door to *instructional learning*, in which a human supervisor can coach individual instances of the agents. We are investigating this form of learning in the context of prior work on taskability and interactive task learning [4, 31].

In all four types of learning, as an agent instance learns, it can share its knowledge with other agent instances, either on the same problem or across multiple scenarios, so that the agent knowledge bases increase in capability over time with minimal need for support by model builders.

4.2 Adaptivity of Synthetic Role-Player Sophistication

As we have suggested, the most effective training environment would serve an entire cyber ecosystem containing a spectrum of actors and role players. For training purposes, a key dimension along which to vary agent behavior is in the level of sophistication of the agents populating the ecosystem. Trainees should receive training experiences containing a variety of levels of competence in terms of attackers, network users, and possibly also collaborative defenders. In addition to simply providing a rich ecosystem, levels of sophistication can be dynamically adapted to suit the level of competence of the trainee or to focus lessons on particular pedagogical goals.

Training dynamics involves two issues:

1. Estimating the proficiency of the trainee
2. Managing the flow of training.

Conventional methods for estimating proficiency focus on explicit testing, which interrupts the flow of the training experience. Conventional methods for managing the flow of training are built around pre-defined lesson modules, which do not readily adapt as the trainee learns. Our colleagues have pursued two capabilities developed in other projects [5], which we have adapted for training in cyber operations: assessing proficiency by observing the trainee's actions without explicit tests, and dynamic tailoring of the exercise as it evolves. The Dynamic Tailoring System is another type of agent that interacts with the ecosystem agents (attackers, defenders, and users) to modulate their behavior (e.g., the range of attack tools available to the attacker; the complexity of the attack, the sophistication of the attack goals or techniques) as a function of the trainee's observed proficiency.

During our development of prototype training systems for cyber operations, we focused particularly on the adaptivity facilitated by Dynamic Tailoring (DT). Dynamic Tailoring is the process of adapting training content and gameplay dynamically based on the observed needs of the individual trainees in the training environment. For example, in a social training simulation the system may automatically demonstrate content to a novice trainee who has demonstrated difficulty.

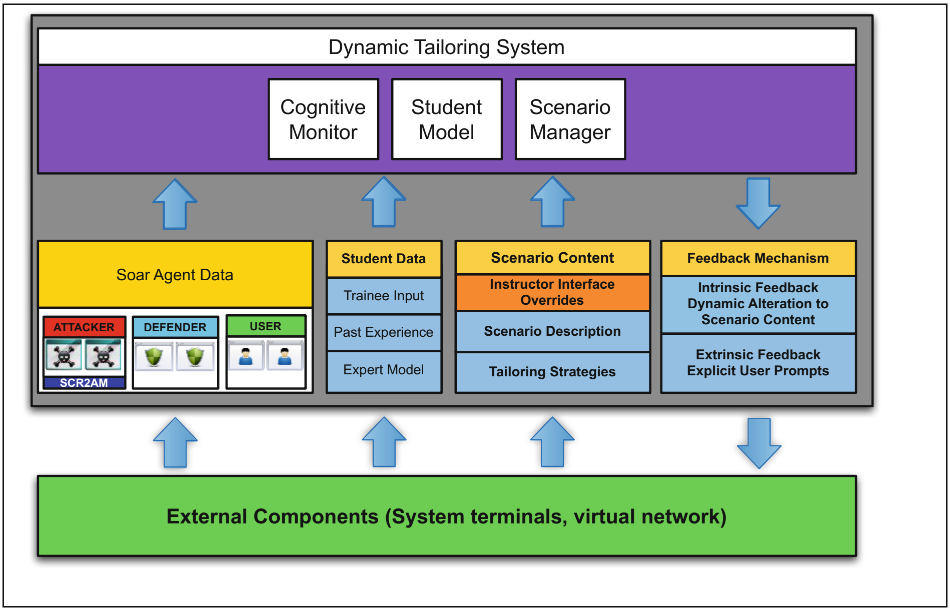

Fig. 4. Dynamic tailoring components and system input and outputs.

Research shows that immediate feedback on errors is a more effective training technique than delayed feedback [28]. Many training and simulation-based environments provide feedback at the conclusion of an exercise – reducing the potential learning effect. Dynamic Tailoring is capable of providing timely feedback, which can lead to a more effective training experience.

The Dynamic Tailoring component is responsible for capturing the state of the trainee, including the trainee’s assessed proficiencies. The DT accomplishes this by building a memory model of the trainee’s most recent actions, assessment of those actions, training strategy, and scenario goals. Based on this information the DT is able to assess trainee performance in real time. As illustrated in Fig. 4, the Dynamic Tailoring component monitors multiple inputs and outputs of the system. These external inputs include information on individual CyCog agents representing the attackers, defenders, and users. The DT monitors and modifies the behavior of these agents as required to fulfill the overall scenario objectives. The system collects trainee data encompassing current and past actions, as well as an expert/novice classification. This data is used to tailor the experience based on trainee skill and performance. Scenario Data includes a scenario description, training strategies, and inputs for the external instructor interface. This information defines the scenario and the approach the

DT uses to complete it. Finally, a feedback mechanism provides the DT with the ability to influence external components. These outputs are used to change the simulation or provide prompting to the trainee.

Internally, the Dynamic Tailoring component has three core components for managing the exercise and monitoring the trainee. A Scenario manager monitors the overall scenario status, controlling the declaration of success and failure. A Cognitive monitor observes the cognitive status of the trainee and CyCog agents working within the ecosystem. The third component is a trainee model that houses information on individual trainee performance. This component evaluates individual trainee actions and assesses their effectiveness. Based on these evaluations, DT can adjust the difficulty level of a scenario. Such adjustments can include altering the number of cyber attackers, changing the network topology, spawning/closing system vulnerabilities, and configuring the level of sophistication of attacks and counter attacks.

The primary method for adapting attack sophistication is to adjust the complexity, aggressiveness, or other features of the goals and TTPs that are assigned to CyCog attacker agents. To foster this type of adaptivity, we have pursued a formal mapping of TTPs (informal descriptions of tactics used largely by military personnel) to the formal knowledge representation language we invented for CyCog and other Soar-based intelligent agents.

The term TTP is pervasive in the cybersecurity literature. Despite this ubiquity, there are no clear definitions allowing the community to differentiate tactics, techniques, and procedures. While ambiguity and imprecision when referring to TTPs is usually not problematic among security professionals, it is a significant impediment to using these concepts in autonomous systems. This problem manifested itself while developing actionable behavioral models of offensive cyberspace operations in the CyCog attacker agent. Our goal has been to ground TTPs in a semantic representation that enables adversarial behavior modeling and autonomous decision-making, reasoning, and learning. This representation will also allow translation of varieties of (informal) TTPs into formal attacker goals and methods, at varying levels of complexity, which DT can then adjust to foster effective training.

4.3 Adaptivity of Synthetic Role-Player Personality

Even the most adaptive and capable automated agents are of limited use if training exercise planners cannot configure and task them according to the exercise objectives. However, it is not reasonable to expect cyber training experts to also be engineering experts. Thus, it is necessary to develop an engine to translate training adaptations into ecosystem and agent parameter settings. CyCog agents are driven in part by their knowledge base of goals and TTPs, and in part by a fairly large number of "personality" parameters that govern decision making. These parameters can include preferences for which types of attacks to use, which attacker tools to use, how stealthy or deceptive to be, and many others. The result is a large and complex amount of formal information that may need to be specified to configure each agent in the ecosystem. Instructors must be able to avoid such fine-grained configuration, instead expressing high-level configuration preferences that can be automatically refined into individual parameter settings. To support these types of training adaptations, we are exploring AI techniques and tools for collaborative planning and decision support to reduce the complexity of configuring and tasking the CyCog agents (see examples in Fig. 5).

Cyber operations are complex, and any approach to adapting the personalities of agents in the ecosystem cannot hope to ignore that complexity. It is important that we create tools that assist exercise planners in configuring, deploying, and tasking the agents, without hiding or obscuring access to key functionality. However, we recognize that instructors do run large-scale, complex training scenarios without having to specify every minute interaction and configuration option. Our approach is to analyze the existing workflow of instructors running exercises with human role players, and adapting that workflow to the synthetic CyCog agents.

We are using cognitive task and workflow analysis techniques to model how exercise planners currently brief human role players and relay *mission objectives*, *tasks*, and *constraints*. Based on the model we are developing for planner/role-player interaction, we plan in the near future to implement a proof of concept approach to configuring CyCog agents. Matching the interaction between instructors and human role players, we are adopting a mixed-autonomy approach. For portions of the training exercise that the instructor wishes to "micromanage", they will be allowed to specify configuration options to any desired level of detail. For the more usual case, where role players must appropriately interpret the "instructor's intent", the CyCog agents will similarly make appropriate inferences about reasonable configuration options that meet the requirement of the instructor's high-level intentions. In general, we envision scenario configuration as a collaborative, iterative process in which the exercise planner specifies tasks and constraints, and the support system recommends changes based on conflicts, errors, and omissions. However, there are other interaction models that may prove to be more appropriate, such as the mission planner specifying generalized requirements and allowing the agents to derive specific parameters based on those general guidelines.

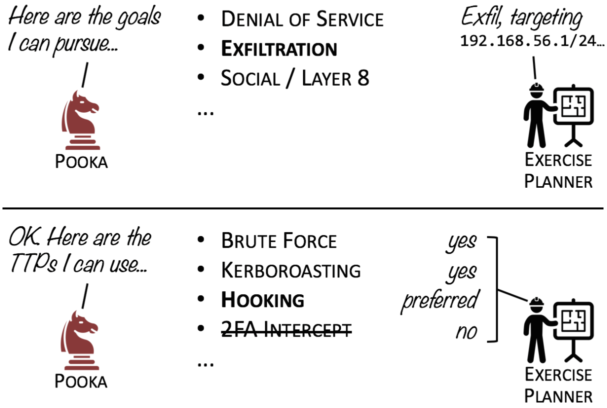

Fig. 5. Guided configuration of CyCog agents

In additional to providing tools to map instructor's intent to individual parameter values, we must thoroughly and formally define the configuration parameters of the agents in the first place. Throughout our design of CyCog, we have seized every opportunity for data-driven configurability through the application of two techniques: *externalization* [23] and *parameterization* [2]. Externalization is the process of making explicit those aspects of the knowledge base (such as data structures and parameters) that can be usefully shared outside the core system. CyCog already externalizes a number of parameters that allow data-driven specification of goals and decision-making preferences, and CyDaR is another example of such an externalization. By externalizing information, it becomes much easier to create tools that can support configuration without having to make changes to the underlying software. This in turn fosters mechanisms from improving the adaptivity of training.

Parameterization is the process of moving from special-purpose to general-purpose implementations that can be configured to behave differently by supplying different parameters. It is the nature of complex intelligent systems that there are many different types of knowledge that cannot easily be generalized into parameter-driven patterns. However, there are also large portions of knowledge than *can* be generalized and parameterized, and we aggressively pursue every opportunity to do so when engineering the CyCog agent knowledge bases. It tuns out that such parameterization efforts serve three important goals: they make knowledge bases easier to extend, they make it easier to develop agents that intelligent adapt their own parameters, and they make it easier to develop training-oriented adaptations. Thus, parameterization is already widely used throughout the CyCog framework, and our efforts at this point are to extend the use of the parameters to foster further types of adaptation.

5 Discussion and Conclusions

The ability to adapt is a key property of most behaviors that we would be willing to call "intelligent". Adaptivity is also a key component of training, because learning is fundamentally change, and effective training must adjust to the level of competence of a trainee. For the domain of cyber operations, in particular, adaptivity plays an even more prominent role, because the domain tasks themselves deal with constantly changing technology and tactics, as well as responding to (or preempting) constantly changing actions that others are changing to achieve they operational goals.

We have developed cognitive models of the major players involved in cyber operations (attackers, defenders, and users). We have also developed a number of approaches to adaptivity in training for interactive domains. We are in the midst of research efforts to combine the two, extending our cognitive modeling and training systems to support effective training in cyber operations. We have enumerated a variety of types of adaptivity that play a role in these research efforts, including adaptation of tactics and techniques, sophistication of scenario actors and role players, and personality parameters (broadly construed) of scenario actors and role players. The goal of all of these forms of adaptivity are to realistically represent the dynamic nature of cyber operations, while simultaneously providing an effective training framework that tailors trainee experience to specific pedagogical goals associated with each trainee.

Our current and future research plans aim to overcome a number of challenges to building a cost-effective but realistic cyber-operations ecosystem that supports domain-level and training-level adaptivity. We are particularly focused on methods of expanding the depth and breadth of knowledge for the role-player models, This includes efforts to develop knowledge representations that ease the acquisition of subject-matter expertise, as well as the engineering of that knowledge into composable goal hierarchies and formal definitions of TTPs. We are also pursuing research on learning and abductive inference methods to build robust models of self-adaptation, based on the ability to explain unexpected outcomes and novel experiences. Farther in the future, we hope to incorporate formal models of deception, which are a key component of advanced cyber operations, and open up an additional category of behavioral adaptivity [3].

References

1. Bowen, B.M., Stolfo, S.J., et al.: Measuring the human factor of cyber security. In: Homeland Security Affairs, IEEE 2011 Conference on Technology for Homeland Security: Best Papers, Suppl. 5 (2012)
2. Goguen, J.A.: Parameterized Programming. IEEE Trans. Software Eng. **10**(5), 528–543 (1984). https://doi.org/10.1109/TSE.1984.5010277
3. Henderson, S., Hoffman, R.R., Bunch, L., Bradshaw, J.: applying the principles of magic and the concepts of macrocognition to counter-deception in cyber operations. In: Proceedings of the 12th International Meeting on Naturalistic Decision Making. MITRE Corp., McLean, VA, June 2015
4. Huffman, S.B., Laird, J.E.: Instructo-Soar: learning from interactive natural language instructions (Video Abstract). In: Proceedings of the AAAI, p. 857 (1993)
5. Folsom-Kovarik, J.T., Newton, C., et al.: Modeling proficiency in a tailored, situated training environment. In: Proceedings of the Conference on Behavior Representation in Modeling and Simulation (BRIMS 2014) (2014)
6. Joint Chiefs of Staff. Joint Pub 3-12: Cyberspace Operations. Joint Chiefs of Staff (2018)
7. Jones, R.M., Furtwangler, S., van Lent, M.: Characterizing the performance of applied intelligent agents in Soar. In: Proceedings of the 2011 Conference on Behavior Representation in Modeling and Simulation (BRIMS), Sundance, UT (2011)
8. Jones, R.M., Laird, J.E., Nielsen, P.E., Coulter, K.J., Kenny, P., Koss, F.V.: Automated Intelligent Pilots for Combat Flight Simulations (1999)
9. Jones, R.M., Marinier, R.P., Koss, F.V., Bechtel, R.: Tactical behavior modeling for ground vehicles. Presented at the SAE World Congress Experience (2017). https://doi.org/10.4271/2017-01-0261
10. Jones, R.M., et al.: Modeling and integrating cognitive agents within the emerging cyber domain. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), vol. 20. Citeseer (2015)
11. Jones, R.M., Wallace, A.J., Wessling, J.: An intelligent synthetic wingman for army rotary wing aircraft. In: The Interservice/Industry Training, Simulation and Education (2004)
12. Jones, R.M., Wray, R.E.: Comparative analysis of frameworks for knowledge-intensive agents. AI Mag. **27**(2), 45–56 (2006)
13. Klein, G.A.: Sources of Power: How People Make Decisions. MIT Press, Cambridge (1998)

14. Kokar, M.M., Endsley, M.R.: Situation awareness and cognitive modeling. IEEE Intell. Syst. **27**(3), 91–96 (2012). https://doi.org/10.1109/MIS.2012.61
15. Lathrop, S., Hill, J.M.D., et al.: Modeling network attacks. In: Proceedings 12th Conference on Behavior Representation in Modeling and Simulation (BRIMS 2003) (2003)
16. Laird, J.E.: Millions of rules, billions of decisions. In: Presentation at the 29th Soar Workshop (2009)
17. Laird, J.E.: The Soar Cognitive Architecture. MIT Press, Cambridge (2012)
18. Laird, J.E., Newell, A., Rosenbloom, P.S.: SOAR: an architecture for general intelligence. Artif. Intell. **33**(1), 1–64 (1987). https://doi.org/10.1016/0004-3702(87)90050-6
19. Laird, J.E., Voigt, J., Derbinsky, N.: Peformance evaluation of declarative memory systems in Soar (2010). Manuscript submitted for publication
20. Maymí, F.J., Lathrop, S.D.: AI in cyberspace beyond the hype. Cyber Defense Rev. **3**(3), 71–82 (2018)
21. Newell, A.: Unified Theories of Cognition. Harvard University Press (1990)
22. Nielsen, K.: Secretary Kirstjen M. Nielsen's National Cybersecurity Summit Keynote Speech, 31 July 2018. https://www.dhs.gov/news/2018/07/31/secretary-kirstjen-m-nielsen-s-national-cybersecurity-summit-keynote-speech. Accessed 19 Jan 2019
23. Nonaka, I., Takeuchi, H., Umemoto, K.: A theory of organizational knowledge creation. Int. J. Technol. Manage. **11**(7–8), 833–845 (1996)
24. Parrish, K.: Cyber May Be Biggest Threat, Hagel Tells Troops (2013). http://www.defense.gov/news/newsarticle.aspx?id=120178
25. Pellerin, C.: Cyber Tops Intel Community's 2013 Global Threat Assessment (2013). http://www.defense.gov/News/newsarticle.aspx?ID=119776
26. Pfleeger, S.L., Caputo, D.D.: Leveraging behavioral science to mitigate cyber security risk. Comput. Secur. **31**(4), 597–611 (2012)
27. Ponemon Institute: Cost of a Data Breach Study: Global Overview. Ponemon Institute (2018)
28. Shute, V.J.: Focus on formative feedback. Rev. Educ. Res. **78**(1), 153–189 (2008)
29. Taylor, G.E., Sims, E.M.: Developing believable interactive cultural characters for cross-cultural training, San Diego (2009)
30. Taylor, G.E., Stensrud, B.S., Eitelman, S., Durham, C., Harger, E.: Toward Automating Airspace Management (2007)
31. van Lent, M.: Learning task-performance knowledge through observation. Ph.D. Thesis at University of Michigan, Department of Electrical Engineering and Computer Science (2000)
32. Van Lent, M., Fisher, W., Mancuso, M.: An explainable artificial intelligence system for small-unit tactical behavior, pp. 900–907 (2004)
33. Weld, D.S.: An introduction to least commitment planning. AI Mag. **15**(4), 27 (1994)