Chapter 15
Software Support for Teaching and Measuring Cognitive Readiness

Allen Munro

15.1 Cognitive Readiness for Tasks

People are more or less well prepared to perform particular tasks. Someone who has never heard of the game of chess, who has never seen a game, and who has not been told how to play the game is clearly not cognitively ready to play chess. The person might be cognitively ready to *learn* to play chess, but that is a different matter. On the other hand, someone who has played in many tournaments and who has attained Master status is cognitively ready to play chess at a very high level, at least so long as that person is not cognitively impaired by drugs, alcohol, or some other agency.

Does it make sense to speak about cognitive readiness outside of the context of a task? That is, can we say that person A is in some sense "cognitively ready" and that person B is not cognitively ready, no matter what the task? Someone who suffers from severe intellectual deficits is likely not to be cognitively ready for most tasks. But this is not a very useful way to use the term cognitive readiness, because there are already other constructs, such as *intelligence*, that seem to fill the bill well enough. Someone might be an expert chess player or an outstanding military tactician, but that does not mean that they are cognitively ready to compose a symphony.

A. Munro (✉)
Center for Cognitive Technology, Rossier School of Education, University of Southern California, 250 North Harbor Drive, Suite 309, Redondo Beach, CA 90277, USA
e-mail: munro@usc.edu

H.F. O'Neil et al. (eds.), *Teaching and Measuring Cognitive Readiness*,
DOI 10.1007/978-1-4614-7579-8_15, © Springer Science+Business Media New York 2014

Cognitive readiness must be restricted to types of tasks. (O'Neil, Lang, Perez, Escalante, and Fox in this volume describe an approach in which some cognitive readiness attributes are treated as being relatively task independent.)

15.2 Levels of Cognitive Readiness

When it is said that someone is cognitively ready to perform some type of task, there is an ambiguity about the level of skill or expertise that they will exhibit when performing examples of that task. One person might be cognitively ready to perform a simple example of the task, but not be able to perform well on a more complex instance of that type of task. Such a person could be said to have a moderate level of cognitive readiness for that task. Another person might have expert knowledge about the task and all its variations. When motivated to engage in the task, that person would exhibit a high degree of cognitive readiness for the task.

15.2.1 Levels of Cognitive Readiness in Complex Decision-Making Tasks

For several years, I have been involved in research on learning decision-making and problem-solving skills in the US Navy contexts. The structure of these real-world tasks leads me to propose five levels of cognitive readiness for such tasks. In the following list, the lowest level or simplest type of cognitive readiness appears first on the list, and the most complex type appears last.

- Cognitive readiness for categorization
- Cognitive readiness to act on the basis of a categorization
- Cognitive readiness to accommodate special conditions in procedure execution
- Cognitive readiness for task complexity (such as the occurrence of instances of multiple concepts, which requires prioritizing or combining procedures)
- Cognitive readiness to generate new types of solutions to novel problems

The "categorization" level of cognitive readiness can be seen as related to situational awareness (Endsley, 1995). In situations in which a decision-maker perceives relevant events and comprehends their import well enough to label the situation correctly, the decision-maker has used situational awareness to usefully categorize the situation.

Someone who knows what to do and how to do it, based on the categorization, can be said to be cognitively ready to act. The two higher levels of cognitive readiness listed above require the application of metacognitive strategies.

15.2.1.1 The *TAO Sandbox*[1]: A Problem-Solving Task

Tactical Action Officers (TAOs) are Navy surface fleet Department Heads who must coordinate information and recommend tactics to the captain during tactical engagements. The TAO Sandbox is based, in part, on an earlier research product that was designed to support TAO planning for antisubmarine warfare (ASW), called the ASW Sandbox (Auslander, Molineaux, Aha, Munro, & Pizzini, 2009; Munro & Pizzini, 2009; Munro, Pizzini, & Bewley, 2009). The TAO Sandbox extends that earlier product in many ways, but principally by supporting TAO planning for Air Defense and for Surface Warfare. In both the TAO Sandbox and its predecessor, a problem-solver can develop a plan and then let simulated time pass to see how the plan plays out. Plans can be reformulated and changes made during the course of a session. As the student uses the Sandbox, every action is recorded in a file. Sessions can be replayed in the Instructor Mode of the Sandbox, which is also used to develop new scenarios.

The TAO Sandbox has been adopted for a variety of uses in the training of TAOs, in the Department Head course at the Surface Warfare Officers School (SWOS). Instructors and students use the TAO Sandbox in a number of ways, including these:

- Instructors build simple scenarios to illustrate particular tactical concepts.
- Instructors illustrate these concepts by using the scenarios in class. In some cases, they record a scenario session in advance and then use the playback feature to present the recorded session, while providing commentary.
- Students solve illustrative tactics problems, often in small groups.
- Students present their tactical solutions, debriefing in the context of a recording of their sessions.
- Some highly motivated TAO students experiment with the Sandbox, building their own scenarios in different combat contexts, such as ASW, surface warfare, or air defense.
- The Sandbox is sometimes used to help assess a student's knowledge of tactics.

SWOS management has identified potential areas for additional utilization of the TAO Sandbox, including the training of littoral operations, which are close to shore, and expeditionary warfare tactics and procedures, which involve landing and supporting ground forces from the sea.

[1] The term Sandbox is an analogy to the physical sandboxes, usually raised on legs, that are used to teach tactical principles to Army and Marine officers, where objects representing tanks, firing teams, squads, platoons, artillery pieces, etc. are placed in terrains modeled using the sand in the box. Our computer-based Sandbox for teaching about Navy tactics allows placements of naval units to model tactical situations. In addition, however, it can simulate the passage of time and automate the movement of units.

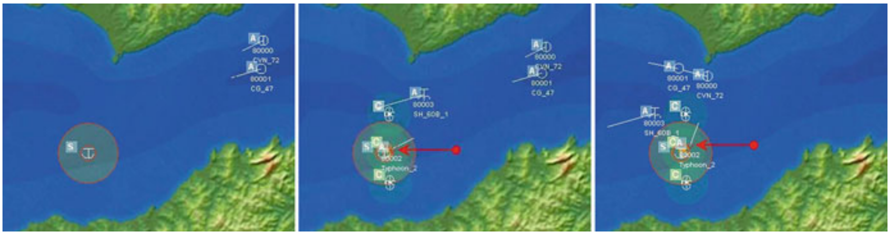

Fig. 15.1 Three states of an imagined transit scenario

The above-listed uses of the sandbox have replaced a number of less interactive methods for presenting and practicing tactics, including the use of static sketches on a white board and the use of canned animations or videos. When assessments are conducted in the context of the TAO Sandbox, the student produces a set of actions that have effects in the simulated tactical areas, rather than writing a short essay about a selected tactical approach and its expected outcomes.

TAO Sandbox "problems" are scenarios that students can "solve" in a variety of ways that can include deploying various sensors to detect potential hostile units and positioning one's own ships to avoid threats, or in such a way that more vulnerable units are protected by more capable units, by issuing appropriate queries and warnings to potentially hostile units, by targeting and attacking hostile units when appropriate, and so on. The combination of actions that constitute an acceptable solution [as judged by human experts in after-action reviews (AARs), for example] will vary, depending on the initial conditions of the problem and on the autonomous actions taken by hostile and neutral units during the course of a session.

One of the advantages of the Sandbox over previously utilized methods is that it does not rely on *mental* simulation of expected results. Working memory and other processing demands make it difficult to accurately predict the changes in relative positions of the involved ships, submarines, and aircraft over time. The built-in logic of the Sandbox handles these effects, and others, such as automated hostile behaviors on the part of simulated opponents, as conditions warrant during a session.

Cognitive load theory is, according to Paas, Renkl, and Sweller (2004), "concerned with the learning of complex cognitive tasks, where learners are often overwhelmed by the number of information elements and their interactions that need to be processed simultaneously before meaningful learning can commence" (p. 1). The theory assumes a working memory of limited capacity. When one attempts to solve a complex tactics problem by mentally simulating the interactions of a number of ships over time, working memory limitations may lead one to oversimplify or to ignore potential interactions between even simple events, such as the independent movement of different ships and how that will affect whether those ships will enter the ranges of each other's weapons systems.

Figure 15.1 represents possible states of a mental simulation of a transit that is threatened by a reported hostile submarine. The circle at the lower left is the Torpedo Danger Area (TDA), an expanding area of risk that may contain the hostile submarine.

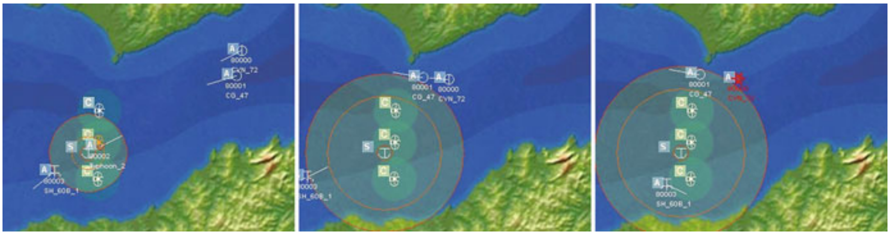

Fig. 15.2 Three states of a simulated scenario

The first state illustrates the initial condition. The two friendly ships (a cruiser labeled CG_47 and an aircraft carrier labeled CVN_72), which are on the right, are to proceed to the west through the Strait of Gibraltar. There is a concern with the report of a possibly hostile submarine near the center of the TDA.

In the second state shown in the figure, the tactician imagines having launched a helicopter that carries sonobuoys and having placed three of the sonobuoys in a line crossing the TDA. He imagines that one or more of the sonobuoys will detect the hostile sub. In this part of the figure, an arrow points to the detected submarine position.

In the third state shown in Fig. 15.1, the tactician imagines that some time later, the sonobuoys are still tracking the hostile submarine and the friendly ships are successfully avoiding it and completing their transit of the strait.

When the same exercise is carried out in the context of the TAO Sandbox, however, the previously imagined results of the plan are shown to be incomplete and inaccurate.

The first part of Fig. 15.2 shows the results of a tactician actually attempting the problem solution that was merely imagined in the illustrations of Fig. 15.1. Sonobuoys have been dropped, and the hostile sub has been detected.

In the second part of the figure, we see that before the friendly units have exited the strait, the submarine has exited the zone of detection of the deployed sonobuoys. Apparently the *mental* simulation shown in Fig. 15.1 did not accurately reflect the relative speeds of the submarine and the friendly units.

In the third part of Fig. 15.2, the submarine has successfully launched a torpedo and destroyed the aircraft carrier, which was undoubtedly the Mission Essential Unit (MEU). (The MEU is the unit, typically an aircraft carrier or large amphibious ship, which must be protected, because the assigned mission cannot be carried out without it.)

When the tactician using the TAO Sandbox attempts to carry out the imagined plan, it doesn't take long to find out that the submarine escapes detection long before the friendly ships reach the potential danger zone. The tactician is operating in the dark and may be placing his or her ships at risk. So the Sandbox user restarts the scenario and tries a different solution, as shown in Fig. 15.3.

This time, when the probable course of the submarine is determined, additional sonobuoys are dropped along that course, so that the sub is continuously tracked. (The tactician also chooses a sonobuoy type with a longer battery life.) See the first part of Fig. 15.3. Because the sub is moving toward the north, the friendly ships are

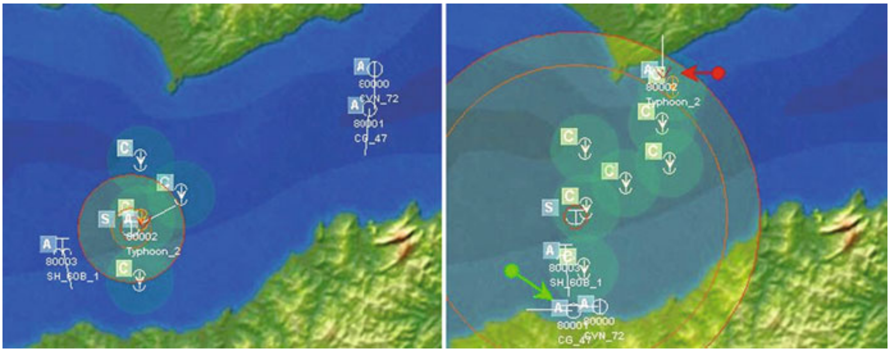

Fig. 15.3 Another attempt—same scenario

turned south and then back to the west, so that they will pass well to the south of the torpedo range of the sub in its known position. See the second part of Fig. 15.3. Here the arrow above marks the tracked location of the submarine; the arrow at the lower left marks the position of the two friendly ships that have successfully evaded the hostile sub.

In addition to providing a more accurate practice environment than mental simulation, the TAO Sandbox offers a number of other features that promote learning. There is considerable evidence that opportunities to practice the application of knowledge with guidance in a variety of environments promote learning and the acquisition of expertise. This work has been reviewed and summarized very effectively by Clark, Yates, Early, and Moulton (2010) extending earlier work by Merrill (2002). Mayer (2004) also cogently reviews related studies that show the importance of combining direct instruction with practice. Practicing certain tactics, such as ASW, in real-world environments on board ships, dueling with realistic opponents, is a very slow, very expensive form of practice. Given the speeds at which submarines can operate in relative silence, ASW tactics can take the form of slow motion duels. A TAO can reasonably be exposed to only a very few such real-world exercises, both because of the time they consume and because of the great expense of devoting entire crews and many hours of ship time to carrying out the selected tactics. The TAO Sandbox, on the other hand, has features that enable time to be greatly speeded up. Some ASW scenarios can be conducted at up to 850 times the rate that it would take to carry them out in the real world. This means that a prospective TAO can be exposed to many different types of ASW scenarios in the time that it would take to conduct one scenario in real time.

15.2.1.2 Illustrating Types of Cognitive Readiness in the Context of the *TAO Sandbox*

Because it is a rich and highly configurable environment for providing practice, and one that has been adopted and is being utilized for training complex concepts and

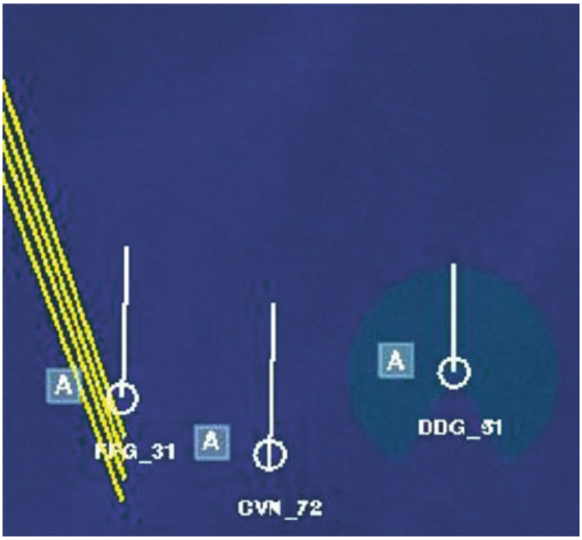

Fig. 15.4 Parallel lines of sound

decision-making, the TAO Sandbox is a good environment for illustrating the types of cognitive readiness training listed above.

Cognitive Readiness for Categorization. Just as a chess player must be able to categorize situations in a game, TAOs must be able to recognize types of tactical situations. When a surface ship is using passive sonar to search for a nearby submarine, the combat information center draws *lines of sound* that pass through the ships current position in the direction of a submarine that the sonar detects. (Decisions about whether to use active or passive sonar are often complex and highly dependent on context. Passive sonar detects by listening for the sounds made by another ship or a submarine; active sonar transmits a loud pinging sound through the water and analyzes the echo of the sound when it bounces back from an object. A much-oversimplified way of thinking about it is to say that passive sonar is more appropriate if you really don't want the submarine to detect your ship. Using active sonar makes your ship detectable at many times the normal range. On the other hand, when the opposing submarine is very silent, passive sonar may not be able to detect the sub even when it is very close, and only active sonar has a reasonable chance of detecting the enemy. At other times, active sonar can be used to "chase away" a submarine that does not want to engage or even to be detected.) Plotting a new line of sound every few minutes using passive sonar produces a *pattern* of lines of sound, when there is a sustained sonar contact. Different patterns imply different types of relative motion relationships between the surface ship and the submarine.

When the lines of sound are parallel to each other, it suggests that the surface ship—in Fig. 15.4, a frigate (a friendly warship, smaller than a destroyer) designated FFG_31—and the submarine are on a converging course. (Each ship has a white bearing line showing its direction of travel. Longer bearing lines reflect higher speeds.) It is important that the TAO recognize examples of this *converging* pattern,

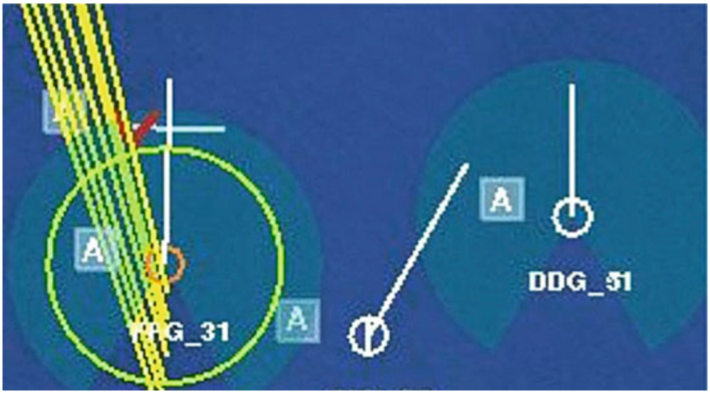

Fig. 15.5 Possible response to converging courses

because it is unlikely that an undetected, silently operating submarine could match speed and course with the frigate. Although the submarine's exact position is not known, it is clear that there is a submarine ahead off the port side (port is the left-hand side of an observer on the ship who is facing the bow) and that it is moving toward the frigate's course. Recognizing this pattern is an example of the first type of cognitive readiness: the ability to categorize objects, events, or conditions in the task context.

Cognitive Readiness to Act on the Basis of a Categorization. In many cases, a learner must learn how to react appropriately when recognition occurs, ordinarily by choosing to carry out an appropriate action or procedure for the detected condition. In the case of a TAO, depending on the current geopolitical situation and the ship's mission, there may be a number of appropriate responses to a categorization such as "possible converging course with hostile submarine."

In Fig. 15.5, the TAO problem-solver has decided on two courses of action. Because his mission calls for the preservation of the MEU, the tactician has turned the carrier, which is labeled CVN_72, to starboard (starboard is the right-hand side of the ship for an onboard observer facing the ship's bow), in order to keep it out of the reach of the submarine's torpedoes. In addition, the TAO has turned on the FFG's active sonar, which has found the exact location of the submarine (shown as a red V shape to the north of the FFG_31 position).

Another pattern that must be recognized is the *passing* pattern, shown in Fig. 15.6. When a series of lines of sound crosses like this, it most likely means that there is a submarine on the port side that is traveling in roughly the opposite direction that the surface ship is. The questions to ask are how near the submarine is and how fast it is traveling.

In this case, active sonar detects the sub quite close to the frigate, just after they have passed each other. Figure 15.7 shows the situation a short while later, with the passing pattern further developed and the position of the submarine detected by the frigate's active sonar.

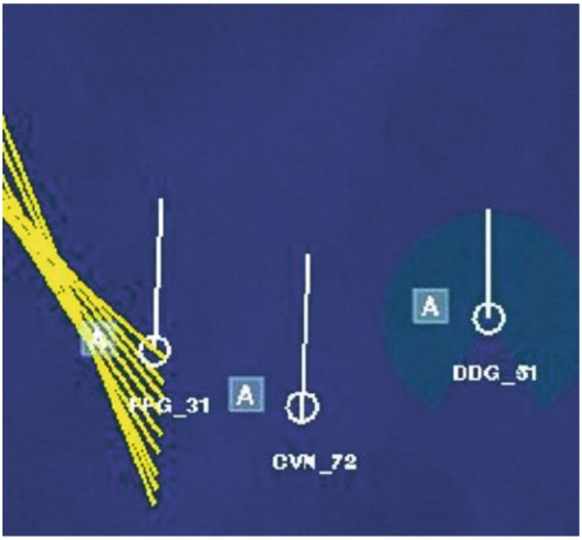

Fig. 15.6 Passing pattern in lines of sound

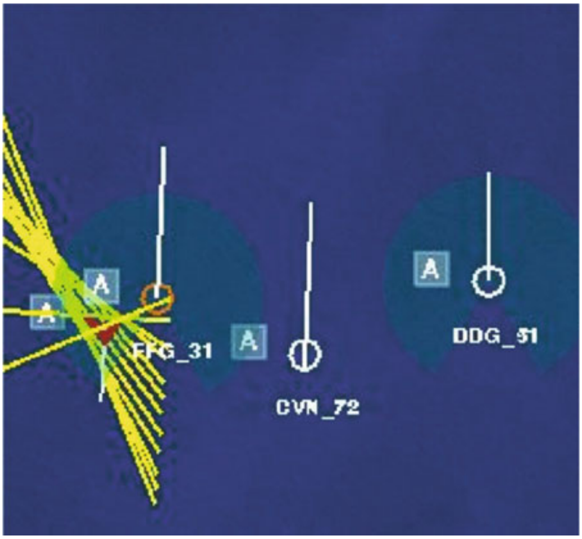

Fig. 15.7 Submarine revealed as it passes

There are many other types of tactical situations that can be categorized in the TAO Sandbox. In addition to many other patterns in lines of sound from passive sonar, there are conditions related to potential danger areas that friendly ships should try to avoid. The emergence of unknown radar contacts with characteristics that mark them as potential threats that should be labeled as *Critical Contacts of Interest* (CCOI) is another.

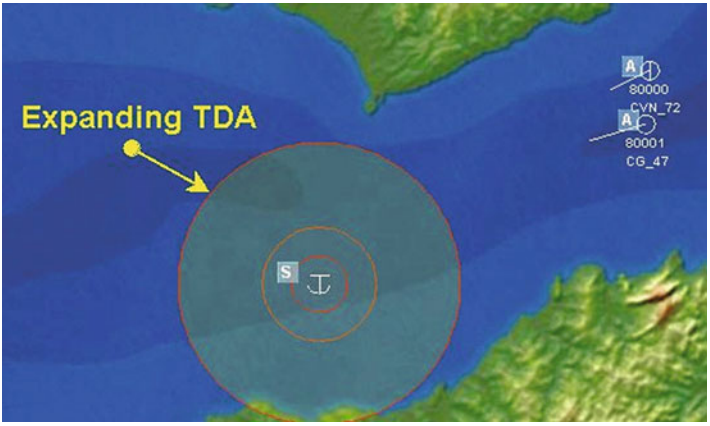

Fig. 15.8 The Torpedo Danger Area (TDA) will expand to fill the strait

In general, training simulations and games can contribute to the training process if they automatically detect opportunities for categorization. In some cases these simulations or games might provide hints or guidance when such conditions are recognized. More generally, however, a simulation or game can inform a collaborating instructional component of such conditions. The instructional component could be a human instructor that is prepared to assess performance or to offer guidance or instruction. Or the instructional component can be a software application that makes analyses based on student actions and emergent conditions, as reported to it by the simulation or game.

Cognitive Readiness to Accommodate Special Conditions in Procedure Execution. Most complex procedures include conditional branching. For example, in ASW, a report of evidence of a hostile submarine at a particular time near a specific point is used to create a point called a datum, with an expanding circle of threat area. Actually, there are two expanding circles: the Furthest On Circle (FOC), which represents the furthest that the now-hidden sub might have traveled at its maximum silent operating speed, and the TDA, which includes the maximum effective attack distance for the torpedoes with which the submarine is presumed to be equipped. In general, TAOs are expected to work to keep friendly ships and especially MEUs out of TDAs.

Sometimes, however, operations may take place in restricted water space that precludes avoidance of the TDA. Normal procedures for avoiding a TDA will not work in such circumstances. Consider the case of the nuclear aircraft carrier CVN 72 and its escort, the guided missile cruiser CG-47. They are to travel west through the Strait of Gibraltar, but there has been a report of a hostile submarine in the strait, originally located at the center of the circle in the figure shown at the left, below. By the time the friendly ships get there, the TDA will have expanded to fill the western end of the strait. Avoidance by maneuvering alone is not possible. See Fig. 15.8.

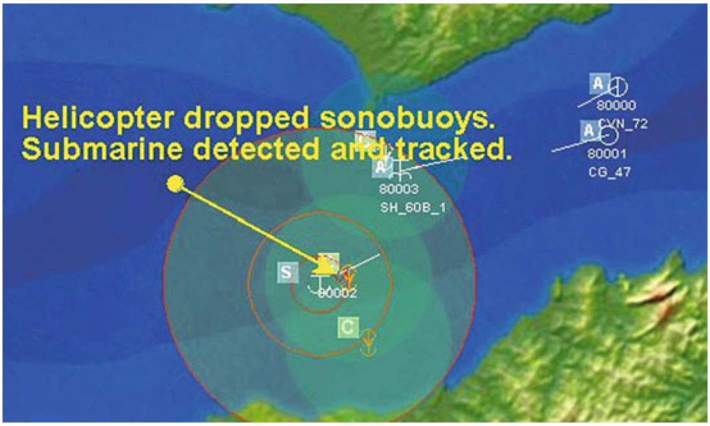

Fig. 15.9 After using sonobuoys to detect the sub, parts of the TDA are safe

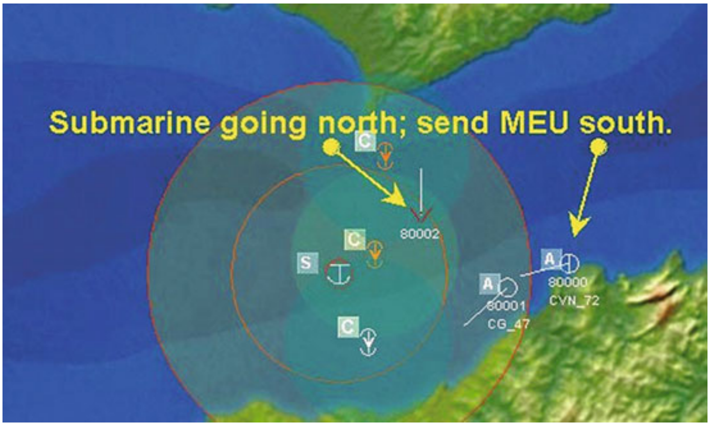

Fig. 15.10 Sending the MEU south to avoid the detected submarine

A solution, in this case, is to launch a helicopter and drop sonobuoys within the TDA to determine the actual position of the submarine. Figure 15.9 shows that two sonobuoys have detected the submarine. The submarine is tracked as moving toward the north-northeast. In this case, the TAO can plan to enter the southern part of the TDA, because the submarine position is known, and it is too far to the north to detect the friendly ships, if they are not moving too fast, and it is too far for its torpedoes to reach the friendly ships. So this can be viewed as an exception to the rule about entering the TDA. See Fig. 15.10.

Alternatively, the TAO can move the datum to the position at which the sub was just detected. The TDA for this new position will be much smaller, because the sonar is showing exactly where the sub is. If the tactician takes this approach, the

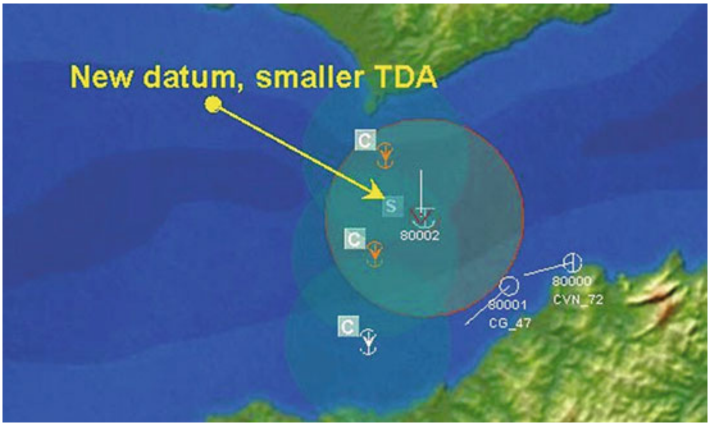

Fig. 15.11 A new datum, a smaller TDA

Fig. 15.12 Dealing with two types of tasks at once

route to the south does not violate the “don’t enter a TDA” rule, because the TDA is now smaller and centered further north, Fig. 15.11.

If the TAO player chooses to either route a path through a “known safe” part of a TDA or chooses to relocate (and thereby resize) the datum and its TDA circle, he or she thereby demonstrates a cognitive readiness to accommodate special conditions in the execution of the “avoid TDA” procedure.

Cognitive Readiness for Task Complexity. There are many possible types of task complexity. For example, someone might have to recognize the need to carry out two procedures at the same time. In Fig. 15.12, the TAO is dealing with attacks from both air and surface units.

Fig. 15.13 Executing two procedures at the same time

Two different procedures need to be conducted at the same time. First, the tactician has to notice and defend against each incoming missile with one or more Surface to Air Missile (SAM) launches. This is a surface unit's major defense against air attack, including surface missile attacks. Second, the source of most of the missile launches, the corvette in the upper left part of the scene, must be eliminated as an ongoing threat. (A corvette is a warship, typically smaller than a frigate.) That requires using a special type of attack, using multiple Harpoon missiles from different ships in a coordinated attack. (The Harpoon is a fairly long-range US ship-to-ship missile.) The defense systems of the corvette will be overwhelmed by the simultaneous strikes from different directions (Fig. 15.13).

A TAO trainee who carries out more than one type of procedure simultaneously shows cognitive readiness for task complexity.

Cognitive Readiness to Generate New Types of Solutions to Problems. Finally, in complex domains such as surface warfare tactics, professional tacticians need to be continually conceiving of new tactical situations and how they could be dealt with. How would ASW tactics change if an opponent's sub could attain higher speeds in subsurface travel than friendly ships can on the surface? If the subs are noisy enough, we could detect them coming, but how would we defend? It is unlikely that brilliant novel solutions to new problems occur in the heat of battle, but expert practitioners may "toy" with such imagined situations in advance of need. They could devise possible approaches to these new solutions and could consult with other expert practitioners about such problems and solutions. A tool like the TAO Sandbox might be able to play a role in encouraging the development of "what-if" scenarios that could eventually be used in real life.

The development of novel situations and solutions can be supported in the Sandbox when users have access to Instructor Mode. In this mode, novel problems can be designed (such as dealing with an attack submarine that is faster than targeted friendly ships). At present, the Sandbox is not instrumented to monitor and report on scenario development. However, once a novel scenario has been designed, it can be tested in Problem Mode, which does report actions and events for evaluation.

15.3 Software Architectures for Assessing Cognitive Readiness

In order to assess cognitive readiness in the context of a game or simulation, there must be an evaluator component that observes performances in that context and makes judgments. The game or simulation will typically need to report three types of information to the evaluator component:

- Initial conditions that may be relevant for assessing performance
- Meaningful actions taken by the player in the game
- Emergent events that may contribute to the evaluator's assessment of the player

Assessment may be conducted naturalistically, by giving the user of the simulation (or the player of a game) a problem and then making judgments based on the actions and events that occur during that problem session. Alternatively, more artificial or staged assessments can be conducted, by presenting a particular problem situation, perhaps with an event history that brought the user to this point, and then offering a menu of possible actions to the user and judging the menu choice made. This approach would be more like conventional assessment and would permit a more controlled presentation of "test items." This chapter deals with the more challenging design and implementation of "naturalistic" assessments in the context of valid problems or scenarios.

A software architecture like the one sketched below would be an appropriate one for conducting such naturalistic assessments in the context of game scenarios or simulations (Fig. 15.14).

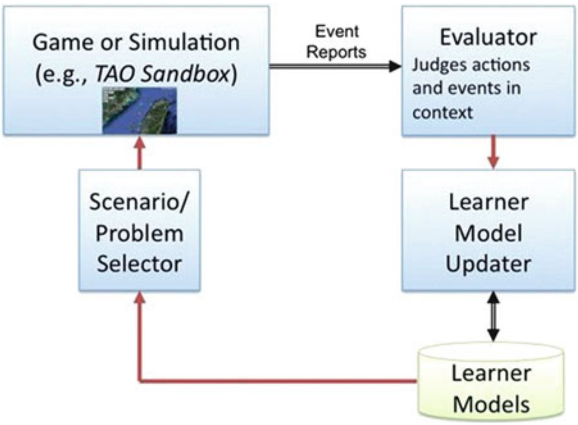

Fig. 15.14 Architecture for assessment in simulation contexts

The evaluator object uses reports of events (and relevant initial conditions) to make judgments about what the player of the game knows about the domain and to evaluate the user's play. These judgments, which could be quite detailed—that is, they could reflect detailed elements of domain knowledge—are passed to an object that is responsible for updating the model of the learner. Learner models are stored in a database. The Learner Model Updater needs to be able to read the model for the current student, because decisions about the goodness of certain actions may be based, in part, on an assessment of how sophisticated the learner already is in the domain. When it is time to select a new scenario or problem in order to continue assessing the learner, the learner models database is accessed, and the selector chooses a problem that will reveal some aspect of learner knowledge that has not already been adequately tested in previously presented scenarios.

Note that this architecture does not specify *how* the next problem or scenario is to be selected. That is a matter for particular implementations of the architecture. One approach would be to store meta-data with problems that reflect the concepts and skills required to perform well on the problem. This map of problem-relevant knowledge could be compared with the current learner model for a particular student, and the problem chosen could be the one which utilizes mostly understood concepts or skills in evidence, but that also requires an increment of additional knowledge or skill that is not yet attested in the learner model.

The evaluator in this architecture must have detailed information about how to judge the goodness of actions and events in particular contexts. One way to achieve this is to code a special-purpose evaluator, adhering to a module interface standard, for each game or scenario that must be evaluated. Another way would be to use a generic evaluator, but to import a set of evaluation rules for the specific domain. These rules could be a set of conjoint Boolean expressions, specifying, for example, that if threat values are low, then entering a TDA is OK, but, if threat values are high, then entering a TDA, especially for the MEU, is bad.

Evaluator judgments can serve to stimulate instructional interactions either during a scenario-based practice session or during an after-action review (AAR).

An expanded version of this architecture for evaluating in game contexts can also be utilized to teach in the context of the game or simulation, as shown in Fig. 15.15.

In this case, some of the evaluator's judgments are also passed directly to an interactive instruction Control object, so that the player can have obvious mistakes pointed out immediately. In addition, however, the learner model updater's synthesis of current evaluations with the learner model can produce insights that can be conveyed under the control of the Instruction Control object. In the above diagram, there is an optional element called the instruction/explanation interface. Some games and simulations might provide interface elements for presenting newly generated explanatory text in the game context. The TAO Sandbox can present very brief text messages. But many games will not have such a subsystem, so a separate module should be available for instructional presentations, especially longer ones. Note that the explanation interface would not have to be textual: text to speech, prerecorded voice, videos, or animations could be presented under the control of the interactive instruction component. Different implementations of the instruction/explanation interface might be utilized for different media types.

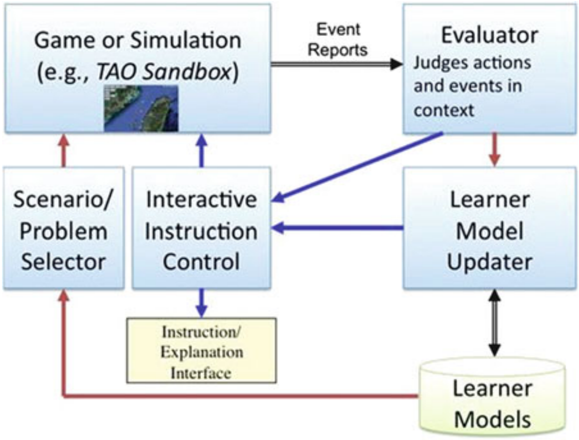

Fig. 15.15 Architecture for assessing and instructing in simulation contexts

15.4 Software Support for Assessing Cognitive Readiness

A game or simulation can support assessment of cognitive readiness by a collaborating application that evaluates student actions in context, such as the evaluator object in the above figures. To be practical, the game must report at a level that is meaningful in terms of the game-player's tasks. That is, if the game only reported that, for example, "the left mouse button was clicked at the point [293, 732]," the action evaluation component would have to maintain its own internal simulation of the game domain in order to make sense of such low-level information. Examples of meaningful reports in the TAO Sandbox would have meanings such as "player launched a SAM from CG-47 against surface-to-air-missile-16 at 14:22:05" or "player turned CVN-72 to bearing 30 at 16:05:00." Here is an actual example of a report of a user action:

```
00:07:29 .k.s_CG_47. LoadandLaunchSH_60B CG_47
```

This means that at the virtual game time of 7 min and 29 s, the cruiser CG-47 loaded and launched an SH-60B helicopter, a helicopter equipped to assist in ASW. (Note: the ".k.s." is part of the simulation's designation of the cruiser CG_47. It can be ignored). The next message from the Sandbox provided specific information about where the helicopter appeared, and what its initial heading and speed were:

```
00:07:29 .k.helo_master. LaunchHelo SH_60B3 400 353 324 .k.s_CG_47.
```

This means that the SH-60B helicopter was launched at the location [400, 353] in the coordinate system used by the Sandbox and that its bearing was 324°.

It is not enough, however, to simply report player actions. It is also necessary that the game must detect and report the occurrence of instructionally relevant emergent events. For example, the TAO Sandbox has semi-intelligent hostile submarines. When they are assigned hostile behavior, they actively search for surface ships using their passive sonar. When they detect one, they move toward it until they come within visual range. Then they briefly raise their periscopes to see whether the sonar hit is actually a desirable target. If it is, they submerge again and close to torpedo range and then fire. This autonomous, rule-based behavior interacts with the results of actions previously taken by the game player. Torpedo attacks cannot be precisely scripted in advance, because they will only occur if the player maneuvers in such a way that the submarine can detect and close on a friendly ship position. Such *emergent* events must be reported by the game to the analyzing evaluator or that component will be unaware of them. There are many examples of such emergent events that the evaluator might need to know about, including a friendly ship entering a TDA, a friendly ship coming within torpedo range of a hostile sub, and a torpedo being fired by a hostile sub. Without being told that a torpedo was launched, the evaluator would have no idea why the targeted friendly ship was taken to flank speed by the user and put through a rapid succession of bearing changes. Here is an example of the report of an emergent event:

```
00:59:45 .k._happenings. friendly_in_TDA CVN_72
```

When the simulation detected that one of the user's ships (a carrier named CVN_ 72) entered a TDA, it announced to the evaluator component that there was a "friendly_in_TDA" event. (This announcement was made by the simulation's .k._happenings object.)

At almost one (virtual) hour into the session, the carrier CVN-72 entered a TDA. Another type of emergent event occurred just over 15 min later, when the carrier was torpedoed:

```
01:15:22 .k._happenings. friendly_killed CVN_72
```

This time the .k._happenings simulation object announced that there was a "friendly_killed" event involving the carrier CVN_72.

A third type of information that a game must provide to an evaluator component is aspects of the initial state of a game session. For example, if a submarine from a country with which the friendly force is at peace is the source of a datum, then that TDA probably does not actually represent a danger area. In this circumstance, it may be appropriate not to avoid the TDA, but to simply go through it on the way to one's appointed task. Initial threat levels and other aspects of the initial state of the game must be reported. In the TAO Sandbox, it is the instructor who authors a scenario who determines what threat level a student should expect. The author writes a mission briefing that describes the situation in sufficient detail that the user should be able to estimate whether there is no threat at all (level 0) or a very high threat level (10), representing open war with a competent opposing force. Because the Sandbox cannot read and understand the briefing, the author must give the Sandbox a clue about the threat by specifying a number in the range 0–10. At run-time, the Sandbox will tell the evaluator about this threat, so that it can take the threat level into account in judging the users actions. (For example, going through elaborate maneuvers that delay accomplishing a mission in order to avoid entering sensor range of an allied submarine—a low threat situation—would not make sense. Going through the same maneuvers in a time of war, to avoid a hostile submarine, might be the correct course of action. The evaluator component has to know the threat level to accurately evaluate actions in context.) In a sense, such initial reports are a special case of emergent events: they are all the events that emerge the instant the game session starts. Here is an example of such an initial state element:

```
00:00:00 Threat_Level 0
```

This means that this scenario is one that, at least initially, is conducted in an environment that is one of relative peace and security. The evaluator will not expect TDA avoidance maneuvers to be carried out in such a scenario.

15.5 Software Support for Teaching Cognitive Readiness

Many of the low-level services described by Munro (2007) that a game or simulation should provide for training will be required by an interactive instruction controller that aims to provide a level of cognitive readiness for specific tasks. These low-level services include these:

Highlight object
Unhighlight object
Set game internal value
Emulate user action
Register interest in an occurrence
Report occurrence

An instructional component can provide remediation or explanations in the simulation context by utilizing some of these services. A request to highlight an object is interpreted by the Sandbox as a direction to create a large semitransparent arrow pointing at the object. Comment objects, which are textual elements that can be made to appear in the simulation window, can be given text values, such as "Using active sonar when trying to avoid a hostile submarine is dangerous," if the instructional component wants to present that message to the simulation user.

Let us consider the five types of cognitive readiness that we listed at the beginning of this essay, looking at them in the context of the TAO Sandbox.

- *Cognitive Readiness for Categorization*. To teach about categories of situations, it is necessary to draw attention to salient characteristics of examples. One way to do that is to highlight graphical elements that illustrate those characteristics. Setting the internal value that names the file with the current scenario data causes that file to be read into the TAO Sandbox.
- *Cognitive Readiness to Act on the Basis of a Categorization*. In addition to using the highlight and unhighlight features, in order to teach a user what to do after recognizing a salient condition, it may be necessary to demonstrate a procedure under control of the instruction module. That requires the use of the *emulate user action* service.

- *Cognitive Readiness to Accommodate Special Conditions in Procedure Execution*. In order to teach learners how to respond to special conditions, the instructional control module needs to recognize when those conditions arise. One way to do that is to register an interest in those conditions and then to receive reports of their occurrence. To teach how to respond to those specific situations, emulating user actions may again be required.
- *Cognitive Readiness to Categorize Complex Situations and Act Appropriately, Prioritizing or Combining Procedures*. Teaching users how to recognize simultaneous conditions and the need to carry out combined procedures or procedures done in parallel requires providing many practice examples, using highlighting, explanations, and demonstrations, as required. The work of Sweller and Cooper (1985) showed that worked examples contribute to effective learning when combined with opportunities to solve problems. Scenario demonstrations play the role of worked examples in TAO Sandbox training.
- *Cognitive Readiness to Generate Solutions for Novel Problems*. Teaching people how to generate new types of solutions to novel problems is the area in which we are least prepared to offer specific software support. The problem is that, if the solution really is novel, it will take an extremely intelligent software system to come up with useful novel solutions on its own. (If this is possible, perhaps the software should be solving the problems, rather than training human beings to do so!) Less ambitiously, the training system might try to evaluate the goodness of a novel solution. But if the novel solution actually violates some of the standard metrics of "goodness," even this could be quite difficult to do. Novel solutions are most likely to be achieved by expert, motivated, professional practitioners who are willing to "play" with solutions to hypothetical problems. Perhaps the best software support that we can offer for this level of cognitive readiness is easy-to-use practice environments that promote trying alternatives and sharing interesting outcomes with others.

15.6 Implementation

The TAO Sandbox is a simulation that has been implemented and continues to evolve as it is used to develop and deliver interactive scenarios for students at SWOS. Instructors and the professional in-house staff of the school utilize the Sandbox in authoring mode to build tactical planning problems in a variety of simulated contexts. This simulation has been instrumented to report actions and emergent events to a collaborating evaluator component. Ongoing collaboration with the designers and developers of one such component, the CRESST Assessment Application (CAA) is currently underway. The Sandbox is undergoing additional instrumentation in order to report to the CAA on the goodness of pedagogically relevant actions and events. The CAA will use a Bayes Net to summarize high-level estimations of the competence of the problem-solvers.

The TAO Sandbox also offers services to support instruction. Working with our CRESST colleagues, we are developing approaches to generating and delivering instructional interactions in the context of the Sandbox. These features will constitute the interactive instruction component shown in the architecture diagrams above.

15.7 Conclusions

Five levels of cognitive readiness for performing tasks have been described. Each of these levels has been illustrated in the context of a game-like tactical planning tool, the TAO Sandbox, which is used for conceptual training of TAOs. A simple architecture for assessing cognitive readiness in the context of games and simulation objects such as the TAO Sandbox was presented, together with a somewhat more complex architecture for teaching in such contexts. Games and simulations that are to work with assessor/evaluator components in a training system must report instructionally relevant elements of initial problem states, meaningful actions taken by users, and certain potentially relevant emergent events. Instruction in such contexts requires that the game or simulations offer certain low-level services to an interactive instructional component. When games and simulations that are to be used to help people learn to perform tasks are developed with such services, their potential to play a useful role in advanced training systems is much greater than it would otherwise be.

Acknowledgements The work reported herein was partially supported by a grant from the Office of Naval Research, Award Number N00014-08-1-0126. The findings and opinions expressed here do not necessarily reflect the positions or policies of the Office of Naval Research.

References

Auslander, B., Molineaux, M., Aha, D. W., Munro, A., & Pizzini, Q. (2009). *Towards research on goal reasoning with the TAO Sandbox* (Technical Note AIC-09-155). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research on Artificial Intelligence.

Clark, R. E., Yates, K., Early, S., & Moulton, K. (2010). An analysis of the failure of electronic media and discovery-based learning: Evidence for the performance benefits of guided training methods [electronic version]. In K. H. Silber & R. Foshay (Eds.), *Handbook of training and improving workplace performance, Volume I: Instructional design and training delivery*. Washington, DC: International Society for Performance Improvement.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*, 32–64.

Mayer, R. (2004). Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *The American Psychologist, 59*, 14–19.

Merrill, M. D. (2002). First principles of instruction. *Educational Technology Research and Development, 50*(3), 42–59.

Munro, A. (2007). Foundations for software support of instruction in game contexts. In H. F. O'Neil & R. S. Perez (Eds.), *Computer games and team and individual learning*. Amsterdam: Elsevier.

Munro, A., & Pizzini, Q. A. (2009). *The TAO Sandbox user guide*. Working paper, Center for Cognitive Technology, University of Southern California.

Munro, A., Pizzini, Q. A., & Bewley, W. (2009). Learning anti-submarine warfare in the context of a game-like tactical planner. In *Proceedings of the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. Retrieved from http://ntsa.metapress.com/ling.asp?id=u1q691546361167

Paas, F., Renkl, A., & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional Science, 32*, 1–8. Retrieved from http://meds.queensu.ca/ohse/assets/cogntive_load_theory_pass_2004_.pdf

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*(1), 59–89.