

Representing Context in Simulator-based Human Performance Measurement

Webb Stacy, Ph.D.¹, Danielle C. Merket², Maj. Matt Puglisi, USMCR¹, Craig Haimson, Ph.D.¹

¹Aptima, Inc.
Woburn, MA & Washington, DC
wstacy@aptima.com

²NAVAIR Training Systems Division
Orlando, FL
danielle.merket@navy.mil

ABSTRACT

During a deliberate attack on an insurgent-held city, a Marine infantry company receives fire from a small building next to a mosque. What should the artillery Forward Observer (FO) do? The answer depends on context. If the fire coming from the building causes casualties, the FO should conduct an Immediate Suppression mission. If the insurgents' fires do not have any effects on the Marines below and they can take cover, the FO needs to formulate a course of action with the company commander.

How would we measure FO performance in simulator-based training for this scenario? It's not enough simply to take obvious measurements like target location error or target/ammunition combination. We must have an understanding of the FO's context, and measure and assess the FO's performance accordingly. The performance measurement infrastructure in the training environment must support these activities.

In this talk, we discuss a formal representation of context for human performance measurement in immersive training environments and how that representation fits into an innovative language for expressing those measurements, Human Performance Measurement Language (HPML). We show how context plays a role both as triggers for measurements and as key information for assessments, and demonstrate a method for convenient elicitation of context information from expert instructor/operators. We provide illustrative examples of training mission contexts and show how they may be represented using this formalism.

We further discuss how we have implemented context representation capabilities in real-world simulator-based training situations, including a forward observer simulator operating in a variant of a Distributed Virtual Training Environment-based federation and an F/A-18 simulator flying in a Navy Aviation Simulation Master Plan-based federation. We conclude by discussing additional benefits of representing context in simulator-based training environments.

ABOUT THE AUTHORS

Webb Stacy, Ph.D., is Vice President of Technology at Aptima. He oversees Aptima's current and future technology portfolios. His focus is the intersection of software and computer science with the science, modeling, and measurement of warfighters as individuals and as teams. He has extensive experience in the development of mission critical software, and holds a Ph.D. in Cognitive Science from the State University of New York at Buffalo and a B.A. in Psychology from the University of Michigan.

Danielle C. Merket is a Senior Research Psychologist for NAVAIR Orlando Training Systems Division. Ms. Merket leads the Distributed Brief/Debrief and Performance Measurement team within the Navy Aviation Simulation Master Plan and the Integrating Instructor Workload Reduction Tools effort within the Air Warfare Training Development IPT. Ms. Merket has 9 years of experience conducting advanced R&D in the areas of tactical aviation training, distributed training systems and aircrew coordination training. Ms. Merket holds a M.S. in Industrial and Organizational Psychology from the University of Central Florida.

Maj. Matt Puglisi, USMCR has expertise in military and health care policy, and the health effects of military service. He has represented membership and trade associations before Congress and the Executive branch, and has served on two expert panels convened by the National Academy of Science's Institute of Medicine. Mr. Puglisi served as an artillery forward observer for the First Battalion, Eighth Marines during Operation Desert Storm and commanded Battery G, Third Battalion, Fourteenth Marines. His personal decorations include the Navy Commendation Medal, Navy Achievement Medal, and the Combat Action Ribbon. Mr. Puglisi holds a Masters in Public Policy from Georgetown University, and a B.A. in Political Science from Siena College.

Craig Haimson, Ph.D., specializes in the development and delivery of automated training and assessment systems. His primary interests are in cognitive modeling, critical thinking, and decision making. Dr. Haimson applies his expertise to simulation-based training and assessment systems for domains such as intelligence analysis, information operations, and dismounted infantry operations. His work includes training teamwork skills using synthetic agents, expert system models, and multiplayer online gaming environments. Dr. Haimson received a Ph.D. in Cognitive Psychology from Carnegie Mellon University and a B.A. in Psychology from Harvard University.

Representing Context in Simulator-based Human Performance Measurement

Webb Stacy, Ph.D.¹, Danielle Merket², Maj. Matt Puglisi, USMCR¹, Craig Haimson, Ph.D.¹

¹Aptima, Inc.
Woburn, MA & Washington, DC
wstacy@aptima.com

²NAVAIR Training Systems Division
Orlando, FL
danielle.merket@navy.mil

WHY CONTEXT MATTERS



A Marine infantry battalion moving north during a deliberate attack on an insurgent-held city receives fire from a small building next to a mosque. The mosque compound is located on the east side of a control line, a road running from north to south. The battalion's area of responsibility is to the west of the road, and another battalion's area of responsibility is to the east of the road. What should the artillery Forward Observer (FO) attached to the company do? The answer depends on the context of the situation. If the fire coming from the building causes casualties among the Marines in the infantry company, then the FO should conduct an Immediate Suppression mission. However, in this case, the artillery fires might have effects on the Marines in the adjacent battalion on the east side of the road.

If the insurgents' fires do not have any effects on the Marines below, and they can take effective cover, then the FO needs to formulate a course of action with the company commander. Using an area fire weapon next to a mosque can risk damage to the mosque. The company commander, in turn, will need to push this problem up the chain of command. The insurgents are holding up the battalion's attack. Ultimately, the regimental commander authorizes an M-1 tank to fire on the building with its main gun. He selects a direct fire weapon rather than an area fire weapon, thereby confining the effects of the fires to the building and not the adjacent mosque. The company's Marines can now continue the attack.

How would we measure FO performance in the scenario described above? The usual measures such as target location error or target-ammunition combination are not sufficient. We must take the context of the situation into account, in this case the effects of the insurgents' fires. If these fires are not

having immediate effects on the Marines in the rifle company, then the FO should not call for fire on the building next to the mosque. If, however, the fires are killing and wounding Marines in the company, then the FO must look for Marines from the adjacent battalion across the road. If he observes them near the mosque compound, then he cannot call for fire on the building and create even more Marine casualties. If he observes adjacent battalions, then he must call for an Immediate Suppression and expect that higher headquarters will block the mission if indeed Marines from the adjacent battalion are too close to the mosque compound. In the final analysis in this small vignette, not calling for fire is the right course of action.

Our performance measurement work in these environments has been motivated by the difficulty of obtaining informative performance measurements in immersive training environments. The availability of raw data is rarely a problem; but selecting, defining, and computing useful measurements based on those data—measures that capture important aspects of trainee performance and that do not overwhelm participants with irrelevant information—is not trivial. Human Performance Measurement Language (HPML) and the infrastructures that use it provide a practical means for transforming raw data into meaningful measurements. Our formal analysis of context is thus shaped to be useful because it is embedded in HPML. This paper concerns ongoing work for the U.S. Navy (Stacy, Merket, Freeman, Wiese, and Jackson, 2005; Stacy, Freeman, Lackey, and Merket, 2004; Lackey, Merket, Stacy, and Freeman, 2004) to provide capabilities to use context to enhance the ability to measure human performance in immersive training environments. We have successfully demonstrated earlier versions of this effort on a multi-platform air warfare simulator involving F/A-18s and E-2Cs as well as in a forward observer simulation-based training environment.

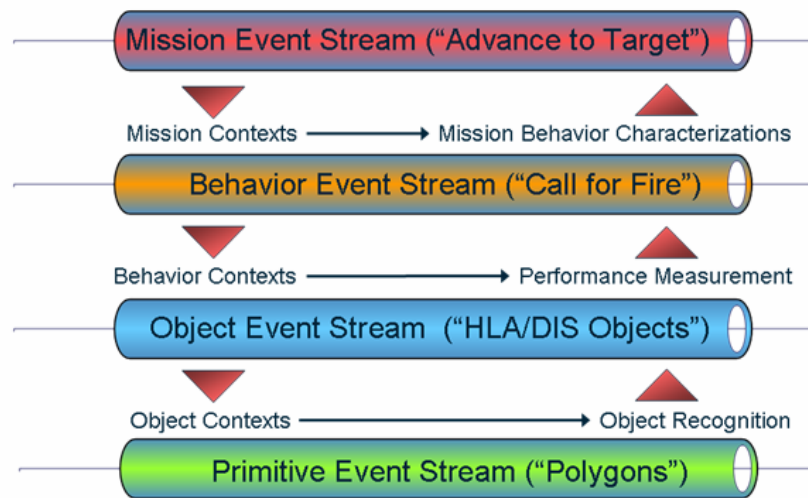


Figure 1. Event stream layers involved in performance measurement.

An Informal Definition of Context

Clearly, measurements of the FO's behavior depend on his context. But what exactly are contexts and how do they relate to measurements? We'll provide a more formal definition later in the paper, but for now, let's say that measurements apply to a stream of trainee or team behavior, and the contexts of that behavior stream include events and other information surrounding it. The behavior stream may be constructed from more primitive event streams, and context may be supplied from higher-level event streams.

Figure 1 shows the situation diagrammatically. In some cases (such as some PC-based games), only primitive events such as polygons and the trainees' current 3D coordinates are available. In others, such as most HLA and DIS-based simulation environments, higher level information about objects in the environment is available. Event attributes at the object layer generally serve as raw data for performance measurement, which is specified at the level of the behavior event stream. In this architecture, higher-level event streams provide context that helps interpret lower-level event streams. Importantly for the present discussion, the mission event stream provides context for interpreting the behavior event stream.

There are two points to note about this informal definition. First, there may be more than one context that applies to a given behavior stream. In the Military Operations in Urban Terrain (MOU) scenario just described, both the fact that the fires

from the building next to the mosque are not having an effect on the Marines below and the fact that there are multiple battalions involved provide separate contexts for the FO's behavior.

Second, context may be either *static* or *dynamic*. A static context is one that will not change during a training mission, and a dynamic context is one that might change. Both are important in the context of performance measurement. In the MOU scenario, the fact that there are multiple battalions involved is an example of static context because it will not change during the mission. The fact that fires from the adjacent building are not currently having an effect on the Marines below is an example of dynamic context, because they might become effective at some point during the mission.

In the next section, we describe two distinct uses for context in a performance measurement system, and illustrate with several examples. We then describe our formal representation of context in HPML. We conclude with a discussion of some of the challenges and opportunities that the formal representation of context provides for human performance measurement in immersive training environments.

TWO USES FOR CONTEXT

In order to understand both of the uses of context, it is necessary to understand the difference between measurements and assessments. We distinguish between *measurements*, which are values on a defined scale that are invariant across contexts, and

assessments, which are interpretations of measurements that may vary with context. For example, the speed of a vehicle, measured in miles per hour, is a measurement. This measurement is relatively independent of driving context. An interpretation of that speed such as “below the speed limit” or “too fast for conditions” is an assessment; these values obviously will change with speed limit context and driving conditions context. Similarly, a score on a standardized test of 653 is a measurement, and an interpretation of that score as being in the 64th percentile is an assessment that depends, among other things, on the context of the group to which the student is being compared.

Parenthetically, assessments need not be judgments or “report cards” for trainees. An important value they bring is to help the instructor and/or after action review leader find starting places for discussion. Measurements with assessments that are either very low or very high are likely to be notable areas of trainee or team performance during the mission, and thus are likely to be among the most important aspects of the mission for trainees and teams to understand.

Context Is a Trigger for Measurements

Context can inform us about the times it is meaningful to take a measurement, and can in fact serve as a trigger for taking that measurement. For example, expert pilot instructors who teach F/A-18 pilots how to perform the notch maneuver—a maneuver that minimizes the aircraft’s visibility on enemy aircraft radar—find it helpful to know the relative altitude and angle of attack of the enemy aircraft when the trainee begins the maneuver. Though at any given point in time when enemy aircraft are present it is possible to take these two measurements, it is impractical to take them all the time in most simulation environments, since a fair amount of interpolation and other computation is often involved. But it is simple and practical to compute these measurements at the single point when context—the beginning of the trainee’s notch maneuver—dictates.

The use of context as a measurement trigger is not limited to automatically calculated measures. The identification of context, the measures themselves, or both, might be observer-based. For example, using

High-Level Architecture (HLA) data to identify the beginning of a notch maneuver or the fact that a MOUT team is about to enter a room is difficult to compute but easy for humans to observe and record. In these cases, the context events might well be provided by observers. On the other hand, it can be difficult for observers to monitor trainee air speed or the percentage of a room covered by a MOUT team’s rifles as they enter it. In these cases, when the automatically collected data is above or below some predetermined threshold, it might trigger observers to, for example, give trainees a score on communications or teamwork

Context Is A Modifier of Assessments

The other main role for context, as hinted above, is to help provide better assessments. In particular, context affects the mapping between measurements and assessments. For example, assessments of measurements of the precision of a landing on an aircraft carrier will depend on things like sea state, weather, and time of day. Assessments of the measured target location error in a FO’s call for fire will depend on whether friendly forces are taking effective enemy fire. Assessments of the efficiency of an Attack Coordinator in prosecuting a target in a Dynamic Targeting Cell in an Air Operations Center will depend on how often they have been interrupted to prosecute higher priority targets. We will discuss additional examples below.

FORMALLY REPRESENTING CONTEXT

We have developed a formal representation of context useful for performance measurement in immersive training environments. In this section we explore the approaches we took to the formalism itself, namely Finite State Machines (FSMs) and then Rules. We embed this representation in an existing XML-schema based performance measurement language, HPML, which already accommodates useful notions such as the distinction between measurements and assessments. To provide background on the medium for the formal representation of context, we next turn to a brief overview of HPML itself. Following that, we discuss the use of an existing standard, RuleML, to representing Rules within HPML.

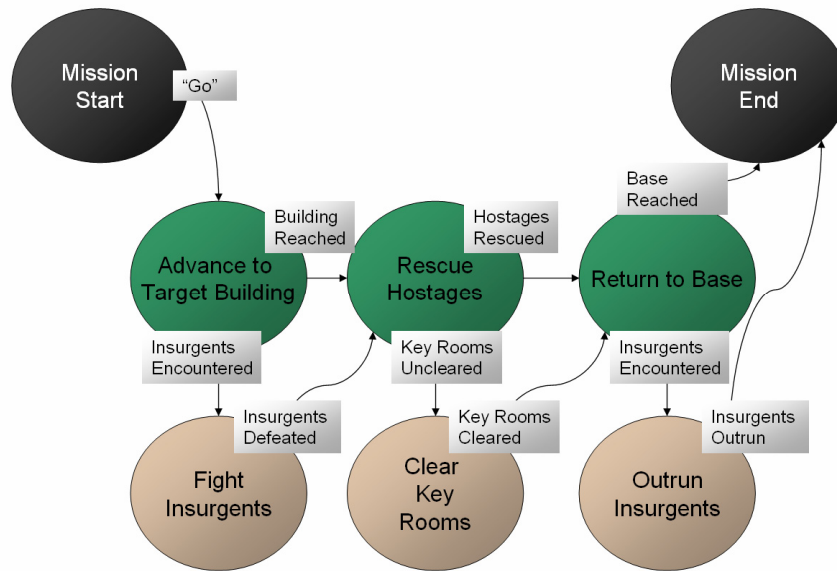


Figure 2. Simplified FSM of a hostage rescue mission.

We conclude this section with a discussion of ways that instructors might specify the use of context in assessments

A First Attempt to Represent Context: Finite-State Machines

A Finite State Machine (FSM) is a simple model of behavior composed of a set of states and the transitions between them. FSMs make the assumption that transitions out of a state follow the same rules no matter how that state was reached. FSMs are simple computation devices that can nevertheless express a great deal of complexity.

An example will clarify how this is done. Suppose we would like to describe progress through the phases of a MOUT mission as a way to model mission context. Figure 2 shows a greatly simplified FSM describing an urban hostage rescue mission. Each circle represents a potential phase of the mission, and the labels on the arrows represent the events that, if encountered, will trigger a transition to a new state. The medium green states in the middle of the diagram represent the phases the mission will go through if no resistance is encountered and the lighter sand-colored states at the bottom are mission phases when resistance is encountered.

The idea of using an FSM as context is that a state-related event such as entering or leaving a state can

serve the two purposes described above, namely to trigger measurements and to provide additional information for assessments.

Table 1. Rules for simple MOUT model, phrased to show equivalence with the FSM in Figure 2.

<p><i>IF</i></p> <p><i>STATE = "Advancing to target building"</i></p> <p><i>AND</i></p> <p><i>Insurgents encountered</i></p> <p><i>THEN</i></p> <p><i>STATE = "Fight insurgents"</i></p>
<p><i>IF</i></p> <p><i>STATE = "Fighting insurgents"</i></p> <p><i>AND</i></p> <p><i>Insurgents defeated</i></p> <p><i>THEN</i></p> <p><i>STATE = "Rescue hostages"</i></p>

It is not hard to imagine the model in Figure 2 doing both—triggering state-specific measurements (How quickly did the team advance to the target when they didn't encounter resistance? What kind of weapons coverage did the team provide when they cleared key rooms?) as well as providing fodder for assessments (Measures of coordination may have a higher tolerance when the team is under attack, and

measures of speed of movement may be interpreted differently when escorting hostages).

As we began to work through common FSMs that would describe training mission context, however, we soon discovered that we needed a context modeling tool that was both simpler and more general than FSMs. On the “simpler” side, many of the FSMs we encountered were either simple alternations between two states (“Feet Wet-Feet Dry”) or were simple unvarying sequences of states (“Find-Fix-Track-Target-Engage-Assess”). We did not find many that were as complex as the simple MOUT mission context model of Figure 2. While these contexts, strictly speaking, can be accurately modeled by FSMs, the machinery of FSMs seems like overkill.

On the “more general” side, we noticed that in constructing context FSMs we often would phrase them as a set of “If-Then” rules. We came to the conclusion that such rules were potentially a more natural way to express context, and, as it turns out, also a more general way to express context.

Rules: A Generalization

The kind of rules we use to represent context have two components, an “If” part consisting of a set of observable or remembered conditions and a “Then” part consisting of a set of conclusions and actions that follow when the “If” part is true. The “If” part is sometimes called the *left-hand side (lhs) or body*, and the “Then” part is sometimes called the *right-hand side (rhs) or head*. One kind of action that the head may specify is to remember a *fact*; such facts may be specified in the conditions in the body.

Rules are usually interpreted by what is called an *inference engine*. Many powerful open source and commercial inference engines—each with its own strengths and weaknesses—are available, including Jess (Friedman-Hill, 2003), JBoss Rules (Proctor, Neale, Lin, & Frandsen, 2006), jDREW (Spencer, 2004), CLIPS/R2 (Production Systems Technologies, 2006), PegaRULES (Pegasystems, Inc., 2006), and a large number of Prolog systems. For expressing context for performance measurement in immersive training environments, this kind of power is not currently required, and in fact would complicate the use of context by performance measurement software. As a result, our efforts have focused on the use of simple rules in very straightforward ways.

In order to understand more about the language in which a Rules-based representation of context is embedded, we now turn to a description of HPML.

Table 2. Rules showing a slight variation on the MOUT scenario in Figure 3 that would be difficult to express as an FSM.

<p><i>IF</i></p> <p><i>Team encounters insurgents</i></p> <p><i>AND</i></p> <p><i>Team is far from base</i></p> <p><i>THEN</i></p> <p><i>Team CONTEXT contains “Fighting insurgents”</i></p> <p><i>Team fights insurgents</i></p>
<p><i>IF</i></p> <p><i>Team has not rescued hostages</i></p> <p><i>AND</i></p> <p><i>Team is not at target building</i></p> <p><i>THEN</i></p> <p><i>Team CONTEXT contains “Advancing to target building”</i></p> <p><i>Team advances to target building</i></p>

HPML

HPML is an XML-Schema-based language that is intended to cover all meaningful aspects of human performance measurement in training in immersive environments. The intent is to make HPML freely available to the simulation training community and to work towards its standardization. It is still early in HPML’s life cycle, though, and it will be undergoing more change than would be comfortable for some applications. For now, therefore, contact the first author for further information on obtaining the HPML schema and associated documentation at no charge.

The hierarchy in HPML (Figure 3) meets the specific requirements of representing both generic concepts (e.g., measurements and assessments) and mission-specific concepts (e.g., instances of measurements and instances of assessments). By making these distinctions, HPML is able both to describe available resources and to express the tailoring of those resources for a given training mission. And of course, HPML has been extended to represent both context and instances of context, as described below.

Benefits

HPML may be used for local communication within the performance measurement system as well as the federates' communication network, typically HLA or Distributed Interactive Simulation (DIS). This provides several benefits:

- HPML provides a rich, flexible basis for the clear, exact expression of both observer-based and automatically computed new measures.
- HPML provides a foundation for sophisticated mission measurement configuration by instructor/operators and automated agents.
- HPML provides a human-comprehensible framework for automated portions of the performance measurement system to communicate about a wide spectrum of measures that is maintainable, and easily and gracefully extensible.

The top-level element in HPML is in fact an HPML element. The HPML element consists of nine sets of entities, as follows: (note: HPML names are shown in **Courier New** font):

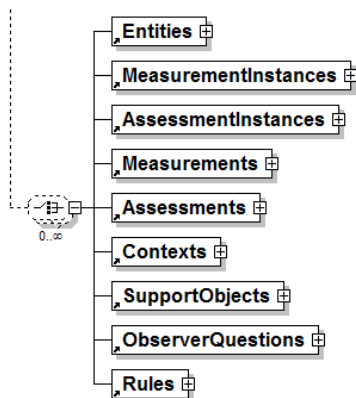


Figure 3. HPML Hierarchy of Elements.

- **Entities.** Objects from the training world: trainees, teams, instructors, observers, tasks, assignments, and so on. These are all mission-specific entities; they will be different from mission to mission if the trainees, teams, or assignments differ, for instance.
- **MeasurementInstances.** Measurements to be taken with respect to specific entities for specific missions. They are generally based on `Measurements` that have already been defined.

- **AssessmentInstances.** Assessments to be performed on specific measurement instances with respect to specific trainees, teams, assignments, performance standards, and mission contexts. They will generally be based on `Assessments` in the library.
- **Contexts.** To be described in the next section.
- **Measurements.** Templates for automatically measuring trainee or team performance from data available over the simulation network, generally available in a library. Data sources and computations are specified. May be parameterized.
- **Assessments.** Templates for assessing measurements of trainee or team performance, generally available in a library. Inputs—one or more measurements or assessments—are specified, as are performance standards and mission context.
- **SupportObjects.** Objects that support measurements and assessment, such as scales and predefined data sources
- **ObserverQuestions.** Specification of questions to appear on observer-based measurement recording devices.
- **Rules.** To be described in the next section.

The initial elements, namely `Entities`, `MeasurementInstances`, `AssessmentInstances`, and `Contexts`, are specific to a given mission. That is, each mission will specify its own set of entities and specific instances of measurements and assessments of them.

`SupportObjects` comprise utility elements such as measurement scales and the details of the configuration of data sources. The remaining elements, `Measurements`, `Assessments`, `ObserverQuestions`, and `Rules`, work across multiple missions.

Using RuleML

Rather than inventing another representation of `Rules` for HPML, we decided to lean on existing standards. One standard that is beginning to be adopted in a variety of places is RuleML (Boley, 2002). RuleML is in fact a set of modular descriptions of various rule features, bundled in various ways into a family of sublanguages. If this sounds complex, that's because it is.

Our task was to navigate these modules and hierarchies to find a suitable language definition to incorporate into HPML. We chose one of the simplest possible sublanguages, namely Binary Datalog. We imported this sublanguage into HPML using its own namespace to avoid name collisions with the rest of HPML.

A Context element is now defined to be a set of facts that may have an initial value. The processing of Rules may add or subtract facts from this set. If this set never changes during the mission, the element expresses a static context; if the rules cause the set to change, the Context element is dynamic.

Contexts apply separately to each trainee or team. One team may be approaching their target building to rescue hostages; another team during the same mission may already have reached their target building and rescued their hostages. Returning to Table 2 above, occurrences of the term “team” in the Rules are replaced by the identifier of an actual team participating in the mission; if the left-hand side of the rule applies to the team, then the actions in the right-hand side are taken on behalf of the team. A similar logic applies to rules involving trainees instead of teams.

Table 3. A Rule from Table 2 Expressed as RuleML/HPML Fragment.

```
<body>
  <And>
    <Atom>
      <Var>Team</Var>
      <Rel>Encounters</Rel>
      <Ind>Insurgents</Ind>
    </Atom>
    <Atom>
      <Var>Team</Var>
      <Rel>Location</Rel>
      <Ind>FarFromBase</Ind>
    </Atom>
  </And>
</body>
<head>
  <Atom>
    <Ind>CONTEXT</Ind>
    <Rel>Contains</Rel>
    <Var>Team</Var>
    <Ind>FightingInsurgents</Ind>
  </Atom>
</head>
```

Examples of Context in HPML

We now present a few examples of the use of context in HPML. Table 3 shows the first Rule from Table 2 expressed in HPML.

Using context in the two ways mentioned above is relatively straightforward. To express a trigger on a measurement (actually, on a MeasurementInstance), simply add a MeasurementTrigger element, as in the fragment shown in Table 4.

Table 4. HPML Fragment Showing a Measurement Trigger.

```
<MeasurementInstance
  InstanceOf="TeamSynchronization"
  ID="TS1"
  SubjectRef="MOUTTeam">
  <MeasurementTrigger
    ContextRef="MOUTTeamContext"
    TriggerValueRef="AdvanceToTrgt"/>
</MeasurementInstance>
```

Table 5. HPML Fragments Showing the Use of Context in an Assessment Instance.

```
<AssessmentInstance
  InstanceOf="TeamSyncAssessment"
  ContextRef="MOUTTeamContext">
  <SubjectRef Ref="MOUTTeam"/>
</AssessmentInstance>
<Assessment ID="TeamSyncAssessment"
  MeasurementRef="TeamSynchronization">
  <AssessmentCase
    MeasurementValue="2"
    PositiveTolerance="0.5"
    NegativeTolerance="0.5"
    ContextCondition="Raining"
    AssessmentValue="Expert"/>
  <AssessmentCase
    MeasurementValue="2"
    PositiveTolerance="0.2"
    NegativeTolerance="0.2"
    ContextCondition="Sunny"
    AssessmentValue="Expert"/>
</Assessment>
```

To use context as a modifier for assessments, the AssessmentInstance needs to know which context to use, as in the HPML fragment shown in Table 5. The Assessment Case elements for the Team Synchronization Assessment show a partial list of cases that provide information on interpreting the referenced measurement. Context conditions help choose which case applies—in this case, there is a tighter timing tolerance on measurements labeled

'Expert' when it is sunny as opposed to when it is raining.

A User Interface for Assessment Context

Specifying exactly what effect context should have on an Assessment Instance turns out to be difficult for professionals not familiar with XML and rule-based systems. This is a problem, because we want to make the specification of context and its uses accessible to instructor/operators and other personnel who will be setting up training missions. We are therefore in process of developing an easy-to-use graphical user interface for this purpose, and, because it is potentially the most difficult to understand, have started with the specification of context for use in assessment.

Figure 4 shows a version of this interface based on our work with performance measurement of forward observers. In the top left portion of the window, the user has chosen to use a Pass/Fail criterion, though they could have specified a set of user-defined categories. In the top right portion of the window, the user is specifying a rule with a tolerance, and can choose specific context values for which that rule applies. In the bottom of the window is a summary of all the rules that might apply to this assessment, together with an opportunity to edit or delete any of those rules.

CHALLENGES USING THIS APPROACH

There are several remaining challenges to integrating a representation of context into an automated performance measurement system for immersive training environments.

User Specification of Context

An important challenge is to find a way to make the specification of context available to users such as instructor/operators whose areas of expertise typically do not include rule-based knowledge representation, human performance measurement, or software engineering. As just discussed, Figure 4 shows how a user interface might help gather the required information, especially as it is put into use in reporting assessments. This is only a first step towards solving the general problem of helping users specify the rules that define context, though, especially as the rules begin to require sophisticated features to develop more accurate representations of context.

Our plan of attack has several facets. First, we intend to develop a library of reusable context rules and rule modules that have been parameterized to encourage adaptation to new situations. Our belief is that it is often easier to recognize and configure existing rule-based specifications than it is to develop new ones.

Figure 4. Graphical User Interface for Specification of Context in Assessments.

from scratch. Such a library can be populated as users develop from-scratch solutions, and, on a slower cycle, can also be populated offline by personnel with more expertise in rule-based systems. Second, as the artifacts of training mission planning become electronic and standardized, we will investigate ways to translate them automatically into a rule-based representation of context. Finally, we believe there are simplifications of the authoring tasks that would allow instructor/operators to easily construct many common contextual representations

Efficient Computation of Current Context

Naïve execution strategies for rule-based systems of any size are remarkably slow; for every cycle of the inference engine, all the conditions on the left hand side of every rule need to be checked against available facts to determine which rules apply. The most widely accepted solution to this is an optimization called the *Rete algorithm* (Forgy, 1982.)

The Rete algorithm basically organizes the conditions on the left-hand side of all rules so that the fastest sequence of checks is performed, and so that redundancy in checks is eliminated. The algorithm shows remarkable improvements in efficiency, especially when, as in many rule-based systems, the set of facts against which rules are checked changes slowly.

When using rules to specify context in immersive training environments, there are two sources of facts: 1) the right-hand side of rules that apply; and 2) attributes of events from the immersive training environment. The second of these is unusual for a rule-based system, and may well reduce the likelihood that the facts will change slowly. In addition, some of the conditions specified in the rules may go beyond simple event attributes—they may require computation on one or more of those attributes from one or more events.

We have encountered few performance problems so far in representing context in real immersive training environments; but our context specifications so far have only involved a small number of rules. We will keep a watchful eye on system responsiveness issues as the number of rules scales up, and will investigate implementing suitable variants of the Rete algorithm as necessary.

CONCLUSION

We are now in a position to use rules to describe the context of the opening scenario where a Marine FO was trying to determine the best course of action when receiving fires from a building adjacent to a mosque. Two of the rules involved in that description are shown in Table 6. The rules in Table 6 can be transformed to HPML in a straightforward way. We can use this context to assess the FO trainee's performance (for instance, there is urgency in an Immediate Suppression mission, so speed becomes an important factor) and to trigger other measurements of the FO trainee (for instance, when speed is important, FOs shouldn't spend too much time generating the call for fire).

Table 6. Rules for the FO's Context in the Opening Scenario.

<p><i>IF</i></p> <p><i>Fires from building are causing casualties</i></p> <p><i>THEN</i></p> <p><i>FO CONTEXT contains "Immediate Suppression"</i></p> <p><i>Call for Immediate Suppression Mission</i></p>
<p><i>IF</i></p> <p><i>Fires from building are NOT causing casualties</i></p> <p><i>AND</i></p> <p><i>Marines below can take cover</i></p> <p><i>THEN</i></p> <p><i>FO CONTEXT contains "Company Commander's Decision"</i></p> <p><i>Consult Company Commander</i></p>

Our experience using the representation of context this way in real-world training testbeds involving F/A-18 pilot trainees and involving FO trainees leads us to believe that the techniques described in this paper have wide applicability to human performance measurement in immersive training environments. It will have the effect of making the measurements and assessments more meaningful. This, in turn, will contribute to more effective feedback to the trainee, directly or indirectly through an after action review leader, and will result in more efficient and effective training programs.

ACKNOWLEDGEMENTS

The Navy Program Manager for Aviation 205 and the Office of Naval Research funded this work. It is managed by NAVAIR Training Systems Division, Orlando, FL. The opinions expressed here are those of the authors and do not necessarily reflect the views of the sponsors or the Department of Defense.

REFERENCES

- Boley, H. (2002). *The Rule Markup Initiative*. <http://www.ruleml.org/>.
- Forgy, C. L. (1982). "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", *Artificial Intelligence*, 19, 17-37.
- Friedman-Hill, E. (2003). *Jess In Action: Java Rule-Based Systems*. Manning Publications.
- Lackey, S., Merket, D., Stacy, E. W., & Freeman, J. (2004). Intelligent Training Support Tools: Technology for the Future. *Proceedings of the 2004 American Institute of Aeronautics and Astronautics Modeling and Simulation Technologies Conference*, 210-217.
- Pegasystems, Inc. (2006). *PEGARules*. <http://www.pega.com/Products/RulesTechnology.asp>
- Proctor, M., Neale, M. Lin, P., & Frandsen, M. *Drools Documentation*. Retrieved June 8, 2006 from http://labs.jboss.com/file-access/default/members/jbossrules/freezone/docs/3.0.1/html_single/index.html
- Production Systems Technologies (2006). *CLIPS/R2*. <http://www.pst.com/clpbro.htm>.
- Spencer, B. (2004). *jDREW: A Java Deductive Reasoning Engine for the Web*. <http://www.jdrew.org/jDREWebsite/jDREW.html>.
- Stacy, E. W., Freeman, J., Lackey, S., & Merket, D. (2004). Enhancing Simulation-Based Training with Performance Measurement Objects. *Proceedings of the 2004 Interservice/Industry Training, Simulation & Education Conference*, Orlando, FL, 250-258.
- Stacy, E. W., Merket, D., Freeman, J., Wiese, E., & Jackson, C. (2005). A Language for Rapidly Creating Performance Measures in Simulators. *Proceedings of the 2005 Interservice/Industry Training, Simulation & Education Conference*, Orlando, FL, 207-218.