

**FACET: A SIMULATION SOFTWARE FRAMEWORK  
FOR MODELING COMPLEX SOCIETAL PROCESSES AND INTERACTIONS**

John H. Christiansen  
Decision and Information Sciences Division  
Argonne National Laboratory  
Argonne, Illinois 60439-4832  
e-mail: [jhc@anl.gov](mailto:jhc@anl.gov)

**RECEIVED**  
**JUL 10 2000**  
**OSTI**

This submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

**KEYWORDS:** object oriented, agent based, decision support systems, resource management, health care

#### **ABSTRACT**

FACET, the Framework for Addressing Cooperative Extended Transactions, was developed at Argonne National Laboratory to address the need for a simulation software architecture in the style of an "agent-based" approach, but with sufficient robustness, expressiveness, and flexibility to be able to deal with the levels of complexity seen in real-world social situations.

FACET is an object-oriented software framework for building models of complex, cooperative behaviors of agents. It can be used to implement simulation models of societal processes such as the complex interplay of participating individuals and organizations engaged in multiple concurrent transactions in pursuit of their various goals. These transactions can be patterned on, for example, clinical guidelines and procedures, business practices, government and corporate policies, etc. FACET can also address other complex behaviors such as biological life cycles or manufacturing processes.

To date, for example, FACET has been applied to such areas as land management, health care delivery, avian social behavior, and interactions between natural and social processes in ancient Mesopotamia.

#### **INTRODUCTION**

For most of the past decade, the Decision and Information Sciences Division (DIS) at Argonne National Laboratory (ANL) has had an on-going effort to develop robust object-based software architectures to address complex, multidisciplinary modeling and simulation problems. Most of our earlier object modeling work (ca. 1990-1995) focused on interactions among various natural

processes (e.g., weather, hydrology, etc.). As the scope and complexity of our simulated domains grew to encompass human societal processes, we saw value in building a unifying software infrastructure to support construction and interoperation of societal process models, as well as interoperation of these societal process models with natural process models, within the same simulation frame of reference. The Framework for Addressing Cooperative Extended Transactions (FACET) is a product of this effort.

Certain general aspects of the behavior patterns embodied in societal processes tend to distinguish them from natural, or physics-based processes. Some of the hallmarks of societal processes are as follows.

- Processes involve some level of cooperation among the agents involved (though coercive behavior patterns are also possible).
- Agents may be concurrently involved in several behavior patterns and may need to prioritize their interactions and commitments to participate in these patterns.
- Behavior patterns can branch to follow several alternative paths and may segue into other patterns, based on social context.
- Processes can be interrupted and resumed at a later time.
- Processes can be preempted in progress, or be scheduled to occur but then be cancelled before they begin.

The FACET software infrastructure is designed to explicitly support modeling representation of these distinguishing characteristics. As a result of this design approach, FACET models can be more closely mapped to the often chaotic realities of social interchange, with less reliance on artificial abstractions and simplifications imposed by the modeling framework.

## **DISCLAIMER**

**This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.**

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

Fundamentally, FACET is a set of software objects that facilitate implementing and running models of complex, cooperative behaviors of agents. Here, the term, "agent," is used to denote simulation domain entities capable of autonomous, cooperative, and adaptive behavior. These agents are typically represented in a simulation by "Person" or "Organization" objects. FACET societal process models can be patterned on, for example, business practices, social mores and traditions, clinical practice guidelines and procedures, governmental and corporate policies, military doctrine, and so on. FACET can also be used to address complex behaviors not associated with human societal processes, such as zoological and botanical life cycles and manufacturing processes.

## FACET SOFTWARE ARCHITECTURE

FACET models of complex, cooperative behaviors are run within a process-based discrete event simulation framework. At ANL, we have used the Dynamic Information Architecture System, or DIAS (Christiansen 2000), a distributed, object-based simulation framework, as the overarching discrete event simulation framework for FACET models.

Within the DIAS framework, new or legacy simulation models are provided with software object "wrappers" that formally define the model's capabilities and limitations, including specification of which abstractly defined behaviors of which classes of domain object they are qualified to implement. The set of models employed to implement domain object behaviors is then selected and linked to the objects whose behaviors they represent at run time, based on user-supplied information defining the context of the simulation. Although DIAS offers some strong advantages as an operating environment for FACET models, it is intended that FACET models be able to operate within other suitable discrete event modeling frameworks if desired.

FACET models can be run on workstations or personal computers under either UNIX or Windows NT. FACET was originally prototyped and built in the Smalltalk object-based language. A port of FACET from Smalltalk to Java is underway and should be completed by the end of summer 2000.

The FACET architecture relies on two main high-level software components, the "Course of Action" (COA) object class suite and the "Participant" object class. The suite of COA objects provides the infrastructure for building FACET behavior pattern models. Participant objects allow domain objects to take part as agents in FACET-based behavior patterns. Simulation domain objects of essentially any object class can participate as agents in FACET model behavior patterns, by associating them with FACET Participant objects.

## Course of Action Object Class Suite

The COA object class suite defines a "collaborative extended transaction" model. A COA model is built as a network of individual steps, represented by "Step Template" objects, each of which is essentially a sub-model in its own right. Each step is roughly analogous to a scene in a play, with its own formally specified cast of characters (agents), props (agents' resources), and locale. Each step:

- Requires resources held by agents participating in the action;
- Consumes a specified (though variable) interval of time, during which the agents and resources are committed to the step and thus unavailable for other activities; and
- Results in changes to the state attributes of one or more of the participants, and/or changes to state attributes of other domain objects that had been declared to be of interest to the COA.

Although COAs are initially invoked by an agent as a response to a simulated situation, they are not necessarily "owned" by that agent. They are templates that provide a pattern for a number of different agents to collaborate to execute a complex dynamic behavior. For example, in a FACET health care simulation, a receptionist at an outpatient clinic, on receiving an event that indicates that a new patient has arrived in the waiting room with a cardiology appointment, may invoke a COA model that characterizes the detailed procedural flow for such an appointment. In this example, although the receptionist may launch the appointment COA model, the model actually represents a behavior of an *organization* - the clinic, or perhaps a particular medical department at the clinic - rather than a behavior of the receptionist. Several agents (doctors, nurses, technicians, receptionist, etc.) participate in the COA's behavior pattern on behalf of their organization. The COA is actually initiated by the organization when the receptionist sends a message that cues the organization to commence the appropriate behavior pattern. Once the behavior pattern is initiated, various agents representing doctors, nurses, etc. are cued to interact with each other and the patient on behalf of their organization.

The key object classes that make up the FACET COA object class suite, the COA Template, COA, and Step Template, are described below.

"Course of Action Template" Object Class (or "COA Template," see Figure 1) objects carry a directed graph of individual steps (Step Template objects). The Step Template objects specify the agents and other resources needed for each step and carry the instructions for what exactly is to be performed in the step. Generally, only one instance of each COA

Template object class exists in the simulation frame of reference at any one time. Each unique behavior pattern, or FACET COA model, is a single instance of a subclass of the COA Template object class.

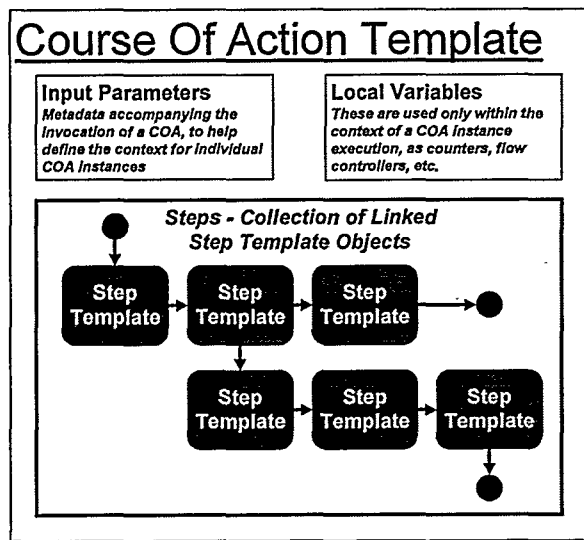


Figure 1. FACET Course of Action Template Object

The "Course of Action" Object Class (or "COA," see Figure 2) is associated with individual active instances of a behavior pattern, or COA Template.

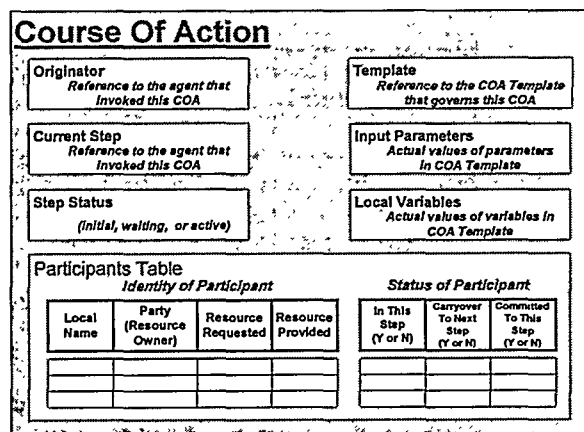


Figure 2. FACET Course Of Action Object

In contrast to the COA Template object class, there is only one *class* of COA object, but potentially a great number of *instances* of that class. Each COA object instance is, in essence, a "bookmark," indicating the current state of a *specific* extended transaction among simulation agents that is being modeled in accordance with a COA Template.

The centerpiece of the COA object class layout is the "Participants Table." This table correlates the

generic names of actors and other resources declared in the Step Template objects' detailed step action layouts to specific resources requested from and provided by various agents in the simulation. Participants are added to and removed from the table as the COA proceeds through its steps. The Participants Table also indicates whether a resource is needed in the current step, whether it carries over from the previous step, and whether the resource has committed itself (or has been committed by its controlling agent) to participate in the step. Note that a domain object can be both an agent and a resource of another agent at the same time, as in the case of an organization (an agent) that employs individual persons. These persons are agents in their own right but can serve as resources within the employer organization's COAs.

Figure 3 provides a health care example of a COA object for a behavior pattern in progress. Here, the behavior pattern is a primary care routine return visit, and the pattern is performed on behalf of a family practice department at a hospital's medical clinic. As Figure 3 shows, the Participants Tables for steps in a COA are flexible. They can require either specific instances of a resource (e.g., a specific doctor at a clinic) or only a class of resource (e.g., "a nurse," or "an examination room").

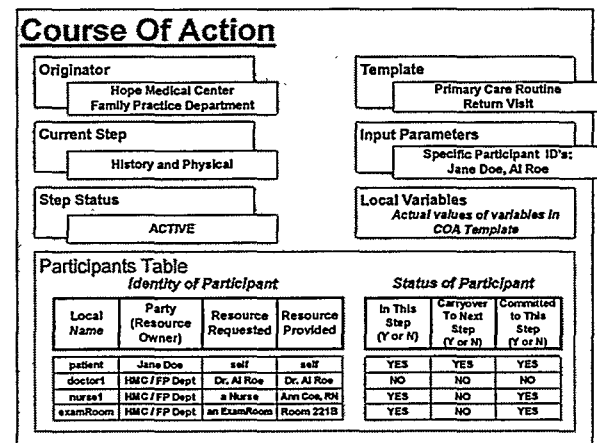


Figure 3. FACET Course of Action Object for a Health Care Behavior Pattern in Progress

The COA in Figure 3 shows a step in progress: a patient and a nurse are in an examining room cooperating in a "history and physical" step, in which the patient describes her symptoms, etc., and the nurse records this verbal information along with the patient's temperature, blood pressure, and possibly other measurements and observations. The doctor is not needed in this step, but is carried in the Participants Table because he is needed later in the office visit, for diagnosis, treatment, and/or referral steps.

A major simulation may have a very large number of agents or *ad hoc* groupings of agents embarked, in parallel, on the "same" COA. To continue with an example in the health care domain, thousands of

individuals may be under the care of a health care organization for diabetes at any given time, and therefore nominally embarked on the same COA (although different individuals would be involved in different *steps* in the COA, both as patients and as health care providers). Figure 4 shows three active instances (COA's) of the same behavior pattern (COA Template) underway at a clinic. Note that the patterns happen to have progressed to different stages of completion (different steps). Also note that the participants in the different patterns have some overlap; the doctor, in particular, must have a means of resolving multiple concurrent demands for his/her time. This mechanism is provided by the "Participant" object class, discussed later in this paper.

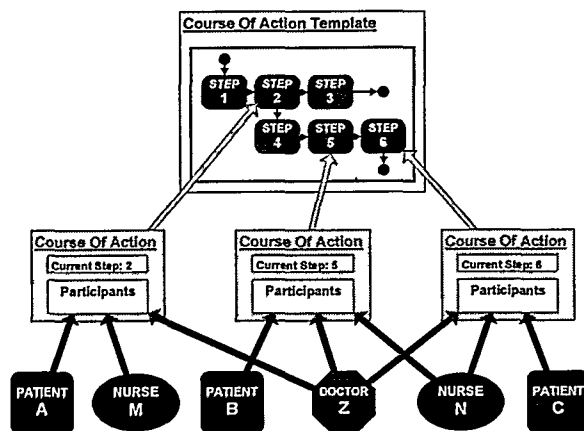


Figure 4. Simple Example of Relationship Between FACET COA Template Objects and COA Objects

The "Step Template" Object Class (Figure 5) carries the actual instructions to be carried out in a step of a COA. The Step Template also declares its own resource requirements. Each step can exist in one of three states:

- *Initial*: the previous step (if any) is over, and participants are being invited to participate in the current step.
- *Waiting*: all participants have been invited, but not all have committed to participate.
- *Active*: the step has begun, and has posted its own completion event for a future time, but has not yet run to completion.

The "Perform Step" code block does the actual work of a step, changing state variables of participating agents or other domain objects, and posting events and messages to other domain objects as needed. The "Decide Next Step" code block may have conditional logic for branching to alternate steps, based on values of participants' attributes, attributes of objects referenced in

COA input parameters, and/or values of local variables relevant only to this instance of the COA. (The latter variables are held in a dictionary data structure in the COA object.)

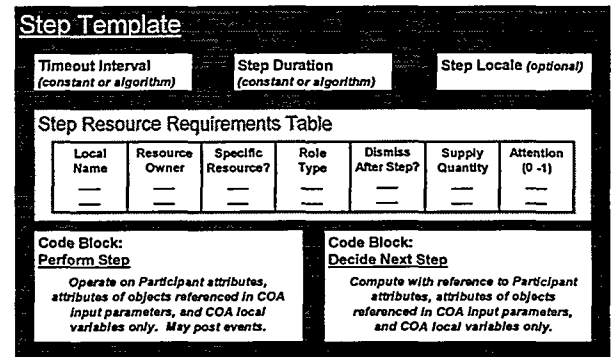


Figure 5. FACET Step Template Object

In Figure 5, the resource requirements table is used to drive the composition and generation of requests for participation in the step, and to update the COA's Participants Table. The "supply quantity" column addresses resource requests for consumables (fuel, food, etc.) as opposed to assets (staff, equipment, etc.). The "attention" column addresses the rather subtle problem of tasking agents in cases such as shift work, in which the task may not require the undivided attention of the agent for the full duration of the task. Via this FACET facility, it is possible to assign aggregate subtasks that are below the temporal resolution of the simulation. For example, a nurse in a hospital ward can be assigned a task such as making rounds to check patients' medications, for a full 8-hour shift but at, say, 0.15 attention. The practical constraint enforced by FACET is then that the nurse's tasks assigned for the shift may not exceed a total attention level of 1.0. In general, most agent participation requests tend to be assigned at an attention of 1.0. At the opposite extreme, however, patients may participate in some steps, such as wearing a portable cardiac monitor for a day, at an attention level of zero – the patient is definitely required for the step, but could in principle be doing something else concurrently as well.

The overall sequence of processing for a single step in a FACET COA can be characterized as follows. All processing described here is asynchronous and non-blocking.

- Enter step for the first time. Step is now in *initial* state.
- Step examines Step Template participants' requirements and posts invitations to all Step participants. Step is now in *waiting* state.
- Step waits for responses to invitations. If sufficient commitment of resources is not forthcoming before

the step timeout interval has expired, exception processing is performed (see "Exception" steps below).

- If sufficient response is received from the invited participants, the step can proceed. An algorithm specified in the Step Template is run to determine the step duration. A "Step End" event is then posted to the event queue for the computed step ending time. Step is now in "active" state - participants are now "busy" performing the action in the step.
- When the COA receives the Step End event, the current Step Template fires its "Perform Step" code block. As a result, participants' state attributes may be altered, other domain objects' state attributes may have been changed, participants' supply resources may be depleted or replenished, and events may be sent to other appropriate domain entities.
- The Step Template now fires its "Decide Next Step" code block, to determine which step will be invoked once the current step finishes.
- Participants are now released from the step; messages are sent to confirm this. The step then ends, and the next step begins.

The Step Template object class is subclassed to accommodate three somewhat different types of step: Actions, Expansions, and Exceptions. Most of the preceding discussion has focused on Actions, for simplicity, but the three types are handled quite similarly.

Actions are a type of step that has finite duration (in simulation time) and ties up assets and may consume supplies from among the agents participating in the Action.

Expansions are a type of step that implicitly calls for execution of another entire COA (thus "expands" a step into a COA). The sub-COA referenced in this Step then executes to completion and returns to the same Expansion step when done. Expansion steps facilitate the use of nested COAs, which is a very helpful capability and a key to ease of re-use of COA components

Exceptions are the key to the ability of FACET COA models to respond realistically to unusual circumstances. An Exception step is executed in place of the normal "current" step in a COA if a specified type of event is received by the COA object. Step timeout messages can also trigger an Exception step. An Exception step by definition has no participants, and takes no elapsed simulation time to perform, but carries the logic needed to route the COA onto a different sequence of steps, or to end the COA entirely. This allows COA's to preempt normal procedures, apply coping mechanisms or crisis management measures, etc., in response to changing conditions. FACET also

supports the introduction of variability via randomly variable step durations and step timeout intervals, as well as random draws within a step's "Perform Step" and "Decide Next Step" logic blocks.

## The "Participant" Object Class

The "Participant" object class provides the software infrastructure that allows a domain object to take part as an agent in FACET COA model behavior patterns. A Participant object is assigned as a delegate object to each simulation object for which COA model participation is desired. During the simulated duration of a COA step, the Participant object for each agent is cognizant of the existence, as well as the nature and extent, of the agent's participation, so that the resources (including the agent's personal attention, where appropriate) committed to the Action cannot be otherwise employed until the step terminates. Participant objects include the following key components:

- *Agenda Object*: maintains an up-to-date record of all unsatisfied requests for this agent's resources, and all of the agent's currently active and pending behavior patterns and resource commitments.
- *Resource Manager Object*: a mechanism to keep track of all commitments of, and requests for, resources held by the agent.
- *Role Objects*: a mechanism to allow agents to respond differently to requests depending on their current stance, or role, in the simulation. For example, a person who happens to be a doctor, a father, and a golfer might take on each of these roles at different times, but the resource allocation behaviors would likely be different for each role.

## Agent Participation Negotiations

The FACET COA and Participant objects provide a highly flexible and expressive means for representing diverse transactions. The COA's essentially negotiate with the participating agents for their cooperation. The agendas of all potential participants are kept fully updated on the status of all *other* participants' commitment to a step, and each participant is free (subject to the constraints inherent in its own behavioral models) to join a step, refuse to join, defer joining pending other developments, join a different step on its agenda instead, designate specific resources from its resource pools to meet a general resource request, etc. For example, assume that a particular health care COA step requires a patient, examining room, nurse, and technician. A nurse might be assigned by his/her department to participate in the step, but if the technician fails to appear within a certain maximum wait time, the nurse can withdraw and await a different task or join a different step.

## FACET APPLICATION EXAMPLES

At ANL, FACET has been applied to a diverse collection of application areas. Three such FACET applications are described here to illustrate the flexibility and range of applicability of the FACET architecture.

### Sustainability of Ancient Mesopotamian Urban Centers under Environmental Stress

ANL is collaborating with the University of Chicago's Oriental Institute to build and exercise a dynamic object model of ancient Mesopotamian urban / agrarian life, using our DIAS and FACET object-based modeling frameworks. We are seeking a better understanding of the dynamics of ancient Mesopotamian urban centers – in particular their sustainability, growth, or decline in the face of environmental stress.

The initial simulation prototype represents a nucleated community of some 5,000 – 10,000 people, and its agricultural hinterlands, in northern Mesopotamia, circa 2,500 B.C. The actual historical prototype is the archaeological site now known as Tell Al-Hawa, near Mosul in northern Iraq (Ball et al. 1989). Simulations will address natural (weather, crop growth, hydrology, population dynamics, etc.) and societal (farming practices, kinship-driven behaviors, etc.) processes interweaving on a daily basis across 200-year simulations. This work is a substantial extension and expansion of earlier sustainability modeling work at the Oriental Institute (Wilkinson 1994). A key outcome of the FACET simulation is expected to be a new level of insight into the sensitivity of such a community to imposed environmental stresses such as droughts and soil erosion, in terms of ability to successfully handle such problems as labor shortages at harvest. Dynamic behaviors of software objects representing households, fields, crops, herds, etc. are simulated using (1) proven, existing natural systems models such as the U.S. Department of Agriculture's EPIC model for field crop dynamics and National Oceanographic and Atmospheric Administration paleoclimate models and (2) new FACET agent-based societal models based on archaeological and anthropological evidence.

Figure 6 depicts a simple subset of the COAs governing field crop management for individual family farming efforts in ancient northern Mesopotamia. In the step template flows shown, some steps have the option of repeating themselves. This reflects the notion that, once begun, those tasks tend to be addressed each day (or perhaps less often, for crop maintenance tasks) by appropriate teams of workers until they are completed. The processes illustrated here are undertaken each year by each of the several

hundred to a few thousand extended families included in the Tell Al-Hawa urban area's simulation domain. Not included in the figure are the COA's that regulate the patronage relations between extended families and their lineage chiefs, nor between the lineage chiefs and the city's palace and temple households. Also not included here are the various economic behavior patterns, such as the undertaking of loans of grain, with repayment often in kind or in labor, as well as the pastoral component of the community and its continual interchange with the agricultural component.

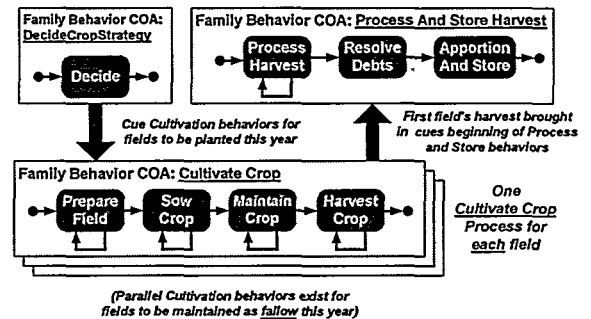


Figure 6. Course of Action Templates for Simplified Field Crop Management Behavior Patterns for a Single Ancient Mesopotamian Extended Family

### Avian Social Dynamics in the Context of Integrated Land Management at U.S. Army Installations

ANL is currently completing a project for the U.S. Army Corps of Engineers Construction Engineering Research Laboratory aimed at exploring the social behaviors and population dynamics of an endangered bird species, the red-cockaded woodpecker (RCW), within the context of training activities and integrated land management initiatives at U.S. Army training facilities (Dolph et al. 2000). Using the FACET and DIAS frameworks, we have constructed an object-oriented individual-based, spatially explicit population dynamics model for the red-cockaded woodpecker, based on a model described by Letcher et al. (1998). This model can then be used to predict RCW populations dynamically in space and time, given that various land use and land management influences are acting within the ecosystem. FACET has been used to model the behavior patterns of the land management activities (planting, burning, etc.) that are included in the environmental stimuli impinging upon RCW populations.

For use in the RCW simulations, FACET models represent social behaviors of RCWs, such as the cooperative behaviors of a breeding pair. The FACET "agents" are in this case individual birds. The resource that each bird provides is simply its commitment to carry out the prescribed behaviors for the specified times and time intervals. RCW social dynamics, as prescribed in the Letcher et al. model, unfold on a



seasonal basis. For each season, the following dynamics are addressed:

- Reproduction (1<sup>st</sup> season of each year only),
- Mortality,
- Movement (dispersal), and
- Competition (basically, males compete for territory, females for mates).

The FACET RCW model implementation uses regularly scheduled discrete events to move the simulated RCW population state forward in uniform seasonal time steps. Other dynamic processes (e.g., weather, hydrology, land management practices) can be running concurrently with the RCW model in the same simulation, allowing for inter-process feedbacks. These other processes can be running with time steps different than those of the RCW model; some may be driven by irregularly spaced events rather than a uniform time step.

### Health Care Clinical and Logistical Simulation

ANL has used both the FACET and DIAS software architectures to assemble HealthSim, a flexible software framework for addressing clinical and logistical aspects of health care delivery over a broad range of simulation level of detail and fidelity. HealthSim has been exercised in extensive applications for a large health care organization in California.

HealthSim uses scores of simulation models running in the same simulation, with many thousands of agents, to address the interplay of diverse dynamic behaviors, such as:

- Human physiological processes, including onset and progression of diseases, development of signs and symptoms, effects of interventions, etc. These are typically modeled using mathematical (e.g., differential equation integrator) biological models.
- Human cognitive behaviors, such as response to symptoms, making and keeping appointments, etc. FACET models are used here.
- Clinical / logistical processes, such as clinical practice guidelines, office procedures, personnel policies, etc. FACET has been used to represent all such processes.
- Medical monitoring equipment response. These are typically mathematical, engineering models.

In HealthSim, the FACET agent objects include organizations, such as medical departments, and individual persons in the occasionally overlapping roles of health care provider and patient. Individual "Person" objects include embedded "Physiology"

objects with nested physiological subsystem objects. Models of normal physiological processes and pathophysiology are run as behaviors of these physiology objects. The Person objects experience symptoms by virtue of sign / symptom events they receive, posted by their own physiological objects in response to physiological changes.

Two brief examples illustrate the range of applicability of the HealthSim framework:

1. Access to Care. Multi-year simulations were performed for the entire 300,000 member population for a large medical center in Los Angeles, with over 400 doctors and other primary care providers. More than one million simulated health care service requests were processed through the system per simulated year. Over 30 medical departments were represented at fairly coarse resolution; the parent health care organization's Call Center was represented at fairly fine resolution. Issues addressed by these simulations included various aspects of the ability of the system to respond to shifting member service area demographics and overall increases in membership. Results from the simulation included wait times for appointments, bed and staff utilization, financial reports, etc.
2. Effectiveness of Cardiac Clinical Practice Guidelines. Multi-decade simulations were performed for the same medical center, for roughly 3,000 males at high risk for heart disease. Detailed normal physiology and heart disease pathophysiology and risk factor models were run for each patient. The dozen or so medical departments relevant to cardiac care, along with the Call Center, were modeled at fairly fine resolution. This required implementation of FACET COA models for the clinical, clerical and logistical detailed behavior patterns that come into play in providing care for heart disease patients. The range of procedures employed by the medical center in providing emergency and inpatient care for cardiac patients was represented by 33 separate COA models, covering behavior patterns, such as:
  - Emergency Room (ER) check-in and triage,
  - ER chest pain diagnosis,
  - Patient admission procedures,
  - Physician supervision in the Intensive Care Unit,
  - Inpatient EKG and echocardiogram tests,
  - Inpatient defibrillation, and
  - Transfer of a patient from one inpatient medical department to another.

The general question addressed by the fine-resolution heart disease simulations was what would be the effect of varying the rates of patient compliance with guidelines for administering cholesterol-lowering and blood-pressure-lowering drugs on outcomes for

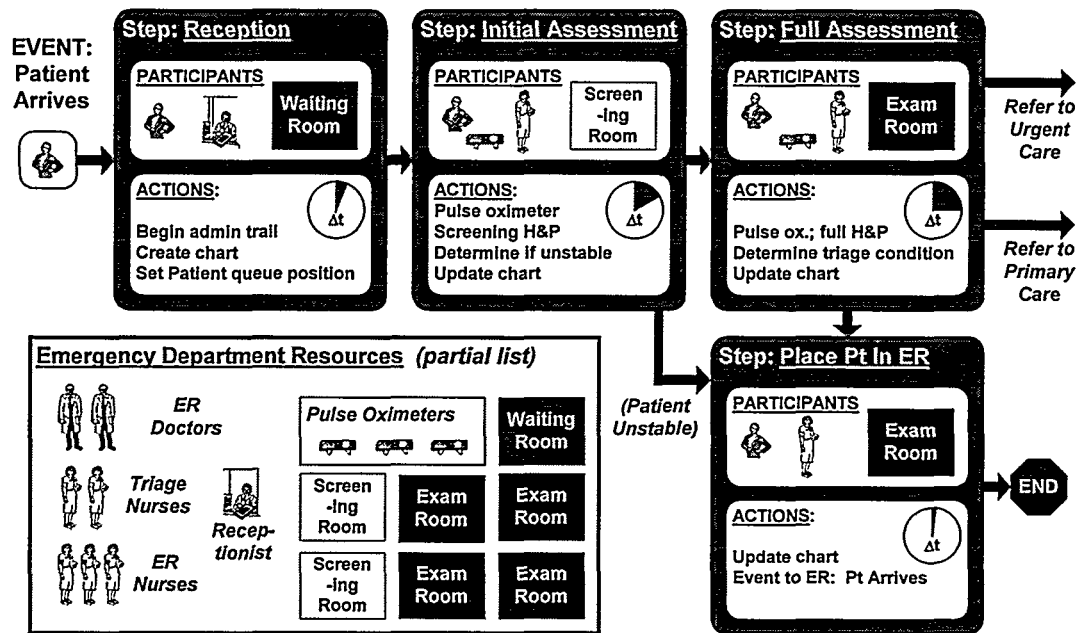


Figure 7. FACET Health Care COA Example: Emergency Room Standard Check-In and Triage

patients with high risk factors for heart disease? Results included incidence of chest pains and heart attacks, rates of heart-disease-related hospital admissions, unplanned emergency room visits, etc.

Figure 7 is a simplified illustration of one of the 33 cardiac care COAs, "Emergency Room Standard Check-In and Triage," which is invoked by the medical center's Emergency Department in response to an event indicating that a new patient has presented in the ER. In our cardiac care simulations, the behavior pattern represented in Figure 7 is implemented by an active COA for each patient who has presented at the medical center's ER and has not yet been treated, admitted, referred to other health care services or otherwise provided for. All of these various follow-up activities are covered by other COAs that are cued to activate based on the possible outcomes of the ER Standard Check-In and Triage COA process. Each active instance of this COA has the patient as a participant in all steps. The COAs contend with each other for the other necessary participants - all of

whom, or all of which, are resources of the Emergency Department, shown in the box in the lower left of the figure. The actual mechanisms for resolving precedence for patients' access to ER resources are carried within the "Perform Step" logic of the different steps.

#### PROSPECTS FOR FUTURE DEVELOPMENT AND APPLICATION OF FACET

Several diverse FACET application areas in addition to those described in this paper are planned or are in the early stages of development. They include:

- Military mission planning and resource management for "Operations Other Than War" applications (famine and flood relief, peacekeeping, etc.);

- Socioeconomic simulation of the South American cocaine industry, in the context of an effort to improve drug traffic interdiction effectiveness in the Caribbean, Central America, and Eastern Pacific regions;
- Operational risk assessment for financial institutions and others; and
- Cross-country logistics planning and management decision support.

It is ANL's intent to make the FACET package accessible to a wide range of users. One short-term measure undertaken to facilitate this goal is the porting of the framework from Smalltalk to Java. While Smalltalk was ideal for prototype development, we believe that a Java-based product will reach a larger audience. A longer-term goal is to provide FACET with a fully featured, intuitive graphical user interface to assist model developers in COA model implementation.

## ACKNOWLEDGMENT

This work is sponsored by the U.S. Department of Energy under contract number W-31-109-ENG-38.

## REFERENCES

- Ball, W.; D. Tucker; and Wilkinson, T.J. 1989. "The Tell Al-Hawa Project: Archaeological Investigations in the North Jazira 1986-1987." *Iraq* 51: 1-66.
- Christiansen, J.H. 2000. "A Flexible Object-Based Software Framework for Modeling Complex Systems with Interacting Natural and Societal Processes." To be published in Proceedings of the 4<sup>th</sup> International Conference on Integrating GIS and Environmental Modeling (Banff, Canada, Sept. 2-8).
- Dolph, J.E.; J.H. Christiansen; and P.J. Sydelko. 2000. "FACET: An Object-Oriented Software Framework for Modeling Complex Behavior Patterns." To be published in Proceedings of the 4<sup>th</sup> International Conference on Integrating GIS and Environmental Modeling (Banff, Canada, Sept. 2-8).
- Letcher, B.H.; J.A. Priddy; J.R. Walters; and L.B. Crowder. 1998. "An Individual-Based Spatially-Explicit Simulation Model of the Population Dynamics of the Endangered Red-Cockaded Woodpecker, *Picoides borealis*." *Biological Conservation* 86: 1-14.
- Wilkinson, T.J. 1994. "The Structure and Dynamics of Dry-Farming States in Upper Mesopotamia." *Current Anthropology* 35, No. 5 (Dec.): 483-519.