

4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4):
Problems, Prospects and Research Needs. Banff, Alberta, Canada, September 2 - 8, 2000.

FACET: an object-oriented software framework for modeling complex social behavior patterns

RECEIVED

GIS/EM4

AUG 04 2000

Q.S.T.I

*Jayne E. Dolph
John H. Christiansen
Pamela J. Sydelko*

The 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.

Abstract

The Framework for Addressing Cooperative Extended Transactions (FACET) is a flexible, object-oriented architecture for implementing models of dynamic behavior of multiple individuals, or "agents", in a simulation. These agents can be human (individuals or organizations) or animal and may exhibit any type of organized social behavior that can be logically articulated. FACET was developed by Argonne National Laboratory's (ANL) Decision and Information Sciences Division (DIS) out of the need to integrate societal processes into natural system simulations. The FACET architecture includes generic software components that provide the agents with various mechanisms for interaction, such as step sequencing and logic, resource management, conflict resolution, and preemptive event handling. FACET components provide a rich environment within which patterns of behavior can be captured in a highly expressive manner. Interactions among agents in FACET are represented by Course of Action (COA) object-based models. Each COA contains a directed graph of individual actions, which represents any known pattern of social behavior. The agents' behavior in a FACET COA, in turn, influences the natural landscape objects in a simulation (i.e., vegetation, soil, and habitat) by updating their states. The modular design of the FACET architecture provides the flexibility to create multiple and varied simulation scenarios by changing social behavior patterns, without disrupting the natural process models. This paper describes the FACET architecture and presents several examples of FACET models that have been developed to assess the effects of anthropogenic influences on the dynamics of the natural environment.

Keywords

Object-oriented, agent-based, social-modeling, land-management, land-use, planning, assessment.

Introduction

The effective integration of societal processes with natural processes is a challenging but necessary component of many environmental applications. Anthropogenic influences continuously and variously modify the dynamics and states of natural landscapes. Thus, realistic simulations must capture the effects of these influences on the physical environment. Geographic Information Systems (GISs) have long been used as a standard tool for environmental modeling and analysis. While GISs are invaluable as spatial analysis tools and are often sufficient for static model implementation, they have proven to be limited as integration frameworks for dynamic modeling (Sydelko et al. 1999,

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.**

Sydelko et al. 2000). Modeling and simulation that successfully capture the interplay between man and nature require an overarching architecture that is more extensive and flexible than GIS alone.

One significant limitation of GIS as a modeling framework is that it is not expressive enough to capture the interprocess dynamics of natural systems. Applications that require more than the simple passing back and forth of data - those that require the ability to reflect the dynamics of living ecosystems, including the effects of land uses and land management practices - are hindered in a strictly GIS-based environment. Developers frequently need to expand the scope of applications by adding in new models or modifying existing ones. In a GIS-based modeling environment, this often entails time- and cost-consuming reworking of model structure and source code. This is due in part to the fact that societal behavior such as land management/land use is incorporated into these simulations by adding management-specific parameters directly into the natural process model source code. Experience has proven, however, that such tight coupling of social behavior parameters with natural systems processes can create inflexible, monolithic models that are difficult and time-consuming to alter, thus making the assessment of alternative management/use policies an onerous task. Furthermore, many environmental land management and use practices involve nonspatial components that must be incorporated into the simulation process, but that can be difficult to articulate in an inherently spatial GIS.

For over a decade, Argonne National Laboratory (ANL) has been developing robust object-based software architectures to address complex, multidisciplinary modeling and simulation problems. Most of this work has focused on interactions among natural processes. However, as the scope and complexity of our simulated domains have grown to encompass human societal processes, ANL has recognized the value of building a unifying software infrastructure to support construction and interoperability of societal process models and of integrating these societal processes with natural process models. The need for such a framework has become particularly evident as ANL has explored the development of adaptive ecosystem management tools. Effective software of this sort requires the ability to articulate societal behavior patterns within a framework that promotes integration and feedback between these societal behaviors and the ongoing dynamics of natural systems. An equally important need, particularly for ecosystem management applications, is a component-based modeling framework that separates the implementation of the societal process models from the natural process models. This modular design philosophy provides a level of flexibility that facilitates rather than hinders application expansion via model additions or changes and that allows the developer to readily generate and implement simulation suites.

ANL has developed the Framework for Addressing Cooperative Extended Transactions (FACET) in response to the need for simulating integrated societal and natural processes. FACET is fundamentally a software suite for constructing object models of social behavior patterns. FACET was specifically developed 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. The term "agent" is used to denote simulation entities capable of autonomous, cooperative, and adaptive behavior. FACET societal models can be patterned on any logically articulated organized social behavior such as business practices, social mores and traditions, governmental and corporate policies, land management and use plans, clinical guidelines and procedures, etc. 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 is generically designed and allows implementation of societal behaviors that range in scope from simple stepwise flowcharts, such as would be needed to articulate a land use plan, to complex interactions between individuals and organizations engaged in multiple concurrent transactions and vying for the same resources.

FACET models operate within a process-based discrete event simulation framework that was built using ANL's Dynamic Information Architecture System (DIAS) (Christiansen 2000a). FACET models can be run on workstations or personal computers under UNIX or Windows NT. Using FACET models to implement societal behaviors of individuals and organizations within larger DIAS-based simulations has made it possible to conveniently address a broad range of issues involving interaction and feedback between natural and societal processes.

FACET Software Architecture Overview

The FACET software infrastructure is designed to explicitly support those behavior patterns characteristic of societal processes, as distinct from natural or physics-based processes. Societal process characteristics include:

- Processes involving 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 on the basis of social context.
- Processes can be interrupted and resumed at a later time.
- Processes can be preempted in progress or cancelled before they begin when "overtaken by events."

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.

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 serve to allow simulation "entity" objects to take part as agents in FACET-based behavior patterns. An entity object is defined as any real-world, discipline-specific object in a simulation (Christiansen 2000a). Examples of entity objects referencing the environmental domain are "Atmosphere," "Stream," "Soil," etc. Entity 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 (COA) Object Class Suite

The Course of Action (COA) object class suite formally defines a "collaborative extended transaction" model (Christiansen 2000b). Developers use the COA object class suite to build a COA model as a network of individual steps, much like a flowchart. Each step is represented by a "Step Template" object, which is similar to a submodel. Each Step Template has a specified list of participants (agents) for that step, resources owned and carried by each participant, and a locale. Each step (1) 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; (2) results in changes to the state attributes of one or more of the agents and/or changes to state attributes of other entity objects that have been declared to be of interest to the COA; and (3) may require resources held by agents participating in the action.

The key object classes that make up the FACET COA object class suite are the *COA Template*, the *COA*, and the *Step Template*. The COA Template (Figure 1) objects carry a directed graph of individual steps (Step Template objects). The Step Template objects specify the agents and other resources needed to carry out the step, along with instructions for exactly what is to be performed in the step (the societal model). Generally, only one instance of each COA Template object class will exist in the simulation frame of reference at any one time. Each unique behavior pattern, or COA model, is a single instance of a subclass of the COA Template object class.

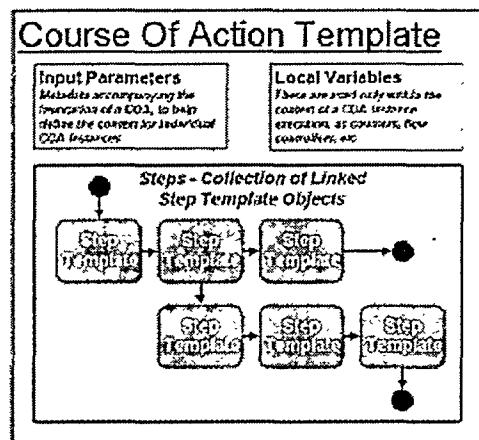


Figure 1 COA Template object class

The COA object class (Figure 2) is associated with individual active instances of a behavior pattern, or COA Template. In contrast to the COA Template object class, there will be only one *class* of COA object, but potentially a great number of instances of that class. Each COA object instance is essentially a bookmark that indicates the current state of a *specific* extended transaction among simulation agents that is being modeled in accordance with a COA Template. In a major simulation, there may be a very large number of agents or ad hoc groupings of agents embarked, in parallel, on the same COA. However, for each agent (or grouping of agents), the patterns will have progressed to different stages of completion.

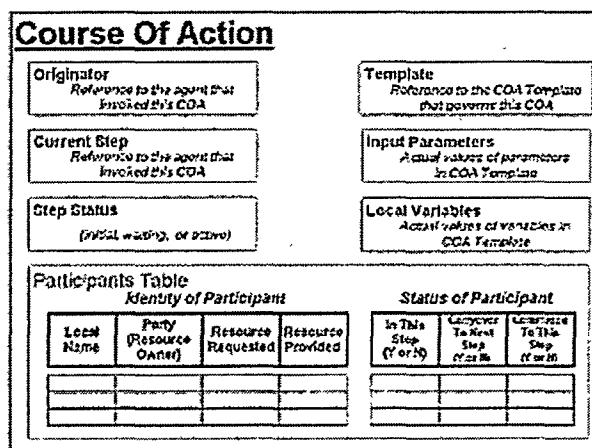


Figure 2 COA object class

The "Input Parameters" to the COA can be attributes of any object in the simulation, and these attributes may be GIS data layers. These input parameters define the simulation state at the start of a specific COA and may be updated by that COA. These parameters can be, and often are, the natural landscape objects in an environmental simulation. The input parameters essentially define the connections that a specified societal behavior (COA) has with the natural environment.

The centerpiece of the COA object class layout is the Participants Table. This table correlates the generic names of participants needed and other resources declared in the Step Template objects' detailed step action layouts with 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 Step Template object class (Figure 3) 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, waiting (for all requested participants to commit), or active. The Perform Step code block does the actual work of a step - changing state variables of participating agents, or other entity objects, and posting events and messages to other entity objects as needed. This is where the actual integration between societal and natural processes takes place. Societal behaviors in the COA can affect the state of the simulation by altering the attributes of, for example, environmental entity objects during the course of the COA execution. During the course of COA execution, the COA is "listening" for the updated state of the entity objects it has declared an interest in. Meanwhile, other dynamics in the simulation that also affect the state of the environment are simultaneously occurring in parallel with the COA societal processes. This allows the dynamics and feedbacks between the societal process and the natural processes in the environment to be simulated in a highly realistic way.

The "Decide Next Step" code block may have conditional logic for branching to alternate steps on the basis of 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 "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.

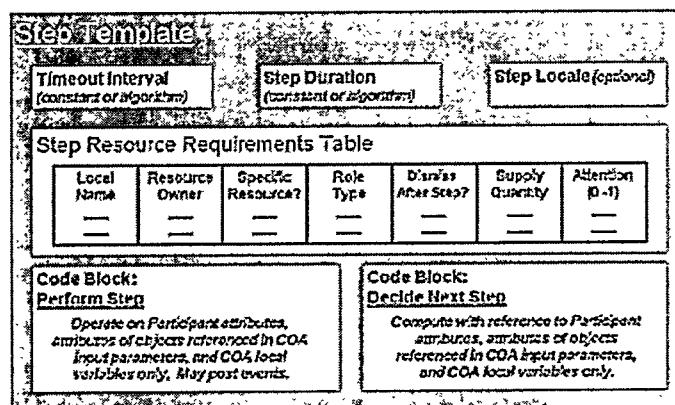


Figure 3 Step Template object class

The overall sequencing of processing for a single step in a FACET COA can be characterized as follows:

- 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 time out interval has expired, exception processing is performed. If sufficient response is received, the step can proceed for the duration specified. Step is now in active state, that is participants are busy performing the action of the step, as laid out in the "Perform Step" code block.
- During the step, participants' state attributes may be altered, other entity objects' state attributes may be changed, resource supply may be depleted or replenished, and events may be sent to other simulation entities.
- The Step Template's "Decide Next Step" code is triggered to determine which step will be invoked once the current step finishes.
- Participants are released from the step, the step ends, and the next step begins.

The Step Template object class is subclassed to accommodate three different types of steps: Actions, Expansions, and Exceptions. 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 expanding a step into a COA. The sub-COA referenced in this step then executes to completion and returns to the same Expansion step when finished. Expansion steps facilitate the use of nested COAs, which is a very helpful capability and a key to ease of reuse of COA components. Exceptions are the key to the ability of FACET COA models to respond realistically to unusual or unexpected 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 time out messages can also trigger an Exception step. An Exception step by definition has no participants and takes no elapsed simulation time to perform. It does, however, carry the logic needed to route the COA onto a different sequence of steps or to end the COA entirely. This mechanism allows COAs to preempt normal procedures, apply coping mechanisms or crisis management measures, etc., in response to changing conditions.

Participant Object Class

The Participant object class provides the software infrastructure that allows an entity object to take part as an agent in FACET COA model behavior patterns. A Participant object is assigned as a delegate object to each entity object for which COA model participation is desired. Essentially, this means that the entity object acquires additional attributes required for participation in a COA. 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 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: keeps track of all commitments of, and requests for, resources held by the agent.
- Role Object: allows agents to respond differently to requests depending on their current stance, or role, in the simulation.

COAs essentially negotiate with participants for their cooperation. The agendas of all potential participants are kept fully updated on the status of all other participants' commitment to a step. Each participant is free to join a step, may refuse to join, may defer joining pending other developments, may join a different step on its agenda, and may designate specific resources from its resource pools to meet a general resource request.

Together, the FACET COA and Participant objects provide a highly flexible and expressive means of representing diverse transactions and interactions. Developers can utilize a portion or all of the COA functionality in building societal process models, depending on the complexity and nature of their application.

FACET Application Examples

At ANL, FACET has been applied to a diverse collection of application areas. Two examples of FACET applications are provided to help illustrate the range and applicability of the FACET architecture, particularly as it applies to the environmental modeling domain.

Sustainable Land Management/Use on Military Land

ANL's Integrated Dynamic Landscape Analysis and Modeling System (IDLAMS) is a GIS-based framework that integrates data, environmental models (natural succession, forest spread, and species habitat), land use planning, and decision support technologies. IDLAMS provides environmental managers and decision makers with a strategic, adaptive approach to integrated natural resources planning and ecosystem management. Recently, an object-oriented approach has been incorporated into IDLAMS and IDLAMS has been reimplemented using the DIAS and FACET architectures. The new Object Oriented-IDLAMS (OO-IDLAMS) supports distributed, dynamic representation of interlinked processes (Sydelko et al. 2000).

For the new implementation, the land management and land use plans of IDLAMS were rewritten as separate COA models that address training, prescribed burning, and planting behaviors (Figure 4). These plans are inherently procedural in nature and readily lend themselves to COA implementation. Beyond the ease with which these plans could be articulated as COAs, there were several philosophical reasons for the reimplementation. First, ANL recognized that the nature of vegetation change is fundamentally different, depending on whether it is initiated by natural or societal processes. Natural succession and forest spread are physical, ecosystem-driven processes, while planting, burning, and training are management/use activities that result in man-made changes to vegetation state and are more transient in nature. By isolating each process as a separate model, the important linkages and feedbacks that may occur between them are more easily identified. Second, separating the management aspect of vegetation change from the physical processes is in keeping with a modular design approach and provides greater flexibility to create multiple and varied simulation scenarios. Developing distinct COAs to express the management and use processes facilitates changing management scenarios without disturbing the natural process model source code. By implementing these processes as separate models, alternative plans can be readily plugged in the future, without disrupting the existing system. This feature represents significant savings in cost and

time.

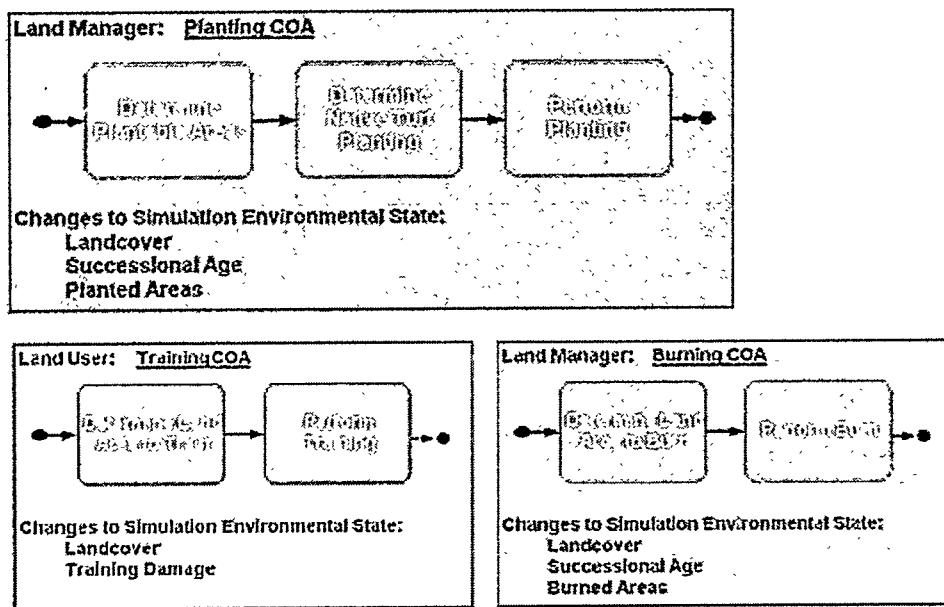


Figure 4 OO-IDLAMS Land Management and Land Use COAs

The initial OO-IDLAMS COAs illustrated here implement simple, limited step land management and use plans that modify environmental entity objects in the simulation. While simple, these COAs do provide run time feedback between natural and societal model processes. In the current implementation, feedback exists between the OO-IDLAMS habitat model and the planting COA. Patches of suitable habitat are created by the planting COA, which fulfills the vegetation requirements as specified by the habitat model. This is analogous to a land manager modifying a planting plan on the basis of the dynamic state of the vegetation and with knowledge of what constitutes an acceptable vegetation patch for a particular species.

While simple, the development of these COAs has laid the foundation for adding complexity into the societal behavior of the OO-IDLAMS simulation in future scenarios. An example of this would be the incorporation of training rotation into the training plan that would allow rest to the newly planted areas. It also may be desirable to have the training COA consider habitat sensitivities, such as whether or not it is breeding season. The extensive and comprehensive design of the FACET architecture also provides for the inclusion of resource contention issues which may be needed, for example, in the case of a forest planning application.

Sustainability of Ancient Mesopotamian Urban Centers

ANL is building a dynamic object model of ancient Mesopotamian life using the DIAS and FACET modeling frameworks. The objective of this work is to gain a better understanding of the dynamics of ancient Mesopotamian urban centers by exploring their sustainability, growth, or decline in the face of environmental stress using modeling and simulation.

The initial simulation prototype represents a nucleated agricultural community in ancient northern

Mesopotamia. Simulations will address the integration of natural (weather, crop growth, hydrology, population dynamics, etc.) and societal (farming practices, kinship-driven behaviors, etc.) processes on a daily basis across 200-year simulations. Dynamic behaviors of objects representing households, fields, crops, herds, etc., are simulated with existing natural systems models, such as the U.S. Department of Agriculture's EPIC model for field crop dynamics and the National Oceanographic and Atmospheric Administration's paleoclimate models, in combination with new FACET agent-based societal models developed by ANL. GIS information drawn from fieldwork and satellite observations is being used to help assemble the ensemble of landscape objects needed for the simulations. A key outcome of the simulation is expected to be a new level of insight into the ability of such a community to adapt in response to environmental stressors such as droughts and soil erosion.

Figure 5 illustrates one COA that has been developed to model cropping behavior by a Mesopotamian family. This COA employs resource management mechanisms that allocate individuals to tasks within a step (harvest, sow, etc.) on the basis of the individual's current agenda and the state of the entire resource pool. As in the previous OO-IDLAMS example, the societal (agricultural) behaviors in this COA serve to modify environmental entity objects in the simulation.

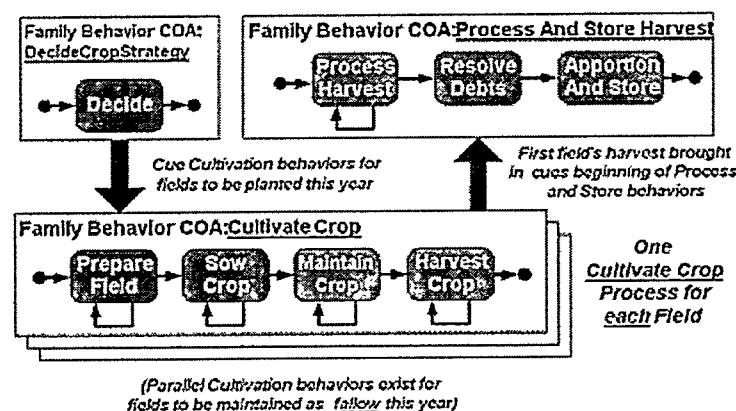


Figure 5 Mesopotamian Agricultural Behavior COA

Conclusions

FACET provides the expressiveness, flexibility, and modularity necessary to address the complex, multidisciplinary nature of environmental processes. The object-oriented design of FACET (and DIAS) promotes run-time feedback between societal and natural models, thereby making it a more dynamic environment for exploring the interactions among ecosystem processes. Separating the implementation of societal processes from the natural process model behaviors they influence, minimizes (or eliminates) the need to rework or recreate a system as new models are added to the suite. This modularity facilitates the rapid creation of multiple and varied simulation suites and promotes the "plug and play" of alternative models.

Future Research Needs

Currently, ANL is exploring the use of FACET for the development of more complex COA models, such as those that would be needed to implement scenarios of urban development and sprawl,

sustainable development in both rural and urban areas, and the socioeconomic impacts of global climate change. ANL is also considering the expansion of FACET to include complex adaptive system functionality such as would be necessary to simulate adaptive land management.

Acknowledgements

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

References used

- Christiansen JH. 2000a. A flexible object-based software framework for modeling complex systems with interacting natural and societal processes. 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs. Banff, Alberta, Canada. September 2 - 8, 2000.
- Christiansen JH. 2000b. FACET: A simulation software framework for modeling complex societal processes and interactions. To be published in: Proceedings of the 2000 summer computer simulation conference of the society for computer simulation international; 2000 Jul 16-20; Vancouver, British Columbia, Canada.
- Sydelko PJ, Majerus KA, Dolph JE, Taxon TN. 1999. A dynamic object-oriented architecture approach to ecosystem modeling and simulation. In: Proceedings of 1999 American Society of Photogrammetry and Remote Sensing (ASPRS) Annual Conference. 19-21 May 1999, Portland, OR. p. 410-421.
- Sydelko PJ, Dolph JE, Majerus KA, Taxon TN. 2000. An advanced object-based software framework for complex ecosystem modeling and simulation. 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs. Banff, Alberta, Canada. September 2 - 8, 2000.

Authors

Jayne E. Dolph, Environmental Computer Scientist, Decision and Information Sciences Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois, USA, 60439-4832. Email: dolph@anl.gov, Tel: 630-252-1654, Fax: 630-252-5128.

John H. Christiansen, Environmental Computer Scientist, Decision and Information Sciences Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois, USA, 60439-4832.

Email: jhc@anl.gov, Tel: 630-252-3291, Fax: 630-252-5128.

Pamela J. Sydelko, Manager, Environmental Modeling Section, Decision and Information Sciences Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439-4832 Email: psydelko@anl.gov, Tel: 630-252-6727, Fax: 630-252-5128.