

A flexible object-based software framework for modeling complex systems with interacting natural and societal processes

GIS/EM4

John H. Christiansen

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.

Abstract

The Dynamic Information Architecture System (DIAS) is a flexible, extensible, object-based framework for developing and maintaining complex multidisciplinary simulations. The DIAS infrastructure makes it feasible to build and manipulate complex simulation scenarios in which many thousands of objects can interact via dozens to hundreds of concurrent dynamic processes. The flexibility and extensibility of the DIAS software infrastructure stem mainly from (1) the abstraction of object behaviors, (2) the encapsulation and formalization of model functionality, and (3) the mutability of domain object contents. DIAS simulation objects are inherently capable of highly flexible and heterogeneous spatial realizations. Geospatial graphical representation of DIAS simulation objects is addressed via the GeoViewer, an object-based GIS toolkit application developed at ANL. DIAS simulation capabilities have been extended by inclusion of societal process models generated by the Framework for Addressing Cooperative Extended Transactions (FACET), another object-based framework developed at Argonne National Laboratory. By using FACET models to implement societal behaviors of individuals and organizations within larger DIAS-based natural systems simulations, it has become possible to conveniently address a broad range of issues involving interaction and feedback among natural and societal processes. Example DIAS application areas discussed in this paper include a dynamic virtual oceanic environment, detailed simulation of clinical, physiological, and logistical aspects of health care delivery, and studies of agricultural sustainability of urban centers under environmental stress in ancient Mesopotamia.

Keywords

Object-oriented, framework, modeling, simulation, complexity, agent-based, socioeconomic, environmental, distributed.

Introduction

Real-world dynamic landscapes are characterized by the interplay of diverse natural and anthropogenic processes interacting both within and across discipline boundaries, and at varying

spatial and temporal scales. It is a major challenge to assemble a simulation system that can successfully capture the dynamics of such complex systems, and an even more serious challenge to be able to nimbly adapt such a simulation to shifting and expanding analysis requirements and contexts. Argonne National Laboratory (ANL) has developed the Dynamic Information Architecture System (DIAS) to help address these needs. DIAS is a flexible, extensible, object-oriented framework for developing and maintaining complex simulations. DIAS supports fully distributed simulations in which the dynamic behaviors of the software “domain objects,” representing the diverse entities that populate a simulation, are implemented by an ensemble of simulation models (new models and/or existing “legacy” models) implemented in virtually any programming language. DIAS is expressly designed to address complex, cross-domain modeling problems over a wide range of simulation fidelity and level of detail.

DIAS has been under development at ANL since 1993 for a series of governmental and private sector sponsors with very diverse needs. DIAS is explicitly designed to be context-neutral; it supports simulations in essentially any domain. DIAS application areas to date include dynamic terrain- and weather-influenced military unit mobility assessment; integrated land management at military bases; a dynamic virtual oceanic environment; clinical, physiological and logistical aspects of health care delivery; avian social behavior and population dynamics; and studies of agricultural sustainability under environmental stress in ancient Mesopotamia.

This paper lays out the salient characteristics of the DIAS software architecture and uses a few recent and ongoing DIAS applications to illustrate the use of various DIAS features and capabilities.

DIAS Software Architecture Overview

DIAS is an object-oriented software framework for building and running process-based discrete-event simulations. DIAS is implemented in the Smalltalk language, with sparing use of C and C++ to improve mathematical and graphical performance. To make DIAS accessible to the largest possible audience, a port of DIAS to Java is underway, to be completed by the end of summer 2000. DIAS is presently hosted on UNIX workstations and on PCs running Windows NT. DIAS supports fully distributed operation via the Common Object Request Broker Architecture (CORBA). DIAS can be supported by a Gemstone object database management system to provide persistence to DIAS simulation objects.

The DIAS software infrastructure allows many disparate simulation models and other applications to work together harmoniously to address a complex problem. The main components of any DIAS simulation are (1) the “domain objects”: software objects that represent the real-world entities that make up the problem space and (2) simulation models and other applications that express the dynamic behaviors of the domain entities. Existing “legacy” models in virtually any software language can participate in complex DIAS simulations.

The DIAS software architecture supports and enforces an important set of design traits that enable it to accommodate widely varying simulation contexts. These design traits include:

- *Mutability of DIAS domain object contents.* DIAS domain objects can tailor their attributes and behaviors to the context of a simulation.
- *Abstraction of domain object behaviors.* DIAS domain objects declare their behaviors abstractly. Linkage of these abstract behaviors to appropriate modeling functionality is a context-dependent, run-time activity.
- *Encapsulation and formalization of model functionality.* DIAS interface objects formally define and isolate models and other applications that implement domain object behaviors, and map them to specific domain object behavior aspects. Models are constrained by DIAS model interface objects to communicate directly only with domain objects, never directly with each other.

The key DIAS software infrastructure elements underlying these general design traits are discussed briefly below.

Much of DIAS's flexibility and extensibility stems from the highly generic high-level representation of domain objects. This is illustrated by the "DIASEntity," "DParameter," and "SpatialDataSet" infrastructure object classes.

DIASEntity Object Class

All DIAS domain objects are derived from the DIASEntity object class. DIASEntity objects and their subclasses carry most of their attributes in a flexible dictionary of "DParameter" objects. Each domain object class derived from the DIASEntity object class carries a dictionary of MetaParameter objects defining the superset of all possible attributes for that subclass. The working attribute set for a given specific DIASEntity subclass instance can be tailored to the precise context in which it will be used by activating DParameter objects (carrying the actual attribute data) selectively from the MetaParameter list only as needed.

DIASEntity-based domain objects also carry a collection of "Aspect" objects that abstractly represent specific behaviors for the subclass. If a particular aspect of a domain object's behavior is needed in a given simulation context, a "DIASProcess" object standing as proxy for a specific model implementation is linked at run time to that Aspect, to allow the domain object to exhibit the behavior.

A DIASEntity object can carry geospatial information about itself by including a DIAS "Locus" object as a parameter in its parameter dictionary. The Locus object class is highly generalized and flexible. It carries the following attributes:

- *Locus Elements:* a collection of geometric objects (points, polylines, simple or complex polygons, etc.).
- *Locus Basis:* a reference to a DIAS LocusBasis object. The LocusBasis class has separate subclasses for polar geocentric, Cartesian projected from geocentric, and simple Cartesian (not georeferenced) coordinate systems. LocusBasis object attributes include (1) a "Projection" object that completely specifies a standard map projection and (2) an optional affine transformation matrix that applies scaling, rotation, and/or translation operations to the projection's coordinates. LocusBasis objects incorporate methods that implement the most

frequently encountered coordinate system transformations, with additional methods added as needed.

DParameter Object Class

DParameter objects carry object attributes, along with metadata (valid range, units, default values, etc.) about the attributes. The metadata are accessed from the DParameter via a reference to an appropriate DIAS MetaParameter object. The MetaParameters include an annotation object that allows different *versions* of the same parameter to be carried simultaneously by a DIASEntity. DParameters are also capable of maintaining a time series of their attribute values.

SpatialDataSet Object Class

The DIAS Locus functionality is very useful but is still not sufficient for representing the attributes of dispersed, heterogeneous environment objects such as the atmosphere, oceans, or land cover. DIAS SpatialDataSet (SDS) objects are essentially an extension of the DParameter object class that allows DParameters to express the degree of spatial variability needed for such complex domain objects. SDSs are software objects carrying (1) a complete geometric specification for a 2-D or 3-D spatial partitioning scheme (grid, mesh, network, patchwork, etc.) that divides a region into 2-D or 3-D cells and (2) a collection of data elements (DParameters) for each cell.

A data dictionary specified at the SDS level is used for every Cell in the SDS. SDSs can extend DIASEntity object attribute specifications to include spatial dependencies and can aid in compression of spatial data by requiring only one specification of a spatial partition that may be used by several attributes.

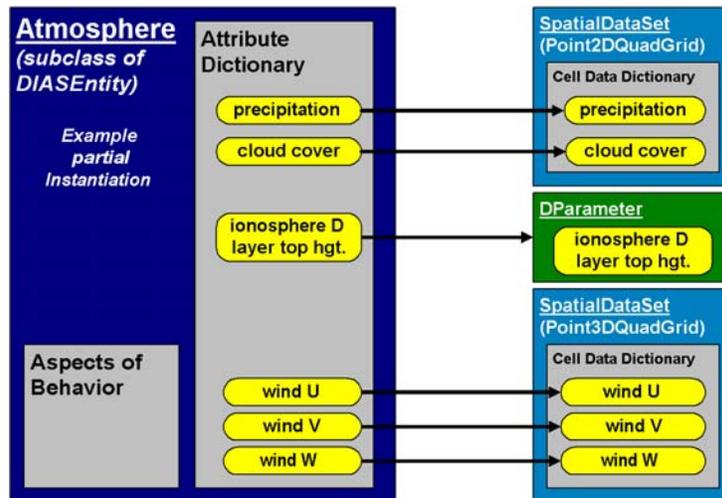


Figure 1 Use of SpatialDataSet and DParameter attributes in a simple Atmosphere object.

SDSs are of two fundamental types: PointSets, in which data are associated with *discrete points* in 2-D or 3-D space, and CellSets, in which data are associated with *regions* in 2-D or 3-D space. CellSet attributes may be associated either with Cells (regions) or with Faces (boundaries between regions). There are presently about 20 different DIAS subclasses of 2-D and 3-D SDSs, embodying different topologies such as grids, meshes, networks, unstructured collections, and patchworks.

Figure 1 provides an illustration of the use of DParameters and SDSs in a simplified implementation of an “Atmosphere” object. Here, the object carries 3-D wind vector components in one SDS, 2-D precipitation and total cloud cover in a different SDS, and ionosphere “D” layer height as a scalar parameter that is assumed uniform across the area of interest. It is important to note that spatial queries directed at the Atmosphere object do not need to have any knowledge of the specific spatial representation scheme that the Atmosphere is using for its attributes; translation to the requestor’s locus basis is automatically provided by the receiving objects. The internal representation can optionally be exposed if needed for a particular type of interpolation or other spatial query.

DIAS Linkage between Domain Objects and Simulation Models

In DIAS, each domain object class carries a set of abstract declarations of the various aspects of its dynamic behavior, rather than hard-coded model functionality. These behavior aspects (in the form of DIAS Aspect objects) are linked, at run time, to models qualified to address them, via proxy DIASProcess objects (see Figure 2). DIASProcess objects carry formal definitions of the

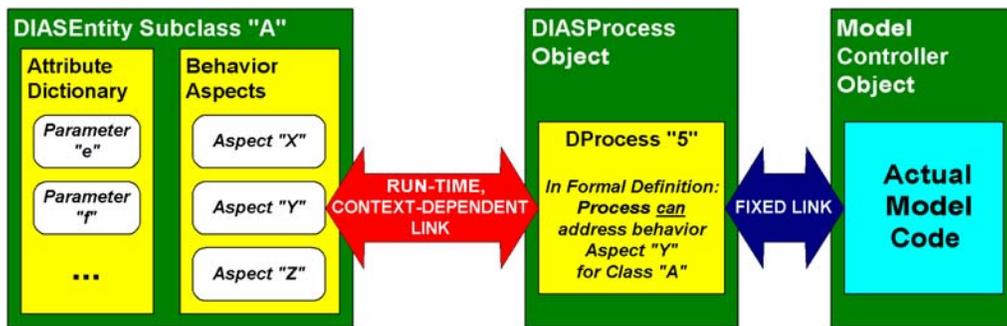


Figure 2 Generic DIAS example of flexible, abstract linkages between domain objects and simulation models

capabilities and limitations of specific, actual model implementations that are available to the simulation framework. DIASProcess specifications identify which object behaviors (as represented by abstract Aspect objects) the application is qualified to implement. Each DIASProcess also carries a formal specification of the input and output data dictionaries of the application it represents, along with a link to the actual application code itself (wherever it resides in a distributed network). Each actual model or other application is accessed by its DIASProcess object via a “ModelController” object, which acts as an “object wrapper” for the model, hiding model internals from the rest of the simulation system. ModelControllers are responsible for translating DIAS object state variables (which are, by DIAS definition, the “common language” of the simulation) into model-specific input terms and for translating model-specific output back into the form of updates to DIAS object attributes. This approach greatly facilitates plug-and-play of alternative model formulations that address the same domain object behaviors but in different ways.

In a DIAS-based simulation, models may communicate only with domain objects, and never directly with each other. This stricture greatly reduces the number and complexity of model linkages needed and further improves reusability. New models developed with DIAS in mind generally access the attributes of appropriate domain objects directly. On the other hand,

existing “legacy” models utilize ModelController object methods to translate idiosyncratic model-specific input and output data dictionaries in terms of the state attributes of the domain objects in a simulation. In broad terms, a generic DIAS simulation framework will take the form shown in Figure 3. As the figure indicates, external models and other applications are linked, via

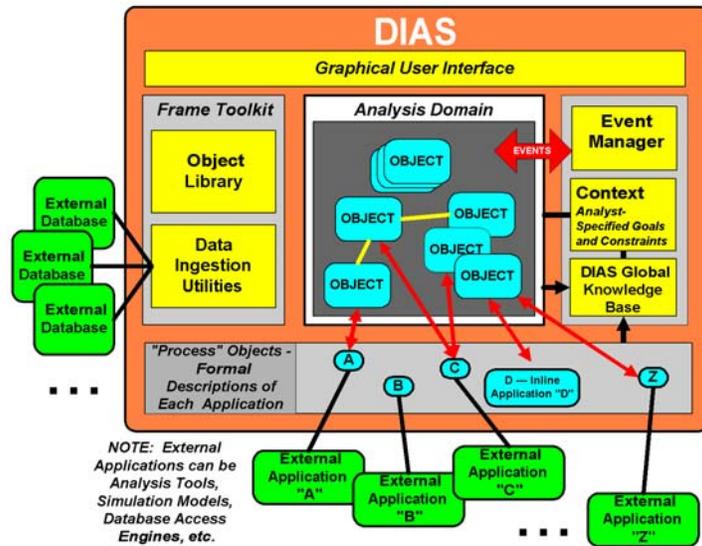


Figure 3 Generic layout for a DIAS-based simulation system

run-time connections between their DIASProcess objects and DIASEntity Aspect objects, to the domain objects whose dynamic behaviors they are implementing for the current simulation context. The “Context Manager” function basically supports assignment of domain object Aspect-to-DIASProcess linkages that are appropriate to a simulation context defined by the user via the graphical user interface.

The GeoViewer

The GeoViewer is an object-based geographical information system (GIS) toolkit developed at ANL. It can function in a stand-alone mode or as a component of a larger framework. In the context of DIAS, the GeoViewer is used to provide access to the objects that make up the modeling domain for simulation construction, query, and results display. In addition to its repertory of standard spatial query, analysis, and display functionality, the GeoViewer has the ability to provide direct access to DIAS simulation domain objects via its geospatial displays. The “lightweight” graphical objects used in GeoViewer displays are dynamically coupled with their underlying domain objects, so that changes in the state of domain objects can be reflected in the geospatial displays.

The FACET Simulation Framework

The most recent extension to the generic simulation support capabilities of DIAS comes from inclusion of societal process models generated by FACET (Christiansen 2000; Dolph et al. 2000), another object-based framework developed at ANL. FACET, the Framework for Addressing Cooperative Extended Transactions, is a facility for constructing agent-based object models of social behavior patterns. 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 among natural and societal processes.

FACET consists of a set of software objects that facilitate building 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, and governmental and corporate policies. 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.

DIAS Applications

Integrated Ocean Architecture

The Integrated Ocean Architecture (IOA) project (Hummel et al. 2000) addresses a long-term need of the U.S. Department of Defense Ocean Executive Agent (OEA) for a virtual maritime environment within which diverse domain processes can interact, to support various analysis and planning functions. The initial OEA need, addressed by ANL in 1999, was for surf zone forecasts for amphibious landing areas. The OEA identified a suite of models that together addressed the relevant domain processes:

- WAM: deep ocean wind-driven wave generation and propagation,
- ADCIRC: tide and wind effects on mean water surface elevation,
- STWAVE: nearshore wave propagation (for simple shorelines),
- REF/DIF: nearshore wave propagation (for complex shorelines), and
- SURF: surf zone wave characteristics and longshore currents.

ANL used DIAS and the GeoViewer GIS toolkit to assemble an object framework that, for the first time, allows these models to work together in the same simulation. The IOA prototype application (Figure 4) exercises the five oceanic models as behaviors of DIAS domain objects.

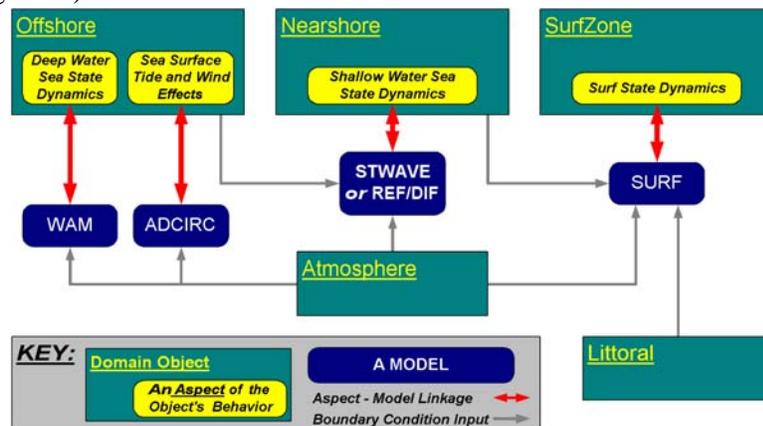


Figure 4 Object interactions for the Integrated Ocean Architecture prototype

The Atmosphere object is populated with data via archived COAMPS model mesoscale weather forecasts, accessed over the internet from the Defense Modeling and Simulation Office's Master Environmental Library. Figure 4 shows that the "Shallow Water Seas State Dynamics" behavior of the Nearshore object can be addressed by *either* the STWAVE or the REF/DIF model. The choice is context-dependent; REF/DIF should be selected over STWAVE if the shoreline of interest is complex enough to create wave refraction and diffraction conditions too difficult for the STWAVE model to simulate adequately.

HealthSim Health Care Simulator

ANL has used both the DIAS and FACET 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 levels 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 by using mathematical (e.g., differential equation integrator) physiological models.
- Human cognitive behaviors, such as response to symptoms, making and keeping appointments, etc.
- Clinical/logistical processes, such as clinical practice guidelines, office procedures, personnel policies.
- Medical monitoring equipment response.

In HealthSim, natural processes (human physiology) are inextricably interlinked with societal processes representing the behavior patterns of patients, caregivers, and the organizational components of the health care delivery system. In these simulations, domain objects include facilities, equipment and supplies, 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 dynamic behaviors of these Physiology objects.

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

1. *Access to Care.* Multiyear simulations were performed for the entire 300,000-member population of a large medical center 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. 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.
2. *Effectiveness of Cardiac Clinical Practice Guidelines.* Multidecade 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 were modeled at fairly fine resolution. This effort required implementation of 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 FACET societal behavior pattern models. These models covered behavior patterns such as Emergency Room (ER) check-in and triage, patient admission procedures, and inpatient EKG and echocardiogram tests. The fine-resolution heart disease simulations were used to estimate the effects of varying the rates of patient compliance with guidelines for administering cholesterol-lowering and blood-pressure-lowering drugs on health outcomes for patients with high risk factors for heart disease.

Sustainability of Ancient Mesopotamian Urban Centers

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, by using the DIAS and FACET modeling frameworks. The objective is to gain 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). Dynamic behaviors of software objects representing households, fields, crops, herds, etc. are simulated by using (1) proven, 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, and (2) new FACET agent-based societal models based on archaeological and anthropological evidence. GIS information drawn from field work and satellite observations is being used to help assemble the ensemble of landscape and infrastructure domain 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.

Prospects for Future DIAS Development and Application

An ongoing research and development goal at ANL is to be able to more fully automate the DIAS "context management" function, in which *ad hoc* connections are made between domain objects and simulation models and other applications to meet a user's simulation goals and constraints. We plan to have an expert system examine (1) the user's formal specification of simulation context, (2) the attributes and abstract behaviors of the domain objects in the modeling frame of reference, and (3) the formal specifications of model capabilities carried in DIASProcess objects, to suggest model linkages that address the user's concerns in various

locally optimal ways. Such a capability was successfully demonstrated several years ago at ANL, using the National Aeronautics and Space Administration's CLIPS expert system shell. A key to this effort will be identification of analysis domains that are complex enough, with enough alternative approaches and tradeoff possibilities, to make an automated or semiautomated DIAS context manager worthwhile.

Other DIAS applications are planned or underway in such domains as structural microbiology, cross-country logistics planning and management, and military mission planning and resource management for "Operations Other Than War," such as disaster relief and peacekeeping efforts.

Acknowledgments

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

References used

Ball W, Tucker D, Wilkinson TJ. 1989. The Tell Al-Hawa project: archaeological investigations in the north Jazira 1986-1987. *Iraq* 51: 1-66.

Christiansen JH. 2000. 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.

Dolph JE, Christiansen JH, Sydelko PJ. 2000. FACET: an object-oriented software framework for modeling complex behavior patterns. To be published in: Proceedings of the 4th international conference on integrating GIS and environmental modeling; 2000 Sep 2-8; Banff, Alberta, Canada.

Hummel JR, Christiansen JH, Hield CW, Taxon TN, Allard R. 2000. The integrated ocean architecture – a framework for generating seamless oceanic representations for modeling and simulation applications. To be published in: Proceedings of the 68th military operations research society symposium; 2000 Jun 20-22; Colorado Springs, Colorado, USA.

Wilkinson TJ. 1994. The structure and dynamics of dry-farming states in upper Mesopotamia. *Current Anthropology* 35 (5): 483-519.

Author

John H. Christiansen, Manager, Complex Adaptive Systems Section, 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.