ERDC/CERL TR-05-1

Construction Engineering Research Laboratory

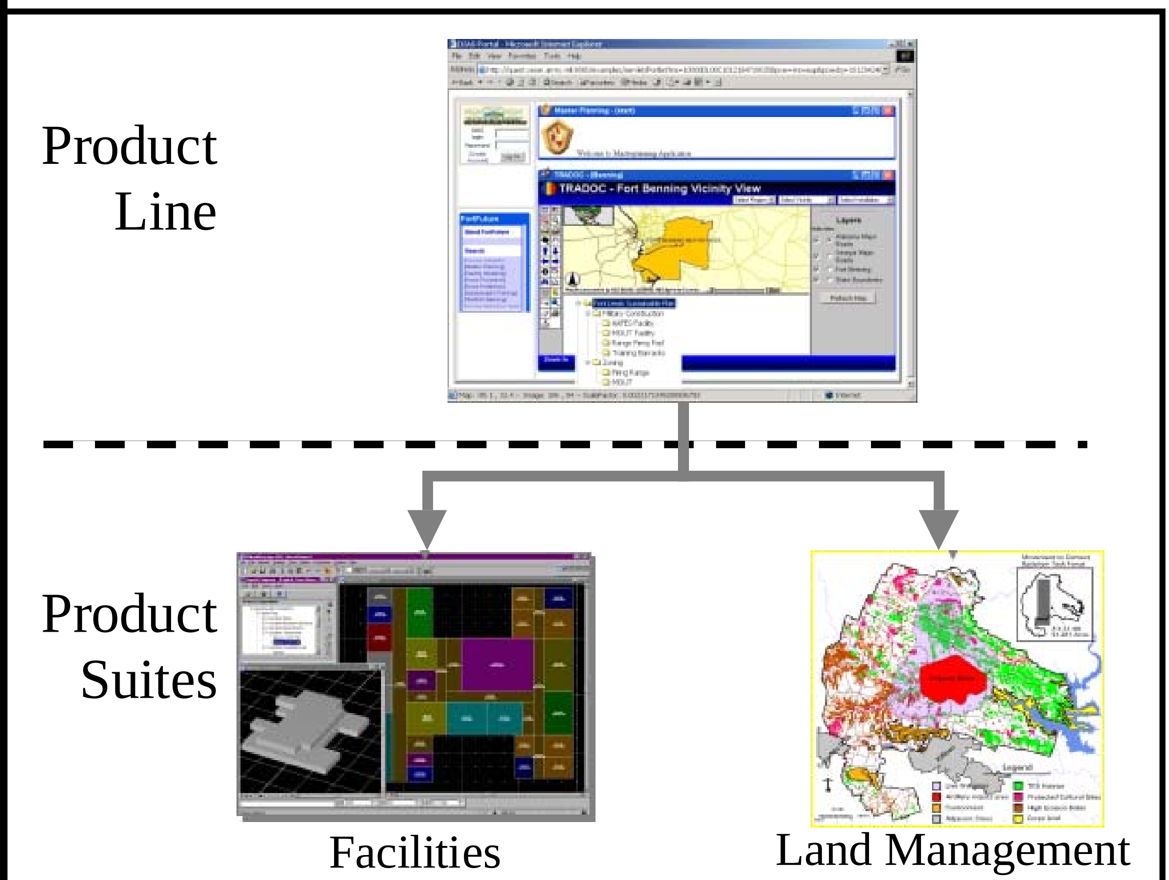
US Army Corps of Engineers®

Engineer Research and Development Center

Preliminary Guidelines for Installation Product Line Land Management Suite Product Developers

James Westervelt, Kelly M. Dilks, and William D. Goran

January 2005



ERDC/CERL TR-05-1
January 2005

Preliminary Guidelines for Installation Product Line Land Management Suite (LMS) Product Developers

James Westervelt, Kelly M. Dilks, and William D. Goran

Construction Engineering Research Laboratory
PO Box 9005
Champaign, IL 61826-9005

Final Report

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit "Land Management Technology Integration"

ABSTRACT: Several generations of land management support software have been developed for and delivered to military installation range control and environmental offices. A new generation of software development tools, hardware and networking environments, and operating systems provide an opportunity to design, develop, and deliver a new generation of tools that will be far more effective than previous versions. These tools will be easier to access, maintain, and integrate. This document provides an introduction to the new technologies and techniques and is written for all involved with installation land management software development including managers, supervisors, principal investigators, and programmers.

Contents

List of Figures and Tables

Figures

Tables

Conversion Factors

Non-SI* units of measurement used in this report can be converted to SI units as follows:

Multiply	By	To Obtain
acres	4,046.873	square meters
cubic feet	0.02831685	cubic meters
cubic inches	0.00001638706	cubic meters
degrees (angle)	0.01745329	radians
degrees Fahrenheit	(5/9) x (°F—32)	degrees Celsius
degrees Fahrenheit	(5/9) x (°F—32) + 273.15.	kelvins
feet	0.3048	meters
gallons (U.S. liquid)	0.003785412	cubic meters
horsepower (550 ft-lb force per second)	745.6999	watts
inches	0.0254	meters
kips per square foot	47.88026	kilopascals
kips per square inch	6.894757	megapascals
miles (U.S. statute)	1.609347	kilometers
pounds (force)	4.448222	newtons
pounds (force) per square inch	0.006894757	megapascals
pounds (mass)	0.4535924	kilograms
square feet	0.09290304	square meters
square miles	2,589,998	square meters
tons (force)	8,896.443	newtons
tons (2,000 pounds, mass)	907.1847	kilograms
yards	0.9144	meters

* *Système International d'Unités* ("International System of Measurement"), commonly known as the "metric system."

Preface

This study was conducted for Headquarters, U.S. Army Corps of Engineers (HQUSACE) under Project 622720A986, “Base Facilities Environmental Quality,” Work Unit “Land Management Technology Integration.” The technical monitors (and associated Technical Directors) were William D. Severinghaus and Cary Butler, ITL.

This work was performed by the Heritage and Conservation Branch (CN-H) of the Installations Division (CN) of the Construction Engineering Research Laboratory (CERL). The CERL principal investigator was Kelly M. Dilks. Lucy Whalley is the Chief, CEERD-CN-H and John Bandy is Chief, CEERD-CN. Appreciation is owed to the following ERDC personnel for their constructive preliminary technical reviews of this work: Cary Butler, Information Technology Laboratory (ITL); Michael Case, CERL; and Denise Martin, Cold Regions Research and Engineering Laboratory (CRREL). The technical editor was William J. Wolfe, ITL. Part of this work was performed in coordination with the Information Technology Laboratory. The Director of CERL is Dr. Alan Moore.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Commander and Executive Director of ERDC is COL James R. Rowan, and the Director of ERDC is Dr. James R. Houston.

1 Introduction

Background

Fort Future is a major research and development (R&D) effort designed to produce capabilities critical to the Army's ability to transform its installations in the tight timeframe required to support our emerging forces. Much as field commanders gain a superior advantage by visualizing the battle space, Fort Future will enable installation planners to make strategic decisions by visualizing results of many different scenarios.

Fort Future R&D is being conducted by the U.S. Army Engineer Research and Development Center (ERDC) in support of the Office of the Assistant Chief of Staff for Installation Management (OACSIM). Fort Future will deliver a suite of tools to support early analyses of alternative approaches available to an installation for supporting a changing mission. The land management suite will contain an integrated and interoperable set of software programs developed to assist installation land managers with national, regional, and local scale planning and management of installation lands including training and testing areas.

Traditionally, software products developed to support military land management have been created as separate, standalone efforts. Table 1 lists some sample products, each of which has unique data requirements and user interfaces, and runs with certain hardware and software support requirements. These systems do not share a common data base. They have unique user interfaces, must be separately installed and maintained, and must be compiled to work on a variety of different operating systems. These standalone software products have great utility, but if they were developed as part of an integrated suite, they would be more useful, less costly to develop, and they would greatly improve communication. This work was undertaken to integrate the components of a land management suite in which:

1. *Models will share a common database.* This will make it easier for customers to maintain necessary input data.
2. *Separate programs will share a common user interface.* This will reduce the learning curve associated with each product.
3. *All software will be marketed, distributed, and maintained centrally.*

Table 1. Sample software products developed to support military land management.

Type	Description	Sample Software
Training capacity	Evaluation of the ability of a areas to support intended training and/or testing	ATTACC—Army Training and Testing Area Carrying Capacity TUDM—Training Use Distribution System
Natural Resource Management	Products that predict the state of installation training areas based on the training/testing.	PRISM—Planning and Resource Integration Stewardship Modules EDYS—Environmental Dynamics System IDLAMS—Integrated Dynamic Landscape Analysis System FHASM—Fort Hood Avian Simulation Model
Geographic Information Systems	Commercial and government GIS	ArcInfo, GRASS
Training/testing area scheduling	Systems used in real-time to schedule and manage ranges	RFMSS—Range Facility Management Support System
Hydrology	Models to predict hydrology and associated erosion/deposition	GSSHA - Gridded Surface Subsurface Hydrologic Analysis RUSLE—Revised Universal Soil Loss Equation SIMWE—SIMulated Watershed Evaluation
Dust and Smoke Simulation	Simulate smoke and dust plumes associated with training	
Noise	Simulate the patterns of noise in time and space associated with training	SARNAM—Small-Arms Range Noise Assessment Model BNOISE—Blast Noise simulation model
Habitat Suitability Models	Models that predict suitability of habitat on a species by species basis.	RCW—Red Cockaded Woodpecker model

Objective

The objective of this work is to provide preliminary guidelines for product developers of the Fort Future product line's land management suite.

Scope

These guidelines are intended for use by product developers and integrators (not for end users of the Installation Product Lines). This work is intended to be general in nature, so that product developers understand the broad approaches being used for this product line. The primary purpose is to provide developers with enough information and resource points of contact so they can target their activities towards the rapidly maturing Installation Product Line environments. User documentation for end users will be published at a later date.

This subject of this document relates to a series of investments, dating back to 1997, all focused on defining a coherent technical architecture that can be used for the development and fielding of applications related to military land management. This guide is intended for all developers creating capabilities for military installation land managers. This includes developers from within the Corps of Engineers research laboratories, from any of the service elements across the Department of Defense that provide data and tools relevant to military installation managers (and specifically military installation land managers), and to those in other agencies or in industry that are targeting capabilities relevant to this community of practice.

Approach

This conceptual work provides early initial guidelines that will allow land management software development teams to develop a next generation suite of integrated capabilities. Separate, but integrated, guidelines are provided for supervisors, principal investigators, and programmers. (This document cites web resources that help shape this maturing content area.) Engagement of a broader community of developers will allow developers to work towards a common target as they build or revise software tools and databases intended towards a common community of practice—the installation managers. This approach allows the development community to help shape approaches for this installation product line by introducing developer forums as part of the installation product line and land management suite. It is anticipated that the guidelines expanded in this work will mature within the next few years.

Mode of Technology Transfer

This report will be made accessible through the World Wide Web (WWW) at URL:

http://www.cecer.army.mil

and also through several additional linked web environments. The information will be further distributed through managers of technology programs that serve this community of practice across all the services. It is also anticipated that this information will be distributed in journal articles and other community of practice media.

2 History

In the past decade, efforts have been underway in the software industry and the government to develop and adopt enterprise software solutions that replace standalone software. Some of the key efforts and initiatives important to the development of a land management suite are discussed here.

Land Management System

This document relates to a series of investments, dating back to 1997, focused on defining a coherent technical architecture that can be used to develop and field applications related to military land management. In 1997, Dr. Ed Link, Corps of Engineers Directorate of Research and Development (CERD), HQUSACE, established a new initiative, called the Land Management System, or LMS. This initiative was focused on improved integration and interoperability of technologies, developed or adapted by the Corps of Engineers laboratories, relating to the management of lands and waterways. A report, entitled *Plans for the Land Management System (LMS) Initiative* (Goran, Holland et al. 1999) documented a plan (1999-2005) for developing this technical architecture and for testing specific applications at selected field sites.

In the process of developing the LMS plans, a team evolved a concept for the **Enterprise Technical Architecture (ETA)** that "expanded" beyond land management across multiple application areas. This concept was communicated back to CERD, and to the leadership of the newly emergent Center to which all Corps of Engineer Labs were assigned, the Engineer Research and Development Center (ERDC). Simply stated, while some of the requirements for improved integration and interoperability of technology were specific to the Land Management mission area, most of the involved information technology issues and approaches were common across all mission areas. Cost efficiencies could be realized by identifying the "common ground" foundational requirements, to be shared across all mission areas, from the efforts specific to each community of practice (e.g., such as military land managers). This recommendation led to a concept for the ETA, to be shared by the entire Corps of Engineers technology development community. The **Common Delivery Framework (CDF)*** emerged from this discussion as a key element of this enterprise technical architecture, providing a "framework" for technical tools and a linkage to technical standards. Using this common delivery framework, Product Lines would be created, relating to major communities of practice. These product lines would be driven by the requirements, legacy environments and specific guidelines of each community of practice, but all Product Lines would share a common information technology framework. After considerable discussion, Corps of Engineers headquarters proponents and the leadership of ERDC generally accepted this technical architecture concept.

The initial applications supported by LMS at military and civil field sites helped lay the foundation for the Enterprise Technical Architecture approach. In the meantime, the Corps of Engineers headquarters proponents for Civil Works research programs decided to help realize the full concept for this technical architecture by making a strategic investment (FY02-FY04) from the cross-cutting Geospatial R&D program, aimed especially at maturing the Common Delivery Framework in coordination with Enterprise Geographic Information System (GIS) approaches. This work is currently underway.

Also in 1997, a new Army Strategic Technology Objective (STO) was proposed and approved, relating to the development and integration of technologies to support sustained use and stewardship of military lands. It was called Sustainable Military Lands and Stewardship and covered efforts over the period from 1998 to 2002.

The key elements of this STO included: (1) technologies related to the interaction of protected species and the habitat for protected species on military lands and military mission activities, (2) technologies that helped military land managers understand the usage capacity of military lands, and could integrate with the Army's **Integrated Training Area Management (ITAM)** program, and (3) technologies related to the protection and restoration of lands impacted by mission activities, such as erosion control and revegetation approaches. While the primary purpose of this STO was to advance technologies in each of these three major areas, all of which in 1997 were identified as high priority Research

* Described more fully at URL: https://cdf.usace.army.mil

and Development requirements by Army proponents, another aspect of this STO was to investigate approaches for improved integration of these different models, databases, spreadsheets and other tools developed to support management requirements in each of these areas. This aspect involved the exploration of industry and other government approaches to technology integration, the evaluation of promising approaches, and the selection of one or more approaches to help achieve these integration objectives. The primary goal was (and still is) to facilitate the combined use of these differing technologies, since mission use, land restoration and species protection are all occurring on the same shared landscape. This integration effort was entitled Land Management Technologies Integration (LMTI).

LMTI allowed for analyzing integration needs and evaluation/testing of a number of different integration environments. An evaluation of potential integration environments was conducted by the LMTI team members as part of the LMTI process. Criteria for the evaluation included aspects such as an open environment, potential for partnership with other researchers, stability of the environment, potential for expansion of environment, and ability to work with various types of software and data used by land management scientists (Majerus 2000).

On completion of the evaluation, the **Dynamic Information Architecture System (DIAS)**, developed at Argonne National Laboratories, was chosen for more intensive analysis and development. DIAS is most useful for developing new simulation models, but also supports tight integration of legacy models. LMTI was designed to interact with the broader LMS initiative. As the Enterprise Technical Architecture concept matured, LMTI became a framework for developing the "product line" technical architecture for the military land management community of practice, using the foundation of the Common Delivery Framework (CDF).

DIAS was originally based on the Smalltalk programming language. The LMTI team partnered with Argonne National Laboratories to create a Java based application programmers interface to DIAS and all of its functionality. The DIAS API User's Guide, DIAS API Example Applications, DIAS Class Diagrams/JavaDocs—DIAS Model Integrator version, and the DIAS Class Diagrams/JavaDocs—DIAS Core Developer Version can be found at URL:

http://www.diasdocs.dis.anl.gov.

The timing for the LMTI initiative was somewhat "out of synch" because it was scheduled for a final year of effort in Fiscal Year 2002, while the Civil Works Geospatial funding for CDF just began in Fiscal Year 2002. In addition, another major Army Strategic Technical Objective (STO) was initiated in FY2002 entitled Fort Future. Fort Future's primary goal is to provide an integrated ability to simulate future states of installations. This capability was primarily focused on the Army Transformation issues, and understanding and shaping future facilities, infrastructure and training assets necessary to facilitate hosting transformed Army units on our military installations. Fort Future's timeline (FY02-06) allows only 1 year of overlap with the Sustainable Lands STO. Yet Fort Future provides the logical context for an integrating technical architecture for military installations technologies to be based on the foundational capabilities emerging through the Common Delivery Framework. In addition to contributing to the foundational capabilities of the CDF, the LMTI initiative resulted in the selection and preparation of a capability that supports tool interactions including feedback and exchanges among tools. This is a critical next step capability for Fort Future to provide simulations that consider the complex relationships of different installation activities under differing future scenarios.

Corps Enterprise Architecture

The Government Performance and Results Act (GPRA) and the Information Technology Management Reform Act (ITMRA), require that USACE adopt a Corporate Enterprise Architecture. The USACE Commanding General approved the planning process and architecture framework that make up the Architecture 2000 Plus, A2K+ (now known as Corps Enterprise Architecture, CEA) on 11 June 1998: "The Corps Enterprise Architecture is an integrated framework and process for evolving information resources to achieve [the] strategic goals [of the Corps]"* (http://www.usace.army.mil/ci/).

The Directorate of Corporate Information is the proponent of the Corps Enterprise Architecture (CEA), which is built on four pillars: *Business*, *Information*, *Application*, and *Technical* architecture views. The CEA lists goals and guidance for the development of information technology (IT) systems at URL:

https://cea.usace.army.mil

Presently, there is no single Corps Enterprise Architecture, but rather a strategy and guidelines for the development of Corps IT systems that have increased in-

* Quoted from URL: https://cea.usace.army.mil/index.cfm?fuseaction=home.content&itemid=143

teroperability, reduced development time, increased operational capability, minimized technical obsolescence, minimal training requirements, and minimized life-cycle costs. Design and development of IT systems must be firmly based on the business of the target user, the processes or analyses needed to do the business, the information available for performing the analyses, and the available hardware and software for supporting the system.

The Corporate Information Office has created five teams to help ensure Corps adoption of the CEA. The Investment Analysis Team (IAT) coordinates IT investments. The Architecture Alignment and Assessment Team (AAA) assesses the alignment of the IT investments with the CEA to ensure alignment with the Business, Information, Application, and Technical architectural views. The Information Assurance Assessment and Privacy (IAAP) Team reviews IT compliance with all assurance and privacy requirements. The USACE Cross-Functional Assessment Team (CFAT) is a management team that assesses the business value and risk of USACE-wide IT investments. Finally, the Life Cycle Management of Information Systems (LCMIS) Team ensures compliance with the life-cycle management requirements.

USACE Product Lines

In response to the goals of creating IT tools within the CEA framework in support of Corps' customer needs, the idea of building large software components that can be reused in support of the needs of different customers has gained support. These components can be assembled to create IT tools to support disparate business needs. For example, a visualization component and a hydrologic model component can be combined to create a flood management tool for a river operator and a training range management tool for a military installation. A research and development group having expertise in these components may develop both end user tools (in addition to similar tools for other customers) as part of their product line. Manufacturers of automobiles, tools, and appliances have product lines—machines designed for different customers, but composed of identical parts. "Product line" is also commonly used to identify a set of products developed for a particular customer. For example, a set of kitchen utensils, mixers, pots, pans, dishes, might be developed to be color coordinated and priced for a particular customer. The first use of "product line" is from the perspective of the developer and indicates efforts to reuse components for the purpose of building better and less costly products. The second use is from the perspective of the target user and suggests a set of tools that are designed to work together. "Product line" in this document is generally used in the sense of an integrated set of tools. However, often the development of an integrated set of tools is optimized through the reuse of components used to build those tools.

Fort Future

The Army is facing another round of Base Realignment And Closure (BRAC),* while simultaneously facing a significant transformation exercise. The final requirements will be submitted to the Commission by 15 May 2005. A Deputy Assistant Secretary of the Army for Infrastructure Analysis (DASA(IA)) has been established within the Office of the Assistant Secretary of the Army (Installations and Environment) to lead this effort (White 2002). That office will supervise a Total Army Basing Study (TABS) Group to conduct the necessary studies and analyses. In response to the BRAC and transformation challenges, ERDC has established an aggressive program, Fort Future, to develop simulation models of the operation of military installations to help installations respond rapidly and correctly to realignments.

ERDC envisions that Fort Future software will be used in a number of different ways, but the most visible expectation is how it will help with the BRAC and transformation initiatives. It is anticipated that Fort Future will play import roles in installation charrettes organized to identify promising solutions to restationing assignments. Solutions require use of existing and construction of new buildings and infrastructure within cantonment areas and across training and testing ranges. They require attention to competing goals that include training, housing, traffic, deployment, protection, welfare, and cost objectives. In support of planning charrettes, Fort Future developers will help generate rapid analyses of planning alternatives. Fort Future is also envisioned to be a catalyst in the development of integrated software products that will be used by military installation planning personnel to plan and manage installation lands and environment.

Fort Future will be a product line in the sense that the it will be a set of tightly integrated tools developed for military installation planners and managers. These tools will share common components such as human interaction and visualization software and common databases. There will be at least two target user

* Described in detail at URL: http://www.hqda.army.mil/acsimweb/brac/braco.htm

groups and, for each, a specific suite of tools and capabilities will be delivered (Figure 1).

The main thrusts of the Facilities Suite are to:

- **Design structures and facilities** on installations that meet the needs of the transformed forces.
- Analyze alternative uses of facilities to efficiently and effectively **project forces** to theaters across the world.
- Evaluate the ability of buildings and structures individually and together to **protect forces** from terrorist activities.

The main thrusts of the Land Management Suite are to assist installation planners:

- **Analyze the ability to train/test** now and sustainably into the future.
- Locate **new training/testing ranges.**
- Meet **habitat and threatened/endangered species** management objectives.
- Control **flooding and erosion.**
- Maintain **training realism.**

These two suites differ with respect to the spatial and temporal scale requirements for conducting analyses. The facilities suite deals primarily with the cantonment area, which is a tightly controlled and management environment. While buildings and roads have a long life-span, management goals seek to ensure that the basic state of the infrastructure is constant and therefore predictable. Hence, over its lifespan, the infrastructure is maintained within close parameters. The activities of greatest interest and variability are short-term and include deployment, emergencies, and responses to emergencies. Cantonment area simulations therefore involve relatively small areas (e.g., a few kilometers) and activities that happen rapidly and in very specific locations. The resolution of simulations can involve meters (or less) and time steps of seconds to minutes. In contrast, the training/testing area questions most often involve analysis of the dynamics of natural processes and changes in the patterns of human development and settlement outside of the installations. Analysis of these processes involves large expanses of area (e.g., 25-50 km) over several decades. Spatial resolution requirements are rarely smaller that dozens of meters and time steps can be weeks to months to years. Some analyses such as the ability to conduct a specific training exercise, simulate a storm event, or simulate the behavior of individual animals can require very small time steps and high spatial resolution.

These Fort Future suites not only share a common database and user interface, but they represent the real interactions among cantonment and training areas that, together, are affected by and affect surrounding lands and communities. Consider Figure 2, which captures some of the major cause-effect relationships that link installations with their surrounding communities and region. Ovals represent aspects of the installation (light) and surrounding region (dark). Arrows indicate that the aspect at the tail of the arrow affects the aspect at the head of the arrow. Some of the arrow lines are double headed, indicating that each affects the other.

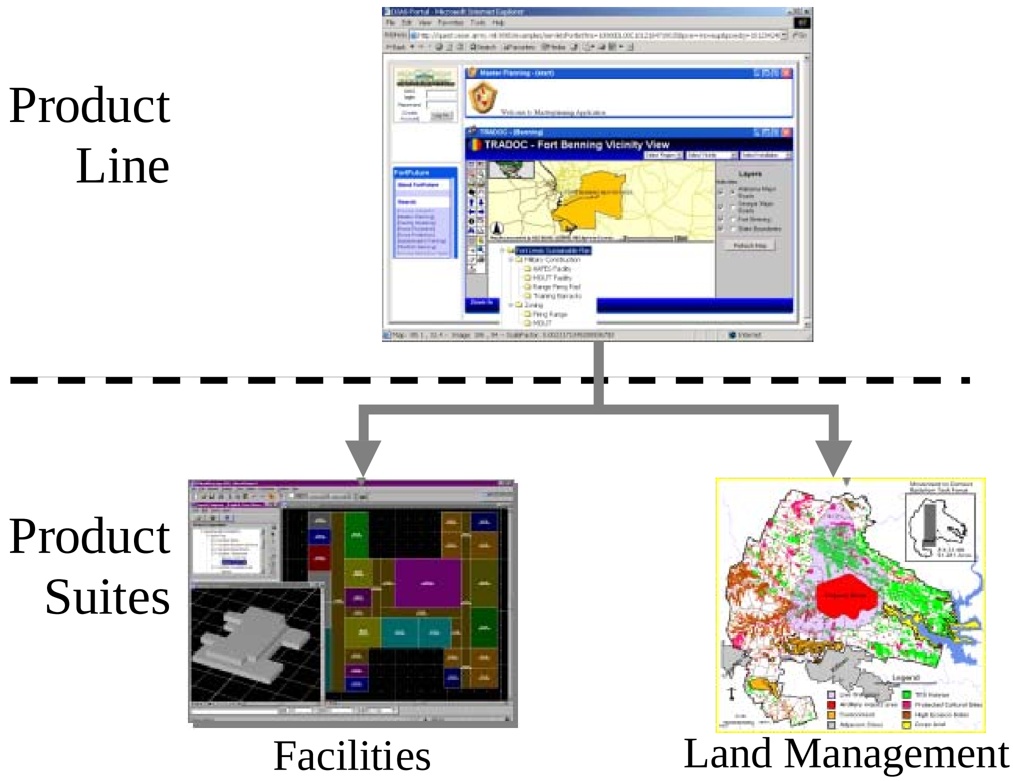


Figure 1. Fort Future product line and suites.

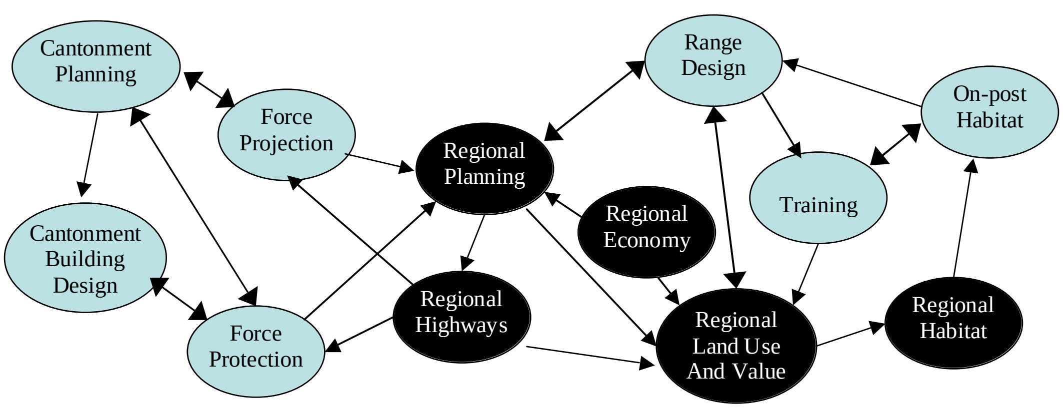


Figure 2. Interactions among installations and communities.

There are two clusters of installation-related aspects, with each interacting with regional aspects. The Fort Future product line will be composed of a product suite that reflects the two clusters of installation aspects. The Facilities Product Suite focuses on the leftmost cluster, considering force projection and protection issues with respect to the planning and design of the cantonment area and its buildings and facilities. The ability to project and protect forces is directly related to regional plans, and especially the regional highway, road, and railroad networks.

This document is concerned with the Land Management Suite, which focuses on the training, range design, and on-post habitat issues captured in the upper-right portion of Figure 2. To complete the Land Management Suite, regional issues represented by the black ovals in the figure must also be considered. The arrows actually capture the most interesting dynamics of the overall system, representing the simulation modeling required of Fort Future. Design plans developed later in this document recommend available models and needed efforts to develop the Land Management Suite.

Note that both suites share needs to integrate with regional planning, regional highways, expectations of changes in the regional economy, and regional land use patterns. The Land Management Suite will need to predict gross structures of land use to allow prediction of future limitations on the use of training/testing areas. Force protection will need more detailed information about the road and other networks, location of buildings, and details about the buildings.

In summary of the relevant history, there have been many important advances in the software development communities, including the Corps of Engineers, that now make it possible to begin developing simulation models of our installations and surrounding communities and regions to better plan for the sustainability of military installations and adaptation of them to support the Transformed Force. Recent ERDC efforts are culminating in the development of the Fort Future Product Line, which is composed of product suites to support analysis of cantonment areas and training/testing ranges to support changing missions. This document provides guidelines for the development of the Land Management Suite.

3 The Changing Face of Software Development

There is an ongoing revolution in software development that reflects continued demand for communication among software programs across the hardware infrastructure of the Internet. The software development industry is now providing increasingly powerful and sophisticated approaches that support business-to-business (B2B) interactions. Increasingly, these interactions are between software without direct human intervention. Sun computers once coined the phrase, “The Network Is The Computer” and that promise is now emerging as a reality. Corps software development must now embrace notions of enterprise data and computing. Execution of software now involves the runtime exchange of information among software programs and other Internet-accessible computers.

Corps Software Delivery Goals

The Corps of Engineers R&D community has successfully developed software over many decades. The goal has been to support the Army and DOD in their military and civil works missions. Reflecting academic disciplines, software development has been constrained with teams representing single disciplines. In support of land and water management, separate software products have been developed for hydrology, water quality, habitat analysis, land use analysis and prediction, training patterns, training suitability, noise attenuation across landscapes, and dust transmission. Additionally, operation of these models has typically relied on the skills of experts trained in the respective disciplines. Application of a set of models to evaluate the impact of alternative management decisions is typically very expensive and time consuming. The goals of Corps software development include the need to be:

1. Scientifically accurate
2. Easy to use
3. Reusable
4. Easy to update and maintain
5. Portable across all user platforms
6. Accessible by a wide range of users
7. Cost effective information technology (IT) with affordable economies of scale.

As computers have become more powerful, networks faster and more extensive, user interfaces and graphics more sophisticated, computer memory cheaper, and users better trained, the opportunities for addressing all of these goals has improved. The computer revolution is still just underway and approaches to software development change rapidly. A goal of this document is to help developers reconnect with the current state of the art in software development so that Corps software development will improve. In particular, this document addresses the goals of reusability, maintenance, portability, and accessibility of Corps software through the Enterprise Technical Architecture approach.

Evolution in the Deployment of Software

Development of software is now substantially different than traditional development efforts conducted by the Corps. To understand development within the new paradigm, it is helpful to first reflect on the historic and current processes for delivering software and then to identify how the Enterprise Technical Architecture approach differs. Computer programs developed for DOS and pre-DOS era computers focused on data analysis and contained code to read and write data as required (Figure 3). Programs in the figure are indicated by the bold-edged rectangles. Computers are represented as the large rectangles. Apple computers ushered in the era of graphical user interfaces (GUIs) and existing software was upgraded to provide a user interface that was compiled into the same program with the data input/output (I/O) and analysis software. With the development of many GUI APIs and the rapid development of increasingly sophisticated APIs it became appropriate to separate the user interface software from the analysis and data I/O programs and in many cases the GUI was run as a separate program. An ERDC example of this approach is the development of the Groundwater Modeling System (GMS), the Surfacewater Modeling System (SMS) and the Watershed Modeling System (WMS) (Holland 1998). Similarly, the GRASS geographic information system (Westervelt et al. 1992) separated the analysis and data I/O from the user interface, which allowed the development of a series of different interfaces and allowed GRASS programs to be used as analysis steps that could be combined with other Unix programs.

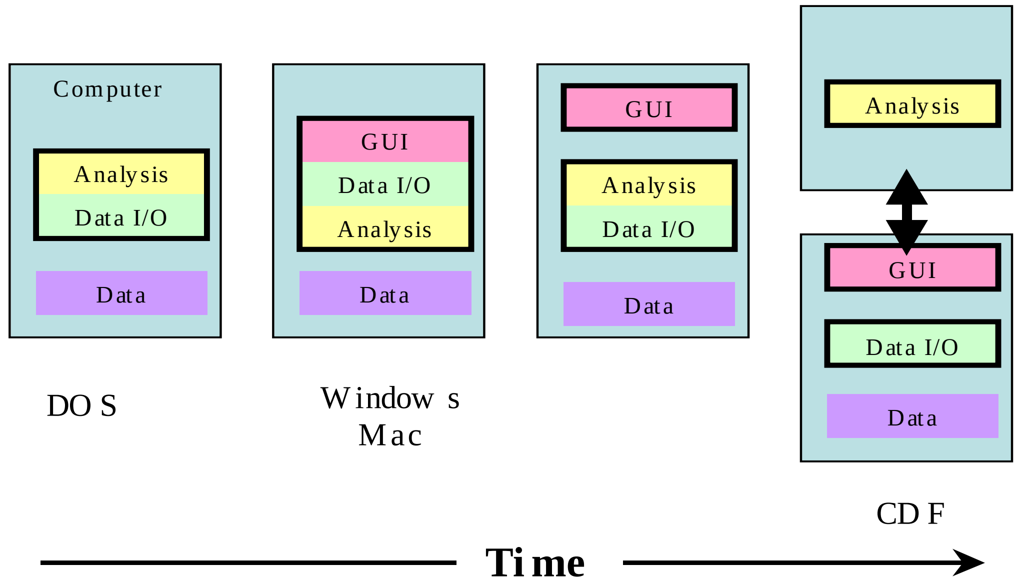


Figure 3. Historic software designs.

A current movement in the software industry is to place the more CPU intensive analysis programs on servers. These programs will accept instructions over the Internet and provide results to the caller. The end-user GUI is further separated from the analysis program (Figure 3), making it possible to run both on completely different systems. There are several important advantages to this approach. First, it becomes possible to deliver software capabilities without requiring the analysis programs to be ported and supported on different combinations of end user computers and operating systems. Second, the analysis service is "reusable" in the sense that multiple user interfaces can be developed. Third, and related, computer programs will be able to request analyses. However, information must be passed across the Internet, which is much slower than moving information within a single computer and also must be associated with significant security checks.

Figure 4 illustrates a future application might run in such an environment. A user invokes a "main" program on a local computer using a graphical user interface (center of the figure). Data are requested from a local program that communicates through Internet channels to remotely running services that send the required data. The main program then invokes, as appropriate, analysis services running on remote machines (left and top-right computers).

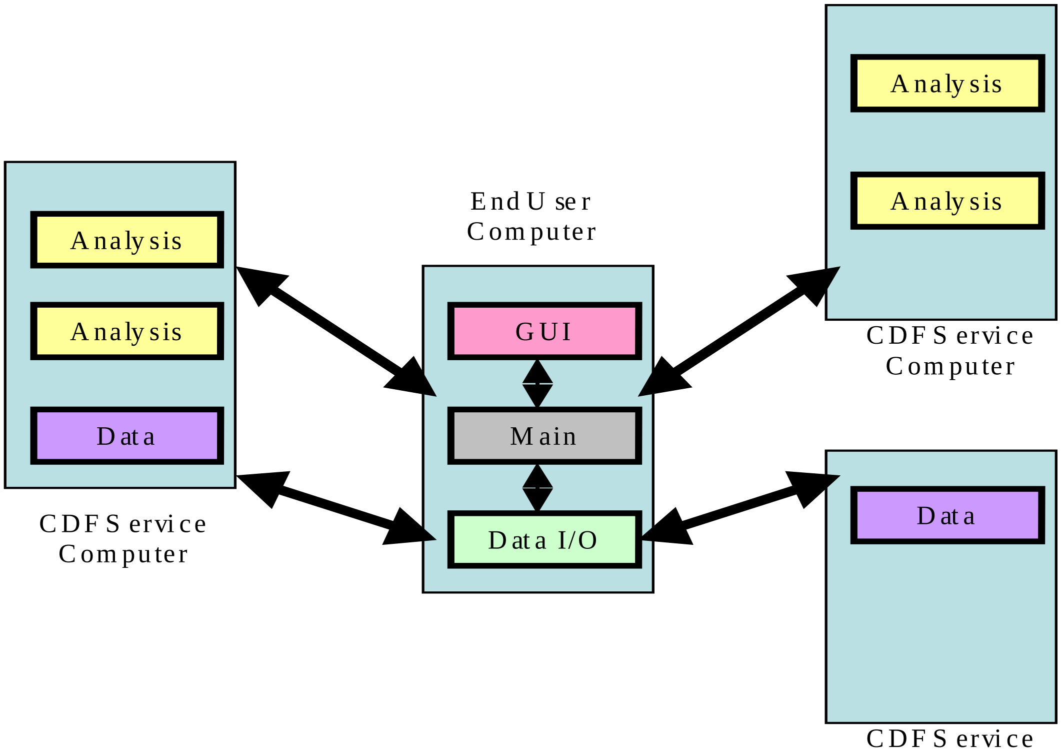


Figure 4. Delivery of Internet service software example.

There are several important benefits of this approach and a number of significant costs:

- Benefits:
  - Minimize porting and maintenance requirements of analysis software.
    Current software is often delivered to run under multiple operating systems on a wide variety of computers with different combinations of peripherals. To help alleviate the porting challenge, the Java programming language was developed and has attracted a substantial following. In response, Microsoft's dotNet (.Net) suite of capabilities includes a Java-like capability with additional features including the ability to program in a number of different languages and still generate code that can run across a variety of platforms. Another approach to alleviating the software porting challenge is to simply place code on servers that run the program in response to requests coming in off the network.
  - Increased security of code.
    Software operating as a service on a secure machine cannot be inspected, modified, or reverse engineered
  - Increased ability to link codes from many different disciplines.
    The advent of object-oriented programming allowed programmers to write self-contained modules in the form of objects that other programmers

could use through advertised methods. Traditionally the objects are still compiled together into a single executable program. Moving these objects to separate programs that run as Internet services now provides programmers with the ability to make object-oriented-like calls to the services without needing to access and compile the objects into the final software.

  - Analysis developers do not develop user interface software.
    Funds can be focused on the development of analysis tools and new science. It also means that software developed by multiple sources for the same customer need not have different interfaces.
- Costs
  - Existing codes must be stripped of user interface and data I/O software.
    Useful analysis software that is currently bundled with user interface and I/O software to read/write files must be separated from those codes. If the original code will be maintained, there will be a need to keep the new and old development efforts synchronized.
  - Code must be developed and attached to applications to read and write via Internet communications lines. This code provides the ability for local and remote programs to run the program by allowing operational requests, data, configuration instructions, and output to be communicated.
  - Developers must agree with other developers on the definitions and formats of exchanged data. For example, if "temperature" is a required input of a particular program it is necessary to know what scale is used, where and when is that temperature taken, how accurate must it be with respect to the measurement and the time and space in which it is taken, how long the temperature remains valid, whether it is an integrated or averaged value, etc. Is it provided as an ASCII value, as a field of values, and as a time-series? Data definitions become extremely important when integrating two or more software programs.
  - Relationships between developers and customers will change. In the past a software developer (individual or small team) worked directly with end users to provide a fully functioning software product. This product combined domain expertise, database and data manipulation capabilities, a user interface, and support. Now, those teams and individuals will focus on the development of application software that will be combined by another individual (or group) with other software capabilities to create end-user needs. Evolving to this approach will be challenging.
  - Creates a dependence on the network operating.
    These are Internet services that allow a client to request that a remote service run and return results. This requires that the client and the service be on the same network, the network connection be working between

client and service, and that the service be working on the target service computer.

- As customers make more use of remote computer hardware running Internet-based services, that customer base will be expected to support the purchase and maintenance of that hardware. This may be via central allocations or through charges to the customers.

Costs and benefits must be evaluated with respect to each software program considered for conversion to or development as an Internet-based service.

Data, Standards, and Current Systems Types of Data

To facilitate the use of modeling and simulation environments, simple and complex data types are necessary. Many military installations have used traditional Computer Aided Design and Drafting (CADD) and Geographic Information Systems (GIS) geospatial analysis tools. Geospatial data are:

> … information that identifies the geographic location and characteristics of natural or constructed features and boundaries on the earth. This information may be derived from, among other things, remote sensing, mapping, and surveying technologies. (Federal Geographic Data Committee 1994)

Tabular data, while not necessarily geospatial data, may be used in conjunction with geospatial data in decisionmaking.

Points, lines and areas within the computerized mapping programs represent the world. Topology, the geometric shape of the feature and its relationships, became important as the analysis potential grew in the user community. As technology and data sources expanded, additional data types such as aerial and satellite imagery became available.

The representation of the world has changed recently from a linear data model, points, lines and areas, to a more logical data model. This data model is composed of objects and represents features in a more realistic way. The main data types within the object world are familiar as they use the same terminology, but it is the representation of them within the world that has evolved. Vectors still may represent linear features. Raster data representing images, surfaces and thematic surfaces and are frequently acquired via remotely sensed techniques and are commonly used in modeling efforts. Triangulated irregular networks (TINs) represent surfaces with dimensions and address representations are available.

Data Standards

Functional area experts have developed various standards to facilitate the use of technology in decisionmaking. The Federal Geographic Data Committee in conjunction with public, private and government entities, developed the Content Standard for Digital Geospatial Metadata. This standard describes the metadata portion of geospatial data. The metadata portion contains information regarding the quality, attribution, history, spatial reference, and context of the geospatial data.

The CADD/GIS Technology Center of the U.S. Army Engineer Research and Development Center was formed to facilitate the development and usage of data standards for facilities, infrastructure, and environment. Products of the center include the Spatial Data Standard for Facilities, Infrastructure, and Environment (SDSFIE). This standard provides graphical and nongraphical content elements and a data structure for geographically referenced features. The A/E/C CADD Standard is a consolidation of existing CADD drafting standards. The facility management standard is designed to support facility management activities. Further information on these standards may be obtained from URL:

http://tsc.wes.army.mil.

In addition to broad standards such as those above, functional area suggestions are available in some domains. The Integrated Training Area Management function of the U.S. Army have developed a list of core and optional GIS data layers to facilitate the management of military training land. Data layers such as fire breaks, military GRID, impact areas, loading ramps, and MOAs are a few of the layers. For more information on the suggested layers and their responsible parties, see URL:

http://www.army-itam.com

Data Sources

Models and simulation environments need quality data to facilitate appropriate usage for decisionmaking. The capability to access different types of critical geospatial data is vital to modeling and simulation as well as to the command and control functions of the installation life cycle (Dilks et al. 2002). In response to the need for and accessibility of the necessary data, enterprise repository efforts are underway in all of the services.

Army

The Army's enterprise repository is called *GIS-R*,* which provides:

- A "one-stop" common repository for GIS activities that are dispersed at all levels.
- All aspects of geospatial data—the spatial, the metadata, and the attribute components—are included.
- GIS analysts who can spend more time providing decision support assistance and less time filling requests for data.
- A common, agreed-on framework for data storage, upload and download, so that less staff time is spent on designing solutions that are ad hoc.
- Connection to applications in a system based on a database management system—a more straightforward connection than in a system based on data read directly from files (Dilks et al. 2002).

Air Force

Geobase, the Air Force enterprise data repository, vision is "one installation one map" to attain, maintain and sustain one geospatial infostructure supporting all installation requirements. More information on *Geobase* can be found at URL:

http://www.il.hq.af.mil/geobase/

Navy

The Navy's readiness-driven data environment is called *GeoReadiness* and is comprised of theater and range information, shore infrastructure management and an environmental management planning tool (Baucom, presentation at Symposium 2002). *GeoReadiness* is coordinated with all other services enterprise efforts.

Data Availability

Geospatial data are readily available from many private, public and government entities. Unfortunately, these data sets rarely represent the land inside the fence boundary of military installations. To alleviate this void, many military installations have developed vast amounts of geospatial data. Unfortunately,

* For more information on GIS-R see URL: http://gisr.belvoir.army.mil.

the access, reliability and quality of the data are questionable by the utilization of less than optimal data stewardship practices.

Enterprise data efforts such as GIS-R and Geobase are designed to facilitate the sharing of military data for installation readiness. For them to succeed, the correct infrastructure must be obtained by the installation to include physical network, access policies, and responsibilities and security as examples. As the Department of Defense works to determine the policies on these issues, installations still struggle to put successful tools, such as data, into the hands of those who need it.

Current Systems

Software can take a long time to be accepted as an integral part of business processes, but over the past two decades many software products have become deeply integrated. Military land management offices are traditionally, but not universally, relatively slow to embrace new technology—relying instead on personal experiences with the land and users of the land. Nevertheless, Windows® based computers running standard office suites supporting word processing, simple data bases, e-mail, and spreadsheets have become commonplace. Geographic information systems (GIS) and Computer Aided Design (CAD) have become common in most offices, but not directly used by more than a few people. Significant investments in these systems and installation-specific data has resulted in a rich legacy of current and historic engineering and land use information. Many offices are now connected to the Internet and workers are expected to use it as their predecessors used the library. E-mail is the most relied-on Internet-based service that has become obligatory for most people.

Delivery of a software suite into military installation land planning offices must not disrupt the current successful use of software, must integrate with that software, and enable easy use of information already stored digitally—especially historic and planned land use and land cover information.

While there are many software products specifically development for military installation land planners and managers, few are in current use. Examples include IDLAMS, FHASM, PRISM, OO-IDLAMS, EDYS, BNOISE, SARNAM, ICBM, MAGIC, CASC2D, and others. Perhaps the most important reason that these, and other, systems are not used to their full potential is that funding to continually support the software either never became or never was available. Even software that is bug-free needs to be maintained as related software changes through upgrades and modifications. Some software addresses the need to plan military land use, but target users find little or no time to conduct such exercises in the face of continuous short-term needs. Such software has the added burden to be nearly self-operating to avoid significant user learning curves. Most software has been delivered in prototype form to installation offices by the developers who are able to provide convincing demonstrations after installation. By the time the intended user returns to run the program any number of causes can make the software inoperable. Instructions can be lost, files might be moved, operating systems upgraded, necessary data changed or moved, and interest lost. Finally, data necessary to run an analysis typically exists first outside of the database established for the analysis software. For example, a watershed model may need data that the user office maintains in a commercial GIS. To run the model, the data may need to be resampled, redefined, reprojected, and converted into a different format—steps that can be arduous and fraught with opportunities for error. Analysis models that have been most successful (e.g., BNOISE and SARNAM) have been adopted not by installation planning offices, but by a national office (CHPPM) that provides analysis services to installations.

The Land Management Suite will seek to deliver the capabilities of current and new ERDC land management and planning tools in a manner that allows easy integration with software already adopted by the offices and low cost to use the ERDC tools.

4 Designing a Land Management Suite

Next-generation software product suites are being developed to support a number of specific user communities, such as Civil Works Environmental Quality, Military Installation Managers, Facilities Designers, and Civil Works Navigation. The CEA pillars for developing future systems are Business, Information, Application, and Technical architecture views, and each is addressed here.

The **Business View** focuses on the need to support rapid analyses of the ability of military installation lands to support training and testing in the future. This dovetails with military installation land management, which employs many people working with well-defined business processes. The business functions involve analyses of alternative land use development plans, investments, operations, uses, and procedures with respect to:

- Sustainability
- Encroachment
- Habitat impact
- TES impact
- Hydrologic and erosion impacts.

Such analyses may be conducted by military installation land management personnel, Corps districts, Corps labs, or at the direction of installations or their regional Installation Management Office (IMA).

The **Information View** addresses the information that is available for address the business processes used to make decisions. This information provides the internally available information that will be analyzed and processed to help make informed business decisions. For military installation land management, the information includes current and historic information about the land use and land cover patterns, land characteristics, habitat suitability, training suitability, vegetation succession, potential for various natural disasters, regional urban settlement patterns and land use change drivers, and local weather and climate data. Historic, current, and planned training and testing activities provide the most critical human impact information. Timeliness, accuracy, currency, availability, accessibility, and security of the requisite information will be critical.

The **Application View** provides a description of the processes that will be applied to the information to generate results used to make business decisions. Processes that need to be applied to the available information include:

- Projection of land cover (vegetation succession)
- Analysis of land suitability for training/testing
- Analysis of land suitability as habitat for various species
- Analysis of watershed responses to sever storms
- Analysis of projected water quality/suitability for swimming and fishing
- Analysis of soil erosion and deposition.

The **Technology View** provides the hardware and software architectural pieces used to create and manage the systems. From the standpoint of the target user, the technology must support the following characteristics:

- All programs will have a consistent look and feel.
- All programs will access data from a common database (or set of databases).
- The system will be installable and maintainable via the Internet.
- There will be minimal system management requirements.
- The system will focus on evaluating the direct and indirect implications/consequences of alternative management decisions, plans, and investments.

The "building blocks" of the system (Figure 5) include the following:

- On the user's computer
  - Programs providing graphical user interface, main operation, and data input/output capabilities
  - Standard Web browsers
- At the user's site
  - Possible CDF compliant database input/output services
- On the Internet
  - Various CDF compliant database access, analysis, and display services.

The cornerstone of any computer system is the data on which the analyses are conducted. Therefore the design of the user's databases and access to that data is critical to the successful use of the system. Development must consider cost, utility, security, ease of use, and long-term supportability. Required data will already exist in various paper and computer readable forms and users will want to maintain those systems, requiring the development of automated processes to read and write that data from the new product line capabilities. Reading/writing data might be accomplished by writing software that allows an application to directly communicate with the legacy databases. Development of a CDF-compliant data service that reads/writes the legacy data will allow a consistency in the machine-to-machine communications that CEA affords.

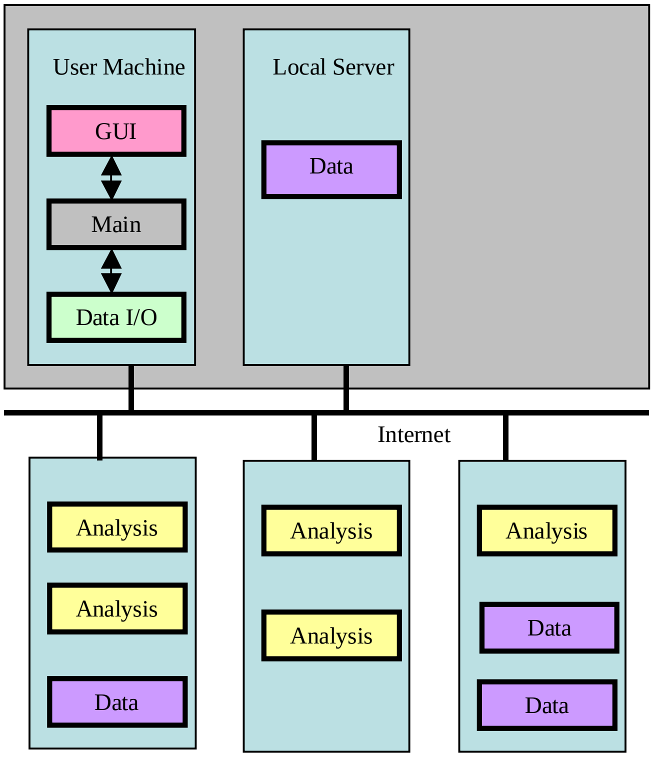


Figure 5. Combining desktop/local, and distributed services.

The following sections view the Land Management Suite from the standpoint of the end user, supervisors and program managers, project principal investigators, and software developers. While each group must understand the business, information, application, and technology viewpoints, each will tend to focus on different parts of these at varying levels of detail. Programmers will be, for example, much more knowledgeable about the technology and applications while supervisors and program managers will focus more on the business requirements.

5 The Land Management Suite—The Land Manager's Perspective

Military installation land managers include individuals at range control, environmental, and military unit offices. Offices exist at the installation, the regional, and the national levels. Collectively, these people are responsible for scheduling activities, protecting resources, rehabilitating over-used land, and conducting training/testing exercises. The Army Corps of Engineers has delivered a wide variety of software to these offices to support their land management activities. Examples of delivered software include:

- Commercial and government geographic information systems
- Hydrology models
- Dust and smoke simulation models
- Groundwater models
- Range carrying capacity models
- Range scheduling software
- Management of habitats
- Management of threatened species and their habitats
- Land condition trend analysis data and software
- Habitat succession models
- Prediction of training use intensity
- Prediction of current soil moisture
- Fire hazard and fire management
- Forest management
- Archeological site location and information
- Noise intensity prediction models.

Each piece of software has traditionally been developed as a standalone product that has been delivered to run on users local computers. As standalone products, each has unique data requirements, requires different data formats, has a different user interface, and has little or no ability to connect with other software. This situation is similar to the original delivery of now-standard office software. Word processors, spreadsheet, presentation, data management, mail, and other software were developed and delivered as separate products. Today we demand that our office software share a common look and feel, shares information, and can work together. The Corps of Engineer goal is to deliver product suites that, like modern commercial software, are integrated. Integration means that user communities will work through a common user interface that provides access to processes and analyses that share a common work and data space.

Entry into this system for the land manager will be through a user-managed portal that contains direct links to the desired information and analysis tools. The following sections describe available portals.

DOD/Service Level

Compare Mission Capacity—This module provides access to a database of installation capacities and current use of that capacity. Installation information includes the opportunity to extend capacities through the acquisition of property rights.

Map Browsing—Basic installation maps can be browsed that show historic, current, and future land use patterns on and around each installation.

Regional Level (IMA)

Current and Projected Training/Testing Range Health—This is a dynamically computed report that identifies the carrying capacity of each training/testing area and range along with the computed used and scheduled capacity.

Schedule Browsing—A view of the land use schedules that is automatically generated from the land management data base.

Map Browsing—Basic installation maps can be browsed that show historic, current, and future land use patterns on and around each installation.

Installation Level

Range planning and development

Construction of New Training Areas—This module will allow first-cut analyses for planning new training areas or changes to existing areas.

Habitat Management—Plans for managing installation lands will be evaluated with respect to habitat suitability and plant community succession.

TES Management—Predict the impacts of alternative management scenarios on the populations of specific TES.

Noise—Evaluation of training/testing plans with respect to the off-post noise patterns in time and space.

Smokes and Dust—Evaluation of training/testing plans with respect to impacts of smokes, obscurants, and dust on local human and natural populations.

RFI—Evaluation of training/testing plans with respect to impacts on the public licensed radio spectrum allocations and local use.

Range management

Training/Testing Area Schedule—This multi-year calendar will show the hour-by hour schedule for the training/testing areas in map and tabular form. With permission, users will be able to make requests and/or schedule areas and see the direct and indirect costs of using areas.

Training/Testing Area Current Condition—A red/amber/green map and table showing current and, optionally, future conditions of the training/testing areas with respect to training realism and environmental condition. Optional submaps include air and soil temperature; soil moisture; percent grass, shrub, and tree cover; and training limitations.

Analysis of Training/Testing Areas—The suitability of areas for specific training/testing events involves a cost-benefit analysis. Benefits include the efficiency and success in conducting the training and testing. Costs include the travel costs in time and fuel, potential for generation of complaints from neighbors, impact on future training/testing realism, and time for natural recovery of vegetation. In addition, certain constraints on training/testing can be identified.

Map Browsing—Installation current and historic digital maps provide substantive information on which decisions are based. Basic GIS tools that allow this information to be browsed and overlaid will be provided through this portal.

All of these capabilities will be available to the military installation land management community through a consistent user interface, and they will share a common enterprise database.

Training Area Planning Use-Case Scenario

To help focus development efforts, two use-case scenarios are developed—one here and one in the next session. First is a training-area planning scenario.

Scenario—The installation anticipates a change in training/testing needs over the course of the next 5 to 15 years caused by an anticipated re-stationing of units involving both the departure of current tenants and arrival of new ones. An analysis of the current training/testing areas and ranges indicates that some reconfiguration and some new construction are likely. The goal is to develop two or more likely solutions that may involve new construction, land acquisition, purchase of property rights, innovative scheduling, use of co-located federal lands, and other schemes.

Approach—To address the challenges of the scenario, suite users will be able to start with a user interface that provides the ability to edit training/testing area and range maps. Existing areas and ranges can be removed and new ones added. Annual schedules can then be developed to accommodate the anticipated needs of all projected installation tenants. These planning alternatives are stored for later editing and analysis. These alternatives can then be analyzed with respect to:

- The impact on the vegetation
- The impact of related noise, dust, and obscurants on nearby current and anticipated housing
- The impact of the training/testing on threatened/endangered species
- The impact on associated streams, rivers, lakes, and reservoirs
- The cost of conversion and construction
- The NEPA and environmental law requirements.

Once consequences of alternatives are analyzed, they are then available for consideration by planners. Availability comes in the form of reports, charts, diagrams, images, and movies that can be viewed and printed at the users site.

6 Guidelines for Supervisors and Program Managers

The Corps of Engineers funding processes provides the greatest challenge to achieving the goal of producing a truly integrated software product lines. Funding streams are broken into numerous military and civil works programs that separately fund arrays of projects. Thus far there is no requirement that ensures that any software developed will fit into a common system. Each funding program manager and staff supervisor is free to choose to encourage and support their funded projects and staff to deliver products that easily integrate with other software.

With respect to the software being developed for delivery to military installation land managers, supervisors and program managers are encouraged to be persistent in asking the following questions. These essentially seek to identify if the products are "LMS compliant":

- Is the look and feel consistent with the land management suite specifications?
- Is information needed to run the software being accessed from and stored in the common land management suite database? Could this come from CDF services?
- Are programs that analyze data being developed as CDF services? Are new capabilities mining CDF services?
- Are simulation programs making appropriate use of DIAS?
- Are CDF services/functionality being used where applicable?
- Are there modules of functionality that could be used by other product lines?

By seeking compliance with respect to these questions, supervisors and program managers will play a crucial role in the development of more useful software. Consider each of these questions in more detail.

Is the Look and Feel Consistent with the Land Management Suite Specifications?

Regardless of the underlying software and hardware and how integrated it really is, the presentation of the capabilities to the user through a user interface is perhaps the most critical aspect of "integrated" software. Regardless of what the user is asking the system to do, the user interface must be consistent to minimize the learning curve for the user. The land management suite will be developed using the Java GUI libraries and routines. Specific packages for presentation of results in graphical, mapped, tabular, movie, and text form will be identified as acceptable for development of the land management suite. Program managers and supervisors should expect to see demonstrations that reflect a common look and feel.

Is Information Needed To Run the Software Being Accessed from and Stored in the Common Land Management Suite Database?

This will not be transparent in demonstrations, but the need is critical for user acceptance. A common database allows users to supply information that the system will only request once—regardless of the program in the suite being accessed. *At the time of this writing, this crucial piece has not been designed.*

Are Programs that Analyze Data Being Developed as CDF Services?

Traditional software development that results in the porting of complex analyses to multiple hardware environments that use a variety of operating systems and software environments must, in large measure, be replaced by the development of Internet based services. End user programs will increasingly rely on the interaction of computer programs running on remote machines that provide a myriad services—including data access, manipulation, and presentation. Program managers and supervisors must encourage the development of these services and forsake the creation of cumbersome programs that must be ported and maintained for a variety of hardware and software environments. Services will be deployed using the Common Delivery Framework specifications.

Are Simulation Programs Making Appropriate Use of DIAS?

Simulation models must, at times, tightly link processes that have been understood and captured by scientists, engineers, and software developers in different computer languages and programs. If the modeled processes are not tightly linked, it is appropriate to run such programs in sequence. Processes that are tightly linked must be integrated into the same simulation model. The Dynamic Information Architecture System has been adopted as the integrating environment for such needs.

7 Guidelines for Principal Investigators

Principal investigators are responsible for the development of products. In most cases the PI manages the projects, which are completed using in-house domain and software experts or through contracts to other agencies, consulting firms, and universities. The PI must be able to respond to the questions that, where applicable, products are developed to be compliant with the installation product line land management suite. Questions that should be addressed to PI's by program managers and supervisors, repeated from above, include:

- Is the look and feel consistent with the land management suite specifications?
- Is information needed to run the software being accessed from and stored in the common land management suite database?
- Are programs that analyze data being developed as CDF services?
- Are simulation programs making appropriate use of DIAS?

To address these questions, PIs need to be familiar with more details about the development environments. They need to understand the software environments within which the Installation Product Line is being delivered before a detailed execution plan is created that will result in the development of a land management suite product. For the purpose of this document, there are four layers or tiers in LMS product line. The top tier is the user interface level, which is supported by the analysis-coordination tier. Users communicate to and receive results from the interface level. The dominating LMS requirement for compliance lies in the use of a common look and feel. The analysis-coordination tier is where the user inputs get translated into requests to execute particular software to retrieve, manipulate, and store data. Processes at this level are typically run on the user's computer. These requests are passed to the third tier, the data retrieval and analysis level, which generally involve the operation of models or analyses on remote computers. Finally, the fourth level provides the foundational software that supports the secure communications among software and users distributed across multiple computers.

The Inter-Process Communication Level

The data retrieval and analysis level depends on the ability to move information among files, computers, people, and other processes. The inter-process communication level provides the communication conduits and language that allows for information exchange. For the land management suite, the Common Delivery Framework (CDF) and the Dynamic Information Architecture System (DIAS) support inter-process communication. Each is discussed towards the end of this document. The Common Deliver Framework relies on software industry standards that have been adopted by many vendors, including Microsoft and Sun. CDF provides a mechanism that allows users and programs to remotely invoke a program running as a service on another computer on the Internet. Communication with a remote service is accomplished via XML based languages defined specifically for each service. CDF is very appropriate for processes that can be run without any run-time interaction with other programs.

The second inter-process communication environment adopted for the land management suite, DIAS, provides the ability for tightly integrating simulations that span a variety of legacy software programs. DIAS is a Java-based simulation-modeling environment that supports the development of object-oriented agent-based simulation models.

The Data Retrieval and Analysis Level

This level consists of services that are available across any number of computers. Centralized enterprise data warehouses provide access to much of the information required for viewing and conducting analyses. National scale data will likely be housed and managed at a single site, while regional and local data may be housed at IMA regional offices and installations. Information will be acquired from these locations via the Inter-Process Communication Level and dispatched to various analysis services. Analysis services are supported by analysis software residing on powerful centralized computers. These services may be accessible as CDF and/or DIAS modules.

The Analysis Coordination Level

This level is directly connected to the user interface level and runs on the end users machine. User requests for analysis, provided through the user interface, are conducted through the orchestration of the available Internet-accessible services and local processes. Data is collected from the user and from available enterprise data services, packaged for submitting to CDF services, and submitted to those services. On receiving results from those services, the information is repackaged for delivery to other services and/or provided to the user interface level for display.

The User Interface Level

The user interface level works with the land manager to identify desired information or analyses, retrieve required information, run analyses, and provide results in various forms depending on the needs and desires of the user.

Principal investigators must understand, conceptually, the role of these different levels and make sure that software developers are adopting the software development standards discussed in the next section.

Putting It All Together

The Land Management Suite will provide the installation and IMA land manager with a number of analysis options. Except for the user interface, all analysis operations will be performed on centralized land management suite operations (Figure 6). Management of the LMS software on the user machines will be minimal and completely automatic. The primary access to the LMS software will be via Internet Web browsers that connect to a single Internet IP address. A centralized interface server, the "LMS Request Processor" in the figure, will drive interactions with the user. This processor will accept data provided by the user and will facilitate the running of available simulation and analysis services that will probably be distributed on a variety of machines at a number of geographical locations. Data will be maintained on a centralized installation data server.

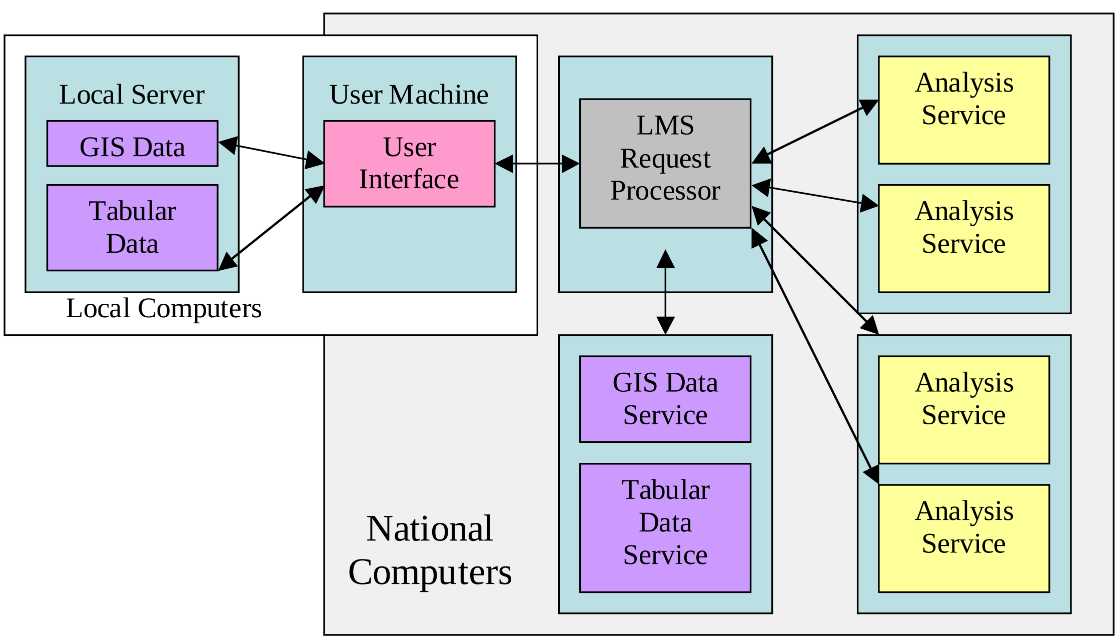


Figure 6. Land management suite gross architecture.

Work Efforts

To complete the Land Management Suite, the following efforts must be completed:

Develop Analysis Services

Analysis services are software programs that can be driven via requests made from a remote machine on the Internet. Services anticipated include:

- Training range suitability and carrying capacity model
- Habitat suitability models
- Threatened and endangered species simulation models
- Urban expansion model
- Vegetation succession model
- Training suitability model
- Urban dust complaint model
- Urban noise complaint model
- Urban RFI complaint model
- Urban light pollution
- Storm hydrology and erosion/sediment model.

Develop LMS National Installation GIS Database

Many of the analysis models require and generate spatially explicit input information. The information must be provided to the simulation models in very specific formats that contain specifically defined information. Historically, the vast majority of the time and effort spent on using spatially explicit simulation modeling has been spent on the acquisition and reformatting of digital maps. LMS will provide tools for accepting maps in a variety of formats and will cache the information for later user by the LMS simulation, analysis, and display tools. The efforts required include:

- Specification of a hardware/software environment for storing and retrieving spatial data
- Specification of the digital maps, data definitions, and data formats
- Initial population of national installation data
- Development of a user interface that allows uploading/downloading of digital maps
- Development of a service that allows access to the stored data.

Computer systems are primarily focused on the entry and management of information. File cabinets are still being replaced with computer files as we continue to embrace the "computer age." Any system must first embrace the need to easily store and retrieve information, even if the target end user does not want to personally be involved with this process. Product lines that will be built with reliance on CDF-compliant services will rely on the accumulation, storage, and management of accurate location-specific information.

In many cases the information will already be available in computer readable form and it will be important to avoid duplicating efforts to store and manage the information. In some cases data will need to be automatically ingested from existing sources. In others, the databases will be associated with services that will make the information available to a calling program at the time it is run. And, finally, it is likely that new data storage and retrieval processes will be built for the customer.

Existing data and databases include geographic information systems (GIS). The location of roads and streams are stored as a series of connected points, while locations of other objects such as wells and sample locations are stored as points. Large man-made objects and properties are delineated as series of connected points that begin and end at the same location. Such objects include buildings, parking lots, municipal boundaries, and other properties. Information characterizing any of these objects (point, line, or polygon) can be stored with the geographical locations. Finally, surfaces or fields of information are stored as raster files. These include digital elevation files, soil types, geology, plant cover, and any other landscape property that varies significantly over space. The product lines rely heavily on the state of landscape systems being managed and therefore on the ability to access and use GIS data.

GIS provides information about the state of a landscape, and commercially available software makes it possible to input, store, view, and access this information. When matched with knowledge of the behavior of the system—especially in response to management options—it becomes possible to project anticipated states of the system. The need to store, access, and manage system behavioral information opens a new demand for which there are no commercially available options.

Much useful data is stored in relational databases using powerful commercially available software. Datasets useful to the target product lines include personnel information, facility data, investment information, scientific measurements, financial data, and activity/use information. In addition to such historic information, it will be necessary to develop and capture knowledge of how the measured systems operate and behave. There are a number of commercial systems such as Stella, Powersim, and Extend that provide a mechanism for easily capturing system behaviors.

Four basic data types have been identified here:

1. System state (current and historic)
    a. Associated with geography and stored in GIS
    b. Stored in relational data bases
2. System dynamics
    a. Associated with objects on the landscape
    b. Associated with human systems.

Outputs of any product line analyses will fall into these types—especially future system state predictions.

Develop LMS National User Information Database

Because the user interface will be driven by a centralized Land Management Suite service, information collected from the user must be stored to avoided repeated requests to input information. Users will log into the suite and will have the ability to develop and store any number of "projects." Each project will be associated with information collected from the user that is necessary to address the user's questions. Users will be able to have information about their projects stored on the server for later retrieval.

Develop User Interface

User interfaces will be developed and will, of course, be the most visible component of the entire system—belying the software and data complexities underneath.

Principal Investigators will be responsible for the development, maintenance, and management of the Land Management Suite. Coordination with other PIs will be essential to ensure appropriate integration with LMS and other Fort Future suites. PIs should read and understand the "Guidelines for Software Developers" (the next Chapter).

8 Guidelines for Software Developers

Guidelines for software developers must take the form of pragmatic instructions that involve identification of specific standards. Therefore, this section identifies those standards that together define what it means to be compliant with the installation product line LMS.

The land management product suite will be based on interactions among five tiers: client, presentation, business, integration, and resources/services—as defined by the Java-2 Enterprise Edition J2EE specifications. Software performing services for these tiers will certainly be distributed among a variety of computers—possibly at a number of different locations. This tiered approach optimizes the opportunity to share and reuse software within the suite. Description of each tier follows using examples representing processes required by the land management suite.

Client Tier

- Presents information to: the end user of the software.
- Gets information from: the Presentation Tier.
- Software: Web browser.

The client tier is the software with which the end user directly interacts on their local computer, which is, necessarily, connected to the Internet. Interactions with the user must involve short learning curves, require little, if any, user installation of software, and be useful to infrequent users. Installation of any browser helper applications must be automatic, requiring only user permissions. Automatic tests of user hardware/software environments must be made to help identify whether minimal configurations are available. From the viewpoint of the end user, they must be doing little more than interacting with their favorite standard browser.

Presentation Tier

- Presents information to: the user's web browser and associated extensions and helper applications.
- Gets information from: the Business Tier.
- Software: Java Server Pages (JSP), Java Servlets, XML, and/or XSL.

Table 2. Presentation tier service examples.

User	Activity	Results
Environmental office	Input natural resource management plans	Stored multi-year plan
	Evaluate natural resource management plans	Long-term analysis of plans with respect to urban encroachment, plant succession, and habitat suitability
Range control office	Input training area schedule	Stored schedule
	Daily evaluation of training area suitability	Analysis of schedule with respect to current and predicted conditions
Training/testing area management office	Input plans for range construction, modification, and use	Stored plans
	Analysis of plans	Long-term analysis of plans with respect to urban encroachment, plant succession, and habitat suitability

The presentation tier generates the display instructions, typically in HTML, that are forwarded to the user's machine and web browser. Sets of programs in this tier provide specific notional analyses that match user conceptions of the work or analyses required. Users can do three types of things:

1. **Browse the database**, which consists of basic data, user modified data, and results of user requested analyses.
2. **Input plans, or courses of action (COA)**, which capture proposed schedules for construction, land management, training use, etc.
3. **Analyze the alternatives**, which involves predicting the consequences of proposed plans by running various analyses and simulations and then visualizing the results (Table 2).

The sets of programs translate between the human requests and the software that does the actual analyses. Requests are turned into invocations of certain software to do the analyses and analysis outputs are packaged for easy human understanding. The presentation tier is not open-ended, accepting natural language. It provides only the opportunity for a user to provide specific requests and input to those requests.

Business Tier

- Presents information to: the Presentation Tier.
- Gets information from: the Integration Tier.
- Software: Session Enterprise Java Beans (EJB).

The business tier is the central focus of an enterprise system as it captures the logic of the human system for which the enterprise system is developed. In a DOS or Unix command-line interface, the business tier is the set of programs available for the user to package into shell scripts (batch files). These programs are reusable and can be combined to address an almost infinite variety of needs. In Unix, standard programs include file search (grep), file sorts, file lists, arithmetic, logic, file parsing (awk), data presentation in tables and graphs, etc. GIS software provides another set of basic operations that allow data to be managed, manipulated, selected, displayed, and combined in many ways. Using these programs, shell script programmers can develop use-specific applications.

Similarly, the land management suite will offer GIS functions and simulation capability functions that will be assembled to address particular land management needs to.

- Store and access data and objects
- Convert data (e.g., re-project a map)
- Prepare for a simulation (e.g., prepare input and configuration files)
- Execute a simulation
  - Habitat
  - Training
  - Noise
  - RFI
  - Wind
- Prepare a graph
  - Scatterplot, trend
- Prepare a movie from a series of images (maps)
- Conduct statistical analyses
- Validate access permissions
- Get lists of objects
- Get metadata.

The business tier does not fulfill these services, but rather coordinates integration tier services to fulfill the requests.

Integration Tier

- Presents information to: the Business Tier.
- Gets information from: the Resources/Services Tier.
- Software: Entity Enterprise Java Beans (EJB).

This tier is responsible for taking data and service requests and arranging to have those requests fulfilled using specific databases and services. Table 3. lists examples of requests from the business tier and the related specific requests to the resources/services tier.

Table 3. Sample conversions of requests by the integration tier.

Business Tier Request	Request to Resources/Services Tier
Get a digital elevation model	Open data base x on machine y using software z and return the data bounded by the coordinates (x1,y1,x2,y2) in format f, projection p, and coordinate system c.
Get training carrying capacity	Run a training and vegetation simulation model to identify the maximum training possible that sustains the desired vegetation/soil.
Get training carrying capacity	Query database db and retrieve carrying capacity cc for training area t.
Run a remote simulation model	Take input data set, run a simulation on that data, and return the results.

Based on business tier requests, the integration tier will be retrieving data from the resources/services tier, submitting that data for processing, and returning the results to the business tier for presentation to a user and/or to the resources/services tier for storage. Some of the processes will involve submitting information to remote CDF (i.e., SOAP) services.

Resources/Services Tier

- Presents information to: the Integration Tier.
- Gets information from: data bases.
- Software: EJB and Java Data Base Connection (JDBC).

The resources/services tier deals with the specific data access and data analysis needs to fulfill the requests from the integration tier. Only at this level does the software connect specifically to models (like LEAM, BNOISE, SARNAM, GSSHA) and data such as GIS data from an ESRI system or tabular data from a particular DBMS. Only at this level are names of files and databases found in the software.

Many software capabilities will be established as services on the Internet that will be accessed as needed. This approach reduces the need to port, distribute, and maintain software on assorted hardware, software, and operating system combinations. The LMS standard for implementing such services is the **Simple Object Access Protocol (SOAP)**, an industry standard supported by many software vendors, including Microsoft. The Corps of Engineers implementation of SOAP is called the Common Delivery Framework (CDF), which is discussed in a section below.

Some services will be available as CDF (i.e., SOAP) services running on remote machines. Land management suite developers will be involved both in using already available CDF services as well as developing new services. Anticipated CDF services developed for the suite include:

- Urban growth modeling (e.g., LEAM)
- Habitat suitability analyses (e.g., HSI)
- Vegetation growth and succession models (e.g., EDYS)
- The SARNAM and BNOISE models
- Dust and smokes models.

Any tightly linked agent-based simulation modeling will be accomplished with the Java-based **Dynamic Information Architecture System (DIAS)—** running as a remote process. DIAS is discussed in a section below. DIAS can use CDF services and DIAS software can be implemented as CDF services.

9 The Common Delivery Framework

Developing CDF Services

CDF Services

Software, in general, consists of the following parts/functions: user control interface, data processing, input data, output, and visualization. Of these, CDF is primarily designed to provide remote data processing and data access. One conceptual view of the CDF approach is diagramed in Figure 5. The "Analysis" and "Data" services (residing here on remote machines) are CDF services that have been advertised as available for building end-user applications. Each accepts analysis requests consisting of all input requirements and returns outputs. The details of the format of the data streams and the formats and definitions of the tables, maps, lists, and configuration information are specific to each service, but well defined. At the time this document was prepared, it was anticipated that XML based data streams and perhaps HDF5 files would be adopted for moving information to and from the services. Many of the first services are to be based on legacy software and will be developed in a manner that requires minimal (if any change) to these codes. Because these codes read information from local files, information transmitted to a remote service will be used to create these files before the analysis program is invoked. On termination of the program, output left behind will be packaged into a data stream that is returned to the client program and the transient input and output files will be deleted. Brand new CDF services are more likely to combine the service request software with the analysis software and require no management of local files.

Services envisioned for development and addition to the ERDC CDF Services Library include, but not limited to:

- Data Access
  - Raster/Vector GIS
  - Tables
  - Financial data
  - Spread sheets
  - Movies
  - Images
  - Tables

  - Graphs
  - Reports
- Data Processing
  - Hydrology (surface, groundwater, channel)
  - Water quality
  - Vegetation growth and succession
  - Habitat suitability analysis
  - Urban growth
  - Population and individual animal behavior
  - Water quality (chemicals, temp, sediment)
  - Dust
  - Weather/climate
  - Noise
  - Radio interference
  - Laser
- User Run-Time Control
  - Land management decisions
  - River management decisions
  - Run-time data probing
- Visualization.

Developing a CDF Service

Legacy software is code that is already written and working and often has a user community and a history of successful application. These codes are extremely important and must be considered for being recast as CDF services. These steps will help guide a development team to the successful deployment of a CDF service based on legacy software:

1. Determine if the software should be recast as a CDF service.
2. Register "Intent to Develop" with the CDF Service Registry.
3. Develop the service.
4. Develop a client application to consume the service.
5. Register the service.
6. Conduct APL.
7. Beta tests conducted.
8. Register approved CDF Service.
9. Move to Operational Platform.

Each of these steps is discussed below.

1) Determine if the software should be cast as a CDF service

Not all software need be or should be recast as a CDF service as the costs are substantial, involving software development, creation of parallel development tracks, lots of coordination, approval steps, and maintenance of the service on servers. Questions that should be discussed before committing to the porting of working software into a CDF service include the following:

- Modularize the software by defining subcomponents such as Input, Output, Visualize, Convert coordinates, Calculate wave frequency, etc.
- Search for existing CDF services* that provide the information/functionality required by the subcomponents. If the service exists, simply make a programmatic call to the service instead of developing new code.
- Consider the reuse potential. If no existing CDF service is located, determine if any of the subcomponents of the software could be reused? For example, if the data input component of the software being developed requires data that is used by other software, then the data input portion of the software is a candidate to be developed as a CDF service. If there is little to no reuse potential, then developing a CDF service would not be prudent.
- Consider the time and cost savings. Will developing this software as a service decrease the time and cost of distribution and maintenance? Does the cost of maintenance include support for a variety of operating systems, hardware and software configurations? If so, a CDF service will provide a more time- and cost-effective solution.
- Does the software require computational resources not generally available at user sites? If so, developing a CDF service which runs in a supercomputing environment will provide additional functionality to the user community.
- Consider the granularity of the software. Does the software justify the overhead of a web service?
- At user sites, is the software naturally part of a set of software? That is, does the user use the code in conjunction with other programs, moving data and information among them? If so, then it is important to connect these programs to provide an efficient operation of them as a package.
- Does the software require computational resources not generally available at user sites? If so, turning it into a service running on a powerful computer will be beneficial.

* Discussed in detail at URL: https://cdf.usace.army.mil

- Is the software associated with a database that is not easy to maintain on each customer's machine? If so, a centralized database with an associated data server can be effective.
- Do target users have a variety of hardware and software configurations—including many different operating systems? If so, time spent porting a service to these various configurations can be saved by developing them into a CDF service.
- How large is the user community? The larger the community, the more expensive the process of distributing and supporting software designed to run on user machines and the more attractive it is to develop a service.
- How intense is the interaction between the software and the user? If low, providing the software as a service is attractive. If high, then it is better to run the software locally. An example might be a flight simulator.

2) Register "Intent to Develop" with CDF Service Tracking System

Once the decision has been made to develop a new CDF service, it is necessary to formally communicate with the CDF community via a CDF service tracking system, and throughout development and release of software. This tracking system is updated to reflect changes in development or availability of services and is also used to collect feedback from users. At this point, a new record is created on the system and is provided with the following facts:

- Proposed name of the service
- Short description of the service
- List of anticipated inputs and outputs
- Developing organization
- Point of contact
- Anticipated release date.

3) Develop the service

At this point, developers are ready to build the service, following guidelines in "How to Design a CDF Service."* This effort involves the following steps:

1. Remove any run-time user I/O.

   Many applications are already associated with user I/O codes compiled with the application code. This code must be removed and replaced, if required, with software that acquires the information in other ways.

* Source: https://cdf.usace.army.mil

2. Design approach for acquiring inputs.

   Many programs get information from several sources including files, input streams, shared memory, and web services. If an existing program will be retired in favor of using the planned CDF service, the opportunities for optimizing the I/O can be significantly greater. Developers that intend to minimize the maintenance cost of keeping a new service and an existing program synchronized with each other will want to minimize the changes necessary to create the service. As a consequence, it may be desirable to feed information to the service through files named, located, and structured identically to those used by the original application. In this case, the CDF plug-in may be developed as a separate program that turns the information supplied via CDF XML streams into the required files and then invoke the service program. On completion, the plug-in converts information stored in files into CDF XML for transmission to the calling program and then removes all temporary input and output files.

3. Design approach for providing outputs.

   Similarly, a CDF service will generate results that will be returned to the calling client and the form and format of that output must be defined. Outputs can vary from a single or a small set of values to large GIS data layers, tables, text, movies, and other types of data. CDF XML approaches must be defined and designed.

The input and output to CDF services are accomplished on the server side via a CDF plug-in. All plug-ins adopt and use the same underlying software approaches for facilitating Internet-based services including SOAP, JSP, and UDDI technologies. The application unique functions involve accepting input information and generating output information—generally via XML. The plug-in may be directly compiled with the analysis program that provides the information processing service, or it may invoke a standalone analysis program. The second scenario can be preferable in cases where it is reasonable to maintain the analysis program as a standalone program that can be used directly, or via CDF. This avoids the need to maintain two versions. In this case, the plug-in will probably generate temporary files that will be read by the analysis program, invoke the program, read program outputs (and relay that information to the client), and finally clean-up all temporary files before exiting.

In addition to the server-side plug-in, a test client must be developed that provides sufficient flexibility to thoroughly test the new service. This software must be developed in a manner that allows it to be released to future client developers. These developers will use the software to test the service and to build new clients.

Table 4. Typical provider information.

Field	Description
Organization Name	Name of the organization providing this service.
Contact Name	Contact person for this service-the representative for all services provided by the organization.
Phone	Phone number of the contact person.
Email	Email address of the contact person.
Discovery URL	Website-homepage of the organization.
	Service description information includes:
Service Name	Name of the service.
Service Description	A brief description that explains what the service provides.
WSDL URL	URL location of the WSDL file that describes this service.
Binding/Router URL*	URL location of this service. (SOAP address tag of WSDL file)
Service Info/Help URL	Overview-homepage with specific information about this service (example: Descriptions of input and outputs to
Client Code Download	Example client code that calls the service, or a small code library that can help end users call the service
Category	Grouping of similar services, such as visualization, database, portal, etc. (pick list provided).

4) Develop a client application to consume the service

To test the service and to provide future service users with sample client code, developers create a client application that will exercise all available capabilities of the service.

5) Register the service

Registration requires both service provider and service description information. Table 4 shows provider information.

6) Conduct Alpha Test

Identify that the software is in alpha testing with the CDF service registry. For the alpha test, the service must be established on an Internet-accessible machine, instructions for using the service must be published, and those instructions along with test client programs and software for developing new clients must be distributed to the test community. Test experiences must be collected and problems with the clients and server must be addressed.

7) Conduct Beta Test

Developers identify that the new service is ready for beta testing on the CDF Service Tracing System. The beta test is conducted using individuals representing the target user community. They are provided with all information necessary to develop their own clients. Problems encountered are appropriately addressed.

8) Register approved CDF Service

Once the software successfully passes the beta release goals it is registered on the tracking system as complete. Client-side software is made available along with documentation via the CDF website.

9) Move to Operational Platform

The final released service is installed on a machine that will be accessible to the user community. Developers continue to support and maintain the CDF service offering updates and new releases as required.

A CDF Example

Urbanizing areas around military installations generate pressure to reduce on-installation training and testing, creating challenges to the military mission. Pressures are associated with the desire of neighborhoods to reduce dust, noise, and interference with radios and television. Also, as the urban areas develop, critical wildlife habitat is lost off-installation resulting in citizen pressure to reduce training to protect the remaining regional habitats. Lights from neighborhoods and from commercial air traffic reduce the dark areas where soldiers can train with night-vision gear. Controlling and guiding urban growth through judicious use of property right exchanges, zoning, ownership changes, placement of major highways and municipal investments can help ameliorate future challenges to military installation training and testing.

A number of models already exist and can readily be combined through CDF. From a land use and military installation planner's perspective, the desire is to evaluate the future risks to military installation training and testing with respect to alternative application of urban development controls. Inputs to the system will include:

- Zoning maps
- Locations of wildlife areas, parks, forests, and other natural areas

- Routing of future limited-access highways and location of access ramps
- Population and economic projections.

The primary output of an analysis will be a time series of maps showing changes to the extent of suitable on-installation areas to train and test. Separate output maps will include:

- Potential areas where generating blast noise will be constrained
- Potential areas where tracked vehicle maneuvers might be constrained
- Potential areas where night vision training will be constrained.

Between the inputs and outputs is a multidisciplinary challenge to combine the predictive and analysis capabilities of many different models and analyses. A traditional approach would be to organize these models within a single system that would be installed and maintained on a user's computer. This creates a number of serious challenges. First, the programs must be compiled for each of the target end user machines. They must be tested and verified for each machine. Second, they must be downloaded and maintained on each machine—generally by the end user. With the CDF approach, development teams associated with each model establish an Internet-based Web service for model utilization. The end user never needs to be concerned with, nor know about these services. The end-user application developer uses these services through subroutines and method calls within the end-user application. The steps in the end user application will be:

1. Collect user inputs and requests.
2. Gather information about the area being simulated.
3. Run an urban growth model.
4. Submit future urban patterns for analysis to the following:
    a. Dust generation analysis
    b. Light-free zone analysis
    c. Noise-free zone analysis
    d. Habitat suitability analysis
    e. Training suitability analysis.

This is a linear process requiring no feedback loops among the CDF services—a key to successful CDF applications. A single end-user software program is developed and assembles the appropriate input information to make calls out to the available CDF services. The application developer will not be concerned with how the requests are processed, but only with the input requirements and the available outputs. Behind the scenes, the analyses and simulation services may be arrayed on a variety of computers on the Internet and geographically located virtually anywhere (Figure 7).

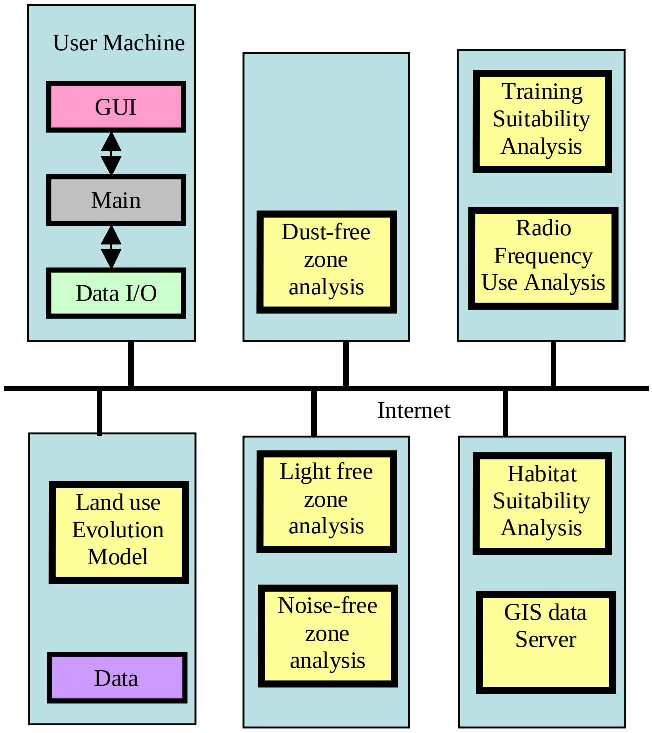


Figure 7. Software plan for the sustainability, encroachment, and room to maneuver program.

10 The Dynamic Information Architecture System

Tightly Coupled Simulation Modeling Needs

The CEA approach is perfect for providing Web-based services that run analyses for clients. A client process can orchestrate a number of services to perform an integrated, multidisciplinary analysis. In cases where very tight interactions among the simulated processes are required, another approach is necessary. Tight interactions are necessary when there are important feedback loops in the system being modeled. Figure 8 represents a variety of objects that include system state initialization objects, simulation objects, a system state storage object, a run-time visualization object, and a user control object. The arrows indicate the flow of information that is required. While most of the flows are fed forward, notice the loop involving ecology, agronomic, and hydrologic simulation. In this situation, the simulation models must be run simultaneously. CDF, as designed at the time of this writing requires that CDF services accept inputs, run to completion and provide outputs. It is possible to make a service request that runs for a single time step or for a thousand. Therefore, an integrated effort might package information for a hydrologic service to run for a month time step with the results packaged for a 1-month ecological simulation run and those results sent to an agronomic service for a month run. Those results get sent to the hydrologic service and so on until the years or more of simulations are accomplished. An alternate approach runs all simulations simultaneously with information passed among the components during the run as required. The Dynamic Information Architecture System (DIAS) (Sydelko, Majerus et al. 1999; Campbell and Hummel 2002) supports simulations run in this manner. Simulation objects are compiled together into a single program. Those objects can be associated with externally running legacy models that support the notion of running a discrete number of steps at a time without closing down the program.

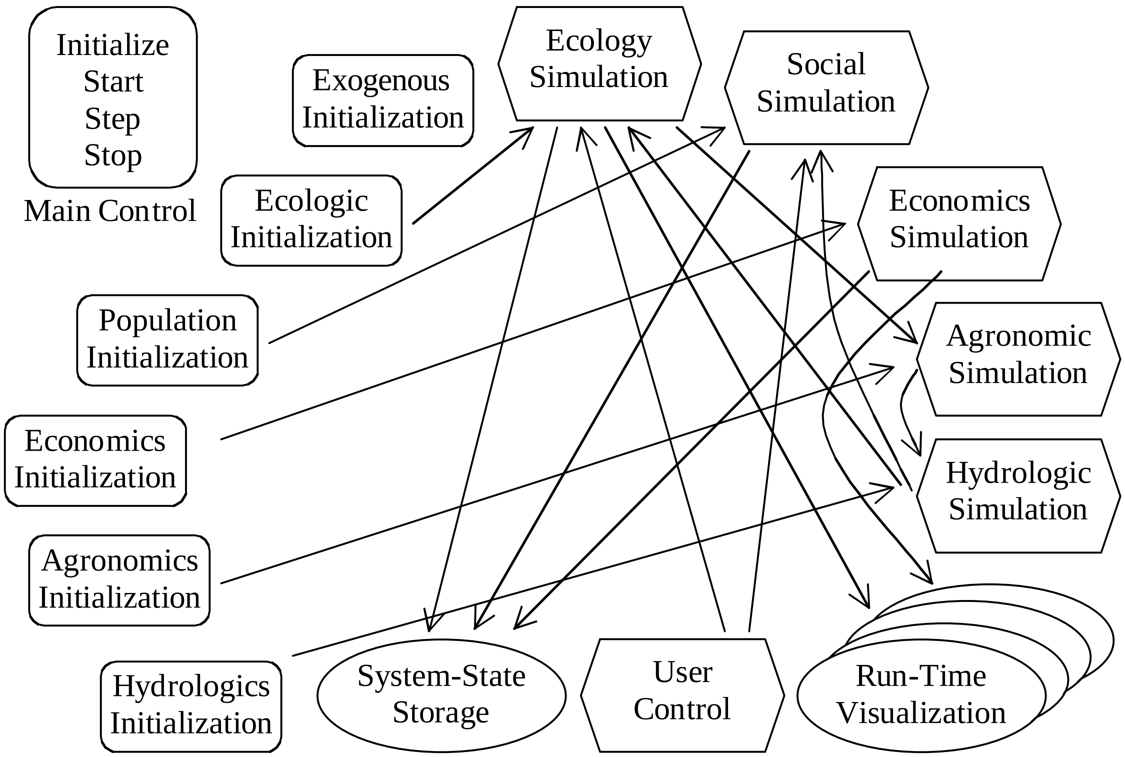


Figure 8. A services centric view.

There is a complexity in the design of a system reflected in Figure 8. Any individual object might need to communicate with any other object to retrieve information about the overall system state. This approach to object development can be efficient, but the resulting objects can rely too heavily on the availability of the other objects. A system-state centric approach can ameliorate this problem and allow any number of objects to be created that know nothing of any other objects, yet be able to work together in concert to simulate a system through time (see Figure 9). Here, every module that provides and/or uses system state information is connected to the system state warehouse, and any exchange or sharing of information among modules is through this warehouse. The entire system works in concert through a main control module that is responsible for initializing, starting, stepping, running, and stopping a simulation. This involves communications connecting the main control and all other modules.

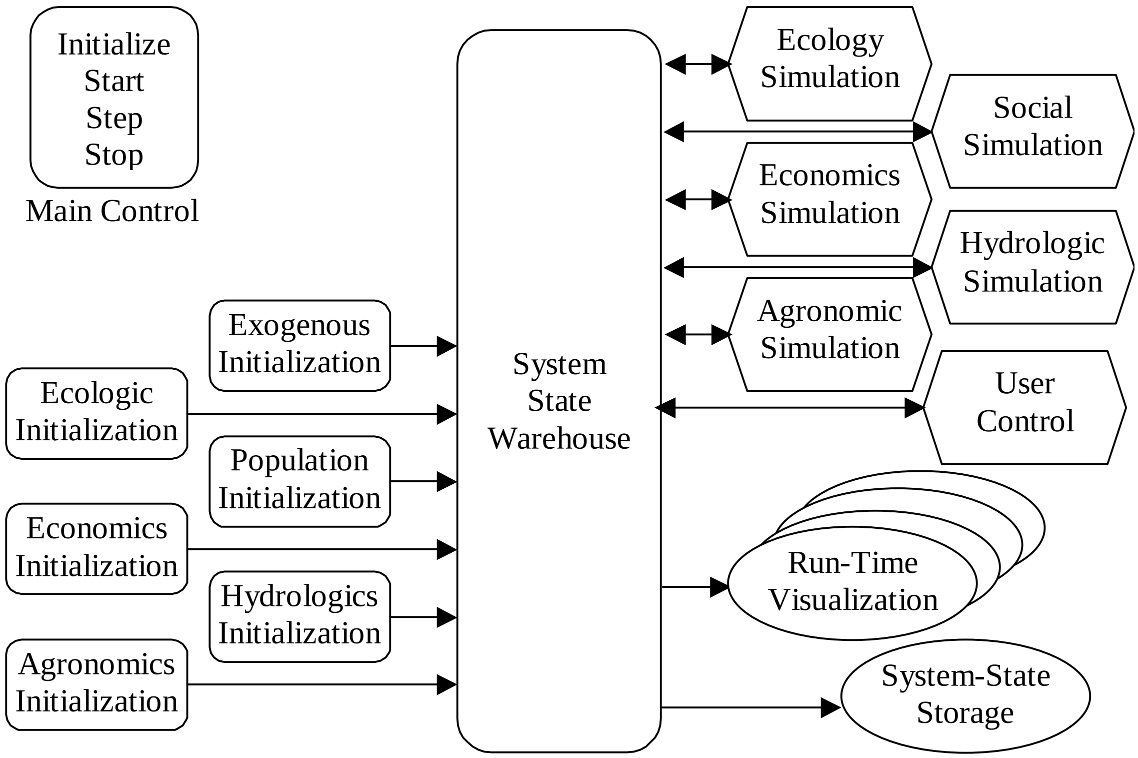


Figure 9. A system state centric approach.

There are five types of modules depicted in Figure 9. The "Main Control" module is responsible for the gross operation of the system, based on user commands and inputs, and controls the order of execution of the simulation modules during a simulation run. The "System State Warehouse" module in the center provides the communications center that allows all of the modules to indirectly communicate information with one another. It is responsible for caching all shared system state information and for providing that information as requested during simulation model runs. System state objects within the Warehouse will have the ability to accept and deliver data in different formats and with some simple transformations. For example, a vector map of Fahrenheit temperatures might be accepted and stored as a raster field of temperature. During a simulation, that information might be delivered in Centigrade as a raster map at perhaps a different spatial resolution. The Warehouse will also be involved with scheduling simulation module update events with the Main Control. The "Simulation" modules provide the engines that update the system state information over time in the Warehouse. To accomplish this, they continually gather information on which their algorithms rely, from the warehouse. The modules in the lower right, represented as ovals, pull information from the Warehouse for purposes such as visualization and storage.

Each simulation module may run with a different fixed time step, or may be event-driven (meaning that the module runs only when something happens—other than the passage of simulation time). Therefore, the timing and execution of simulation modules must be event driven to accommodate all differences. This means that a simulation event “calendar” must be maintained, managed, and followed. This is the responsibility of the Main Control module. The individual simulation modules themselves, especially those that operate on a fixed time step, will schedule many events. The Warehouse, based on specific changes in the data, will schedule some events. For example, a storm-event hydrologic module may give instructions to the Warehouse that it should be only called on rain events that generate more than 1cm of rainfall in an hour. The Main Control module steps through the events on the calendar and executes them by calling the associated modules and continually accepts new calendar events.

To better understand the roles of all the modules, it is useful to explore the communication requirements during the course of a simulation, beginning with initialization.

Initialization

Main Control Module—This module initializes the system state warehouse and then initializes each simulation, visualization, control, and data output module. Each is given a pointer to the system state warehouse and is provided with geographical extent and simulation time-frame information.

Simulation, Visualization, Control, and Data Output Modules—These communicate data input requirements and output possibilities to the system state warehouse. They also return event requests to the Main Control Module that will result in module execution requests called during the simulation.

Data Initialization Modules—These similarly communicate to the system state warehouse providing information about data that is available to initialize the data fields in the system state warehouse.

Main Control Module—Requests verification from the system state warehouse that all data needs are accommodated.

System State Warehouse—Performs an analysis on all data requirements with respect to availability of initialization data and run-time support for update of the data (if needed). If all requirements are met, space for all of the data needs is allocated and the initialization modules are called to provide the required data. An "initialization complete" message is returned to the main program. Note that all data fields are associated with time stamps indicating when the field was updated (in simulation time) and how long that information is valid. During the initialization step, simulation modules are called by the Warehouse to ascertain when the modules should be called to update the Warehouse data fields for which they are responsible. The returned information results in events to be scheduled by the Main Control Module.

Run or Step

Main Control Module—As maintainer of the event calendar, the main control module takes a user request to either run until further notice or run for a specified length of time (a "step"), and calls on the modules to execute the scheduled events. During execution, new event requests are accepted and placed on the schedule. (Users will have the option to watch a dynamically changing event schedule unfold and be executed.) During a "step" operation, the events on the calendar are executed until some specified simulation time is reached. A Run operation continues until a Stop is requested or there are no more scheduled events.

Simulation Modules—These run only when called by the Main Control Module and runs for only the length of simulation time identified by the Main. When asked to run, the Simulation module checks the time stamp of any cached system state information against the time stamps of that data in the System State Warehouse. The cached data will be updated as required. The simulation module then runs and then provides updates to system state data to the System State Warehouse. Before finishing, it has the opportunity to schedule its next event by sending a request to the Main Control Module.

Control Modules—These run in an identical fashion to Simulation Modules, the only difference being that system state information is changed by a user rather than through simulation.

Visualization Modules and Data Output Modules—Identical in operation to Control Modules, except that they do not change any system state information. Visualization Modules generate various transient displays of data during a simulation and the Data Output Modules capture system state information in files for later use.

The DIAS Approach

The Dynamic Information Architecture System (DIAS) is an object-oriented simulation-modeling environment developed to support the simulation of landscape/watershed processes. DIAS provides the programmer/developer of a simulation model with a framework within which the developer builds and/or adapts Entity Objects. These objects represent real-world objects that dynamically interact with one another during a simulation. These objects contain system state information (Properties) about themselves and methods (Aspects) for responding to interactions. Sets of Entity Objects are held by an Analysis Frame, which defines the simulation "world" and its purpose. As DIAS model developers have created a wide variety of models, a library of dozens of reusable Entity Objects has emerged, which are available for subsequent model developments.

The DIAS development environment allows for integrating external models. Like CDF, DIAS supports the notion of initializing simulation programs running on local or remote machines, running those programs, and receiving output. DIAS, however, further supports the ability to run the remote simulations synchronously with a DIAS-based model. Remote simulation models are initialized, are asked to run a number of simulation time steps, and are asked to change internal states and reveal internal system states. Within a DIAS model, an Entity Object responds to commands to advance in time and change/access state information by passing the requests to a matched remote simulation model. Legacy simulation models that support the notion of simulation steps can be easily implemented into the DIAS framework using a DIAS Model Wrapper. It provides the mechanism for linking models using a Model Controller and a Model Object. The Model Object is part of a compiled DIAS model and communicates via CORBA to the Model Controller, which is compiled as part of the legacy model.

Information is passed among Entity Objects through the Process Object, thereby fulfilling the role of the System Sate Warehouse (in Figure 9). Simulation time is managed by the Discrete Event Simulation Manager, which essentially manages a calendar of scheduled events. Events are associated with a particular moment in simulation time and can carry data.

A DIAS Example

Fort Future[*] is a major development effort that will allow military installation planners experiment, via simulation models, with the ability of their installation to support assigned military missions in the future. The five principal objectives of Fort Future are to:

- Evaluate the ability to project the force into a theater
- Evaluate the ability of the installation to protect stationed forces
- Analyze the ability of the infrastructure to support a change in the mission
- Evaluate the impact of facility acquisitions
- Evaluate the affect of urban encroachment on the ability of the installation to support the mission in the future.

These objectives are confounded by the fact that there are many cause-effect relationships among the physical, human, environmental, hydrologic, and economic conditions and processes on and off installation. It has been determined that the best way to address the Fort Future objectives is through simulation modeling, focusing primarily on the development of software objects that match real-world objects. Objects will include infrastructure such as roads, railroads, buildings, parking areas, training areas, and test ranges. They will include human objects such as battalions and companies, headquarters staff, and support staff and equipment such as tracked vehicles, weapon systems, trucks, and other vehicles. Objects representing services provided off-installation such as utilities, travel routes (such as roads and rail), and services such as water and recreation will also be necessary.

The Fort Future user environment will allow users to proceed through the following steps to assist in addressing specific questions regarding future options, impacts, and risks:

1. *Set Goals and Metrics*
   Identify alternative plans that involve changes in mission, investments, ownership and operation of land on and off the installation, and the objectives against which the plans will be measured.
2. *Gather Data*
   Information concerning the current state of the installation and its operation are gathered to support the analysis.

[*] A general description of the Fort Future Program is available through URL: http://www.cecer.army.mil/EARUpdate/NLFiles/2002/CASE-srmPV.cfm

3. *Build World*
   Pre-created agents representing the various key, interacting, parts of the installation are selected, placed geographically into the "world," and connected to represent the real-world interactions.
4. *Plan*
   Submit alternative plans of action to the model. These might involve changes in the use of buildings, new buildings and/or infrastructure, changed interactions among the agents, different layouts of the training/testing areas, or the creation/use of alternative routes for deploying troops.
5. *Simulate*
   The installation simulation model and planner specified inputs are brought together to generate outputs that show implications, risks, and costs/benefits associated with the alternative plans.
6. *Report*
   Simulation model outputs will be captured into movies, charts, tables, images, and text to rapidly communicate the results of the simulation runs.

Behind the user interface that allows planners to work through these steps, a DIAS engine is envisioned. A library of DIAS objects will be available that contains Fort Future agents that have been developed to interact with one another depending on the needs of the planner. There will be three fundamentally different types of ways that agents will be developed. First, DIAS supports the development of simulation objects that are wholly compiled into a DIAS model using the Java programming language. It is anticipated that all of the new Entity Objects will be developed this way—especially the smaller agents. Some of the behaviors required are already captured in legacy software that must be captured as software that runs as separate programs—either on the same computer or on other Internet accessible machines. Those that will have tight feedback loops with other DIAS agents will be encapsulated as DIAS support programs using the DIAS Model Wrapper software. Those that are not involved in feedback loops and can be run either at the beginning or end of a DIAS simulation can be made available as CDF Internet services. For example, GIS data necessary to initialize a DIAS model could be accessed via a CDF service to a centralized enterprise database. A DIAS simulation model that predicts future land use patterns might call a storm simulation model available as a CDF service to evaluate the ability of the pattern to maintain water quality standards.

Figure 10 sketches a possible Fort Future scenario that involves a user's machine and four other machines accessible via the Internet. In this scenario, a Fort Future DIAS runs on a remote machine managed by the Fort Future Development team. Users are provided with an interface that runs on their local machine, which controls the main program. Entity Objects in that program are internal and external DIAS objects and objects that make use of remote CDF services. External DIAS objects and CDF services are housed on any machine on the Internet—including that which contains the Fort Future model. In practice, the services will primarily be run on a cluster of machines geographically associated with the main Fort Future machine to ensure high bandwidth for communication among the coordinated processes.

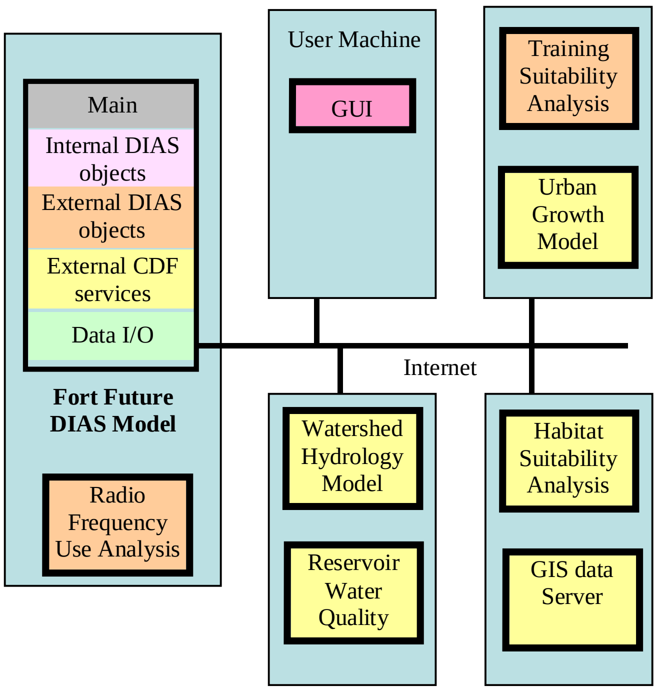


Figure 10. A Fort Future DIAS model.

11 Conclusion

ERDC has a long history of developing and delivering quality software that has been used to plan and manage military installations. Recent revolutions in hardware, software, and electronic networks have made it possible to now deliver software that can be even more useful. The communication and computer revolutions continue to provide ever-changing approaches to software development and deployment. Recently the infrastructure supporting the Internet has developed to reliably support the commercial sector of the economy. Because of the reliability, the opportunity to continue dividing software into reusable and distributed components continues. This dramatically changes the approach to software development, provides the ability to offer enterprise solutions that are composed of reusable components, decreases the installation and maintenance of software on user machines, and improves the management of enterprise software.

This work has explored the Land Management Suite concept from the standpoint of the Corps Enterprise Architecture, the five software design tiers of the Java Enterprise Edition's recommendations, and from the viewpoints of intended users, managers/supervisors, principal investigators, and software developers. The CEA pillars for developing future systems are Business, Information, Application, and Technical architecture views. The business area that this document addresses is the planning of Army installation training and testing areas. Available information includes current and historic land use, land cover, and land characteristics that include use by various species, by commercial/private users, and by the military. The principal application will be to evaluate alternative land management and land use plans with respect to cost, training/testing, agricultural, social, and environmental goals. Hardware and software architecture plans for supporting the business goals involve the application of Web-based enterprise solutions that provide users with a consistent look and feel interface that drives analysis processes that work on a common database.

Unlike the situation a decade or more ago, software delivered to military installations to support land management now joins a wide variety of hardware and software already at the installation. That is, there is already a suite of software at the installation, which includes all of the standard office software, various GIS software, Internet Web browsers, and other software. This document argues for the need to deliver an integrated suite of software from the Corps and discusses how to approach accomplishing this objective. However, the suite of software at the installations must now be considered as well. Information stored in installation GIS databases, spreadsheets, and other databases must be tapped.

It is now critical that ERDC developers of software capabilities developed to assist the planners and managers of military installation lands coordinate to ensure that a common database is used, a common user interface is presented, and appropriate use of the Internet is adopted. Appropriate use requires consideration of the development of remote services to avoid installing and maintaining software on user machines and consideration of the development and maintenance of enterprise databases shared by all installation offices, their supporting IMA office, and various contractors.

Success in the optimal expenditure of resources demands a new level of coordination and cooperation among program managers and developers.

References

Campbell, A.P. and J.R. Hummel, *The Dynamic Information Architecture System: An Advanced Simulation Framework for Military and Civilian Applications* (Argonne National Laboratory, Advanced Computer Applications Center, 2002).

Goran, W.D., J.P. Holland, et al., *Plans for the Land Management System (LMS) Initiative* (Engineer Research and Development Center, Construction Engineering Research Laboratory [ERDC-CERL]), 1999).

Holland, J.P., "Hydroinformatic Advancements in Integrated Hydroenvironmental Modeling Systems," *Hydroinformatics* (IAHR, Copenhagen, Denmark, 1998).

Majerus, K., *Evaluation Elements and Results: Use of DIAS and OO-IDLAMS Research for LMS* (2000).

Sydelko, P.J., K. Majerus, et al., "A Dynamic Object-Oriented Architecture Approach to Ecosystem Modeling and Simulation," *1999 American Society of Photogrammetry and Remote Sensing (ASPRS) Annual Conference* (American Society of Photogrammetry and Remote Sensing, 1999).

Westervelt, J.D., M. Shapiro, et al., *Geographic Resources Analysis Support System (GRASS) Version 4.0 User's Reference Manual*, Technical Report (TR) N-87/22, rev. (CERL, 1992).

White, T.E. *Transformation Through Base Realignment and Closure* (Secretary of the Army, Washington, DC, 2002).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 01-2005

2. REPORT TYPE Final

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
Preliminary Guidelines for Installation Product Line Land Management Suite (LMS) Product Developers

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
James Westervelt, Kelly M. Dilks, and William D. Goran

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Research and Development Center (ERDC)
Construction Engineering Research Laboratory (CERL)
PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/CERL TR-05-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Headquarters, U.S. Army Corps of Engineers (HQUSACE)
441 G St., NW.
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)
CEERD-CV-T

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

14. ABSTRACT

Several generations of land management support software has been developed for and delivered to military installation range control and environmental offices. A new generation of software development tools, hardware and networking environments, and operating systems provide an opportunity to design, develop, and deliver a new generation of tools that will be far more effective than previous versions. These tools will be easier to access, maintain, and integrate. This document provides an introduction to the new technologies and techniques and is written for all involved with installation land management software development including managers, supervisors, principal investigators, and programmers.

15. SUBJECT TERMS
fort future
land management
software
military installations

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	SAR	76	James Westervelt
					19b. TELEPHONE NUMBER (include area code) 217-373-4530

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18