Considerations for an Enhanced Decision Mapping System: An Information System Architecture Approach

Tim Nyerges and Christina Drew
Department of Geography and
Consortium for Risk Evaluation with Stakeholder Participation
University of Washington

Abstract

The current design of the Decision Mapping System (DMS) provides geographically-referenced access to information about Hanford 100-area soils cleanup decisions using WWW html pages and static map images. This report considers certain enhancements to the design. An information system architecture composed of six information system components at six levels of detail provides a framework for describing considerations for an enhanced design. Data management and WWW mapping tools are identified as part of that enhanced design to be able to support information for more areas and more decisions. Brief comments about next steps in the developed of an enhanced DMS are provided.

1. Background

The Decision Mapping System (DMS) provides geographically-referenced access to information about 100-area soils cleanup at Hanford (Drew 2001). Drew (2001) outlined user requirements for DMS through the use of interviews and literature search. In that requirements document, the strategy for information access is based on six information structures -

Decision Maps – maps locating decisions on the ground
Background Information - reference information as appropriate to a decision
Timelines (decision paths) – sequencing of decision events
Geographic Library – archive of documents referenced by geographic location
Value Trees – presentation of values, objectives and criteria for decisions
Commenting – general comments about the information presented

Those information structures are meant to provide different ways that people can access information about cleanup, interim protection, and long-term stewardship, and in particular100-area soils cleanup. As a research project, we are interested in examining a) the character of transparency as a concept, b) how each of those information structures separately and together enhance the transparency of cleanup decision-making, and c) ways that information can be organized to facilitate understanding about complex topics through the use of geospatial visualizations.

A number of design concepts for implementing the DMS user requirements come from geographic information system (GIS) developments, particularly examination of participatory GIS development and use in recent years (Harris et. al. 1995, Jankowski and Nyerges 2001). A GIS can be defined as a combination of hardware, software, data, people, procedures, and institutional arrangements for collecting, storing, manipulating, analyzing and displaying information about spatially distributed phenomena, including but not limited to people, places, and infrastructures … for the purpose of inventory, problem solving and/or decision making in operations, management, policy, and research contexts (Nyerges 1993). A participatory GIS is a GIS meant to be used to foster group-based exploration, analysis and conversation. The original definition for a GIS indicates that there are six integral components – hardware, software, data, people, procedures and institutional arrangements needed to create a functioning system, hence to "create information". Those same six components are integral to design considerations for participatory GIS, and thus for the DMS as well. In a review of the current design of DMS, it was concluded that an enhanced design was needed if the pilot prototype was to mature into a more robust design that could handle more cleanup decisions and more areas using a more flexible data management approach (Nyerges 2001).

In this report, we present initial considerations for an enhanced DMS design. To do this we make use of an "information system architecture" framework (Zachman 1987). The framework helps use describe the six significant components of an enhanced DMS are various levels of detail. Each of the six components of the definition for GIS presented above have a corresponding component in information system architecture. Although different labels have been used in the two contexts within the literature, we provide the correspondence as part of this work. In addition, the information system architecture framework contains six levels of detail, through which an information system design can be specified. Consequently, an architecture framework provides for a systematic presentation of the various nuances for describing the DMS in a comprehensive way.

To undertake the above, this report proceeds as follows. Section 2 presents general concepts about systems development. In section 3 we describe the current design architecture of the DMS, as a backdrop for comparison to the enhanced design. In section 4 we present an overview of the information systems architecture in the form of a table that consists of the 6 system components (columns) and 6 levels of detail (rows). In section 5 we further describe important considerations for the data and software components only. Section 6 provides comments about potential next steps in this effort.

2. Systems Development – an Evolving Concept

Traditionally, software (information) systems development has been described in terms of four phases as follows.

1) Requirements analysis – create a plan (specification) for what capabilities users need (although many developments speculate on user "wants")
2) Design – create a plan for how those capabilities are to be provided using software tools
3) Implementation - carry forward on the first two plans
4) Evaluation/testing – determine whether appropriate capabilities are actually provided?

Through improvements in approaches to software engineering and information systems development these four phases are now seen as "not so linear", because mistakes made early in the development process get carried forward as mistakes in design and implementation. Users are very good at spotting mistakes in functionality early on, developers are not so good at it, simply because of "whose information needs" are trying to be satisfied – the user's, not the developer's needs. Therefore, current development strategies encourage "participatory interaction" with users through all phases of the development process. Clarifying "proper direction" early on results in more usable, implemented systems. Getting the capabilities right is not the only challenge in system development. Flexible extension and "scaling" of capabilities (systems) depends on the technical design. That is, with a design being a plan for what technical tools to use to implement the capabilities, if the system is to be extended, some tools are better to use to enhance flexibility than are other tools. However, more flexible and detailed designs take more time, thus users do not get a chance to see some "semblance" of functionality for a while. It is for that reason that rapid prototyping has been encouraged in system development. Rapid prototyping is a way to provide users with a working prototype of information technology without going through full design and implementation, just so they can provide feedback to system developers to suggest whether the developer can get close to what is "really needed" in terms of information use. Development for DMS has proceeded using the prototyping approach, and we expect this approach to be used in the future because the DMS is a relatively novel concept at this time. Nonetheless, traditional ideas of systems design provide a "first-pass" guideline, and we expect to make use of the traditional concepts to help organize our prototyping effort.

3. Current Status of DMS Design

The initial prototype of the DMS development process created a specification based on user requirements generated through a participatory process. Since a DMS has never been created before, and users are not familiar with the concept, the challenge has been to provide a "vision for a DMS", based on empirical research with complex decision problems and concepts synthesized from the literature. This vision has come in the form of a collection of six "information structures" proposed to be useful for addressing decision situation issues and promoting transparency, and thereby, providing shared understanding of the complexity of the 100-area soil cleanup decision problem. The decision situation issues, the information structures proposed to address them, and the transparency components to be measured to determine success in helping to promote a shared understanding are provided in Table 1.

Table 1. Issues, Information Structures and Transparency in the Decision Mapping System

Issues	Information Structures	Transparency Concepts
Decisions ill-defined	Decision Maps, Background info	Clarity, Integration, Rationale
Complex decision steps	Timelines, Background Info	Logic, Integration,
Geographic complexity	Decision maps, Geographic Library	Clarity, Integration, Accessibility
Information Overload	Geographic Library Decision Maps, Timelines, Background Info	Accessibility, Integration, Clarity
Many stakeholders with Inexplicit values	Value trees, Commenting, Background Info	Rationale, Clarity, Accessibility

The six types of information structures constitute the types of pages that have been developed. Table 2 lists the six types of information structures and the tool used for implementing it.

Table 2. Current Design of the DMS

Information Structure	Design as Format of Structure
Decision Maps	ArcView map as .jpg image
Background Information	Text as .html document
Timelines	Line graphic as .html document
Geographic Library	List of resources as .html document
Value Trees	Graphic or indented list as .html
Commenting	Pop-up box as .cgi script to collect words

A user interface design, as the layout of information structures on the screen, and how they will interact, has recently been finalized, at least in a preliminary form. The user interface design ties together the six information structures into a functioning presentation for users. The initial prototype can be viewed at "http://students.washington.edu/cdrew/dms/index.html".

In the current design, the data for all information structures is instanced as WWW pages using hypertext markup language (html) and image data protocols. The html provides for static pages, i.e., no interactive requests for user input, expressed using html and image formats. The computer graphics interface (cgi) script to be used provides for an interactive capability, but it is a low-level language to provide capabilities.

Although an html formatted WWW page is editable by almost any application, an "image" format is a fixed display created "off-line", specifically for a particular context and then brought on-line as a "static picture". No new information can be added to an image once it is created. Thus, every time a different picture is needed to satisfy an information request, e.g., new layer of information on a map, that map must be created off-line and then posted to the DMS web site. In essence, the application is not "database driven". That is, the database(s) in the form of documents, maps, risk criteria, etc. are not "on-line", connected in as part of the system. As a result, "new displays" cannot be created in real-time, but must be developed "off-line" and then posted to the DMS web site. Consequently, it takes considerable effort to generate the images for each area being represented. That advantage in this approach is that we are working through the information needs. Thus, effort is being prioritized as in any "prototype development project" to make sure that the information requirements are appropriate – at least within the limited evaluation and implementation being performed.

The current software design of the Decision Mapping System architecture can be said to be an extended two-tier architecture, as depicted in Figure 1. That is, a browser and a web server constitute the two tiers. The discussion application is the extension of the second tier – providing a minimal level of feedback to a user.

The Web client as a standard or specialized browser handles information structure retrieval, using HTML and XML coding. The client package requests and decodes responses to position information on the screen. The Web page server handles conventional HTML retrieval from the web page archive, including .jpg and .html formats. The discussion server supports interactive commenting on web pages, supported by partial hypertext protocol (php). JavaScript has been used to assemble pages as part of the user interface.

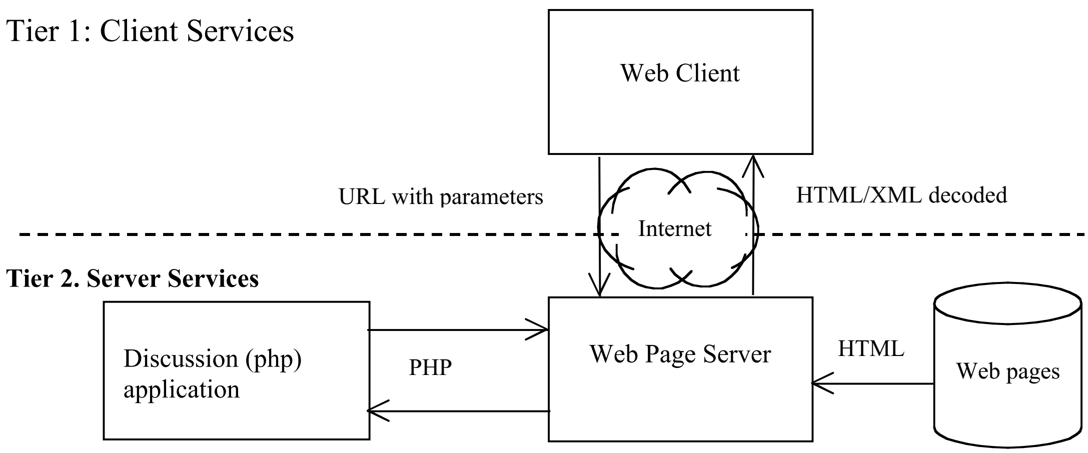

Figure 1. Current two-tier architecture of the Decision Mapping System.

The current design contains no data management and no computer map generation technology to provide for a system that scales to address more decisions, areas, and/or other sites. Even if the information requirements are deemed sound and applicable across other areas, which are likely to be the case, a more robust design is needed to support larger amounts of data and to be able to generate maps on-line. A more robust data management approach is needed to serve all of the information structures. A data server can "scale" the amount of data, as well as to make it easier to bring other areas on line. Flexibility in data management leads to effective representation when it comes to displaying information for different requests.

4. Architecture Considerations for Enhancing the Decision Mapping System

In this section, we outline an information system architecture (Zachman 1987). An information system architecture provides a framework for describing the six components that constitute the building blocks of an information system. In addition, the architecture can be used to describe those six components at different levels of detail, such that "users" can understand the language of the higher levels and system developers can understand the language of the lower levels. In this manner development can progress though a series of "language expressions" in an incremental manner, making sure that "users" can communicate what they need in terms of information capability to the designers and on to the implementers. It is possible to build an information system without using all of these levels, as fewer levels are needed for smaller systems (fewer people and less functionality are involved) and more levels come in handy for larger systems (more people and/or more functionality). Consequently, an architecture provides for a better understanding between users and developers of the "challenges" to be faced within information systems development. According to Zachman (1987) and subsequent IBM user group articulation, the six system components of an architecture (with corresponding terms from the GIS definition components in parentheses) are:

1) Data (Data)– What are the different data to be used in the system
2) Process (Software) – How will the information structures be provided to users
3) Configuration (Hardware) – From what locations will people be accessing the system, using what kind of hardware technology?
4) Role (People) – Who will be doing what in regards to developing and maintaining the usability of the data?
5) Timing (Procedures) – When will certain information be available to a certain level of user satisfaction?
6) Motivation (Institutional arrangements)– What fundamental concerns among people and organizations encourage access to certain kinds of information?

One can see the correspondence between the respective terms in the definition and the architecture. This should not be surprising since the definition that was provided evolved through various published literature to arrive at a reasonable consensus, and the architecture specification was created by the IBM Users Group in the late 1980's to early 1990's. The consistency provides a level of confidence that these components are indeed the major ones to consider if one desires a robust design. The six components are however, only one-half of the architecture framework. We have to know to whom those components are oriented in terms of "detail" of design. Each of the components in the architecture framework can be specified in terms of six levels of detail, whereby the top levels are oriented to users and the bottom levels are oriented to designers and developers. In actually, the very bottom level is the "working information system", when all languages are fully implemented. The six levels of specification for implementation are as follows.

1) Business scope – What phenomena are involved in a decision situation from a decision worker's perspective?
2) Business model – For that decision worker's business activity what flows of information are relevant?
3) Information systems model – How will the information flows be characterized within the information system environment?
4) Technology model – What specific data/software/hardware tools are needed to instance those information flows?
5) Technology definition – Specify in detail the formats/protocols etc. for those data/software/hardware tools in terms of a computable language
6) (Implemented) Information system – implement the computable language

Bringing the components and levels of detail together we have a plan for implementing a DMS as outlined in Table 3 – an information systems architecture. A major challenge over the past two decades of information systems implementation is to be able to have users, designers and developers work with a language that they all understand, but is robust enough to be translated easily into a computable form. Supposedly, the Unified Modeling Language (UML) is the most comprehensive requirements, design and computing-capable language, that has been developed and user world-wide (Rumbaugh, Jacobson and Booch 1999). UML is reasonably new such that only recently have geospatial data models appeared that are expressed in this language (ESRI 2001).

We might consider that the combinations of six system components and six levels of detail in Table 3 proceed through development in four steps as follows.

1) Information Needs in relation to each of the components
2) Tool Design Requirements in relation to each of the components
3) Implementation (tool) language to turn design requirements into a workable system

4) Evaluation (usability test) to review the implementation of the system in relation to needs

Table 3. An Information System Architecture For A Decision Mapping System.

	Data	**Process**	**Configuration**	**Role**	**Timing**	**Motivation**
Business scope	What decision situation aspects are important to cleanup and stewardship in 100-area soils and B/C reactor area?	What will be the decision activities that are to be addressed within what decision situations?	From where and how will information be accessed? Accessible anywhere in the world over WWW?	What are the responsibilities to make stewardship a success? Decision makers, stakeholders, technical specialists?	What is timing of decision activities to make stewardship a success?	What is the mission and goals of the stewardship activity?
Business model	What are the inter-relationships among the cleanup and stewardship information as highlighted by the ROD?	What decision activities are to be addressed in terms of information relationships?	What storage nodes and communications links are necessary for performance?	What organizational units have responsibility for what aspects of the system? What role do they play?	- What is the periodic nature of the stewardship activity?	What are the objectives for meeting each of the goals?
Information systems model	What set of information structures (and basic constructs) are to be made available to convey the information content? What granularity is appropriate?	What basic information manipulation capabilities (display, search, edit, construct) are needed for each information structure?	What is the hardware functionality necessary to get the information to people?	How are the roles to be implemented in terms of security?	What is the timing of the control structure for access to information?	How can the information objectives be addressed in terms of processing objectives?
Technology model	How are the information structure constructs to be represented in software? What granularity is achievable?	Information structures (are to be) developed in what technology?	What is the hosting hardware configuration?	Lay out the access groups in a security design that can be implemented in the chosen software.	When is information to be processed for a most efficient, effective, and equitable supply of information?	Specify the performance levels to be attained as in benchmark levels.
Technology definition	Prepare the database schema to implement entities and attributes	Implement application processes in appropriate tool	How will the hardware capacity be set up?	Specify the access definitions in the system.	Specify the maintenance schedule for updates of the information.	Established agreements among responsible parties to meet those objectives
Information System	Implemented data storage structures Actual data	Executable program code Actual software	Hardware System configuration Actual hardware.	Access privileges Actual User ID's, access control, etc.	Processing schedule Actual schedule to perform work	System management Actual monitoring facilities

Specifying, designing and implementing a system through refined steps allow many people to understand what is being done. If we look back to a traditional development approach, the first two (business scope and business model) levels are associated with information needs stage. The next two (information systems model and technology model) levels are associated with design stage. The last two (technology definition and information system) levels are associated with implementation stage. The evaluation stage takes a functional system and tests its usefulness to all users. Nonetheless, progress has been made with the DMS. User requirements have been articulated, thus one could say that the business scope and business model levels have been addressed. Since a prototype exists with working information structures, in a sense the information systems model at the third level is still useful. The fourth level is where a new design is called for, assuming of course that the first three levels provide appropriate information.

5. Technology Model Details of the Architecture

The technology model is where we enhance flexibility and/or performance of current capabilities. The details of the technology model include the data content and the manner in which that content is processed by software.

5.1 Data Content/Structure of the Technology Model

At the current time the 100-area soils cleanup is the focus of the DMS implementation. The main purpose of this enhanced design is to add additional decisions and/or areas. There has been some discussion about what to do with the 100 BC area – reactor, and it has been suggested as a museum area. Perhaps this is a direction of development for the DMS, but such a decision would be made by the Tri-Parties and interested and affected parties through further discussion.

In addition to decision and area content, the information structures maybe in need of refinement. An information structure organizes the content of information for a decision situation in such a way as to provide easy understanding of information for most people. Thus, an enhanced content might be multiple levels of granularity depending on the scope of the decision situation at hand as proposed in Table 4.

Table 4. The Constructs of Six Information Structures

Information Structure	Basic Constructs
Decision Maps	Map layers that portray any aspect of decision situation, e.g., boundary of area, operable units
Background Information	Words, sentences, paragraphs, bullets structured as per text document
Timelines	Line graphic, event, dates, descriptions
Geographic Library	Document resource(s) available online
Value Trees	Elements to form a graphic semantic hierarchy composed of goals, objectives, criteria, and attributes plus database categories that instance the attributes (criteria)
Commenting	Window for providing feedback in text form

4.2 Process (Software Tools) of the Technology Model

The information systems model focuses on the generic manipulation capabilities that are possible. That is, the kind of interaction that is required for each information structure. We can list these in terms of the basic DMS and the enhanced DMS.

DMS

- Display the information structure content

Enhanced DMS

- Search/query/display the information structure content
- Comment on information structure content
- Vote on information structure content to set priority
- Construct/edit/display information structure content

Thus, we can describe the manipulation capabilities for each information structure as in Table 5, which then directs the appropriate software technology to be used as specified in Table 6.

Table 5. Requirements for Information Structuring at Five Levels of Capability

Information Structure	**Possible capability**
Decision Maps	- Retrieve/display decision information in the form of spatial and attribute data as map data by selecting a category - Search/query decision information (map data) for an area(s) and/or a category as represented by attribute - Comment on a specific feature or group of features - Vote on the priority of considering a feature, e.g. operational unit - Construct an option for consideration
Background Information	- Retrieve background information as html pages - Generate information in Adobe Photoshop
Timelines	- Display image map of timeline (CERCLA normative path) - Generate the timeline online as an agenda for review - Update the timeline online
Geographic Library	- Retrieve and display Html - Link documents through database
Value Trees	- Display Graphic image map or indented list - Retrieve from database and portray connections among abstractions - Search connections among abstractions
General Commenting	- Collect comments - Organize comments in categories

Table 6. Software Technology to be Used for Implementing the Enhanced DMS

Information Structure	**Tool for Implementing Enhanced Structure**
Decision Maps	ArcIMS - Arc Internet Map Server driven by ArcSDE – Arc Spatial Data Engine supported by Oracle or SQL Server
Background Information	Text as .html document
Timelines	ArcIMS – Arc Internet Map Server
Geographic Library	SQL Server for documents
Value Trees	ArcIMS supported by ArcSDE and SQL Server for graphical depiction
Commenting	Phorum or other open source discussion

To support the information structure capability described above using the components mentioned, a three-tier software architecture is necessary, as depicted in Figure 2. The enhanced design introduces the ArcIMS (Arc Internet Map Server) in Tier 2 and the data management system in Tier 3 (ESRI 2000).

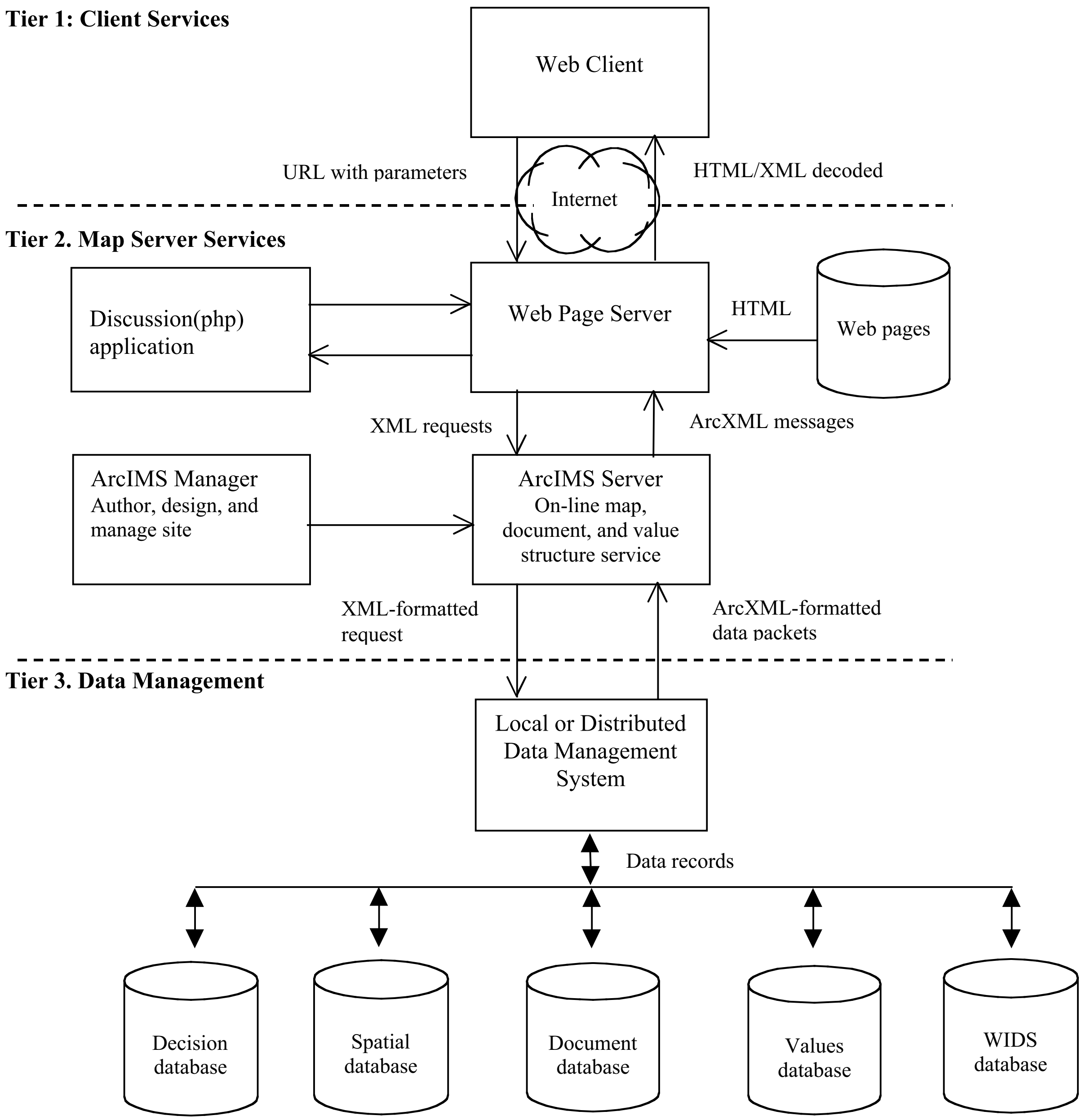

Figure 2. Three-Tier Architecture for an Enhanced, Decision Mapping System

When a three-tier architecture is used, the browser client can provide for dynamic creation of information structures, i.e., map retrieval (construction), value structures, and timelines, as for example the mapping capabilities in Table 7.

Table 7. Browser Capabilties Supported by an Internet Map Server

- pan and zoon on a map
- identify and find features
- search for features
- query data
- display map tips
- select and buffer features
- measure distances and work in scale bar units
- add map notes
- use edit notes
- open layer properties
- change layer properties
- add data

The **ArcIMS manager** is used to Author, Design, and Manage an ArcIMS web site. Each of the three stand-alone modules is wizard-driven, but they all work together to create functionality. The output from the ArcIMS Author is a map configuration file, written in ArcXML (Arc eXtended Markup Language). The ArcIMS author allows users to define map application content in that configuration file. The ArcIMS Author is menu-driven, allowing a user to step through the map content definition process. ArcIMS Author provides users with the capability to connect to databases, symbolization, and other mapping parameters. Generating an on-line map involves adding data content and setting map properties that establish a "map service". A map service allows the content of a map configuration file to be published on the Internet and sets the framework for the website functionality. The map configuration file can be edited in a text editor independent of the ArcIMS Author module.

The ArcIMS Designer is used to define the Web service that the end user will view. The module leads a designer through a series of steps including selection of MapServices, templates, and the operations and functions that will be available to the client Web browser.

The ArcIMS Administrator is used to control the operation of the Web mapping site. It assists with managing MapServices, servers and folders, even while servers and services are added and removed.

The **ArcIMS server** decodes XML to make requests on the data management system. ArcIMS server can handle most graphical services, but would not be sufficient for analytical processing. That would have to be performed by an application server, if such a capability is necessary. Results are packaged as ArcXML messages and delivered to the client as decoded into HTML pages.

The **Data management system** stores spatial data, document data, and values data in tables, when large amounts of such data exist. At the current time the application services to construct timelines and value structuers are not presented in the enhanced architecture, as they might not be needed in an enhanced Decision Mapping System capabiltiy. This remains to be determined through experimentation and user feedback with the current design.

5.3 Host Server Configuration Considerations

Options for host server environment are: 1) the U of Washington Dept of Geography Windows 2000 server, 2) the U of Washington Computing & Communications Unix server, 3) Rutgers University CRESP Server, 4) the U of Washington CRESP Unix server, 5) DoE Server or 6) an independent Internet Service Provider. At the current time the host server environment for the DMS will be the Department of Geography's Windows 2000 web server. This choice has been made because of development flexibility under full control of the development team.

5.4 Role

A major issue in the development process concerns the individuals or organizations that have been involved in the development of the basic DMS. Several representatives of stakeholder organizations have been part of the participatory design of the basic DMS, as organized by Christina Drew as the main thrust of her Ph.D. dissertation. Prof. Tim Nyerges has been the project coordinator on the basic DMS, in his role as Christie's advisor. Thus, CRESP has acted as developer, and stakeholders have been envisioned to be the users, but it has been a participatory development process.

Representatives from a broad array of Hanford Stakeholder groups have provided their insight and feedback through interviews, including U.S. Department of Energy – Richland Operations Office, U.S. Environmental Protection Agency – Region 10 Hanford Field Office, Washington Department of Ecology – Nuclear Waste Program, Oregon Office of Energy, Columbia River United, Government Accountability Project, University of Washington Evans School of Public Affairs, CRESP Researchers. Other groups provided feedback after presentations at conferences or to advisory group meetings - e.g., Hanford Reach Conference, Eco-Informa 2001, Hanford Advisory Board Public Involvement and Communications Committee.

A fundamental question for the enhanced version is who is involved in the development of the enhanced version – the same groups, or broader roles for groups? The technology model incorporates the various organizations access to both development and use. This question is likely to become clearer as the testing of the basic DMS proceeds as organized by Christie Drew.

5.5 Timing

A fifth component of the architecture involves timing. It what timely manner will the data be updated? Who will perform the updating under the different system implementations? There is no question about timing in the basic DMS as it is under the supervision of Christie Drew. She has full responsibility to make the updates. However, in an enhanced DMS, when other parties are involved, the custodianship of the updates may well be a different group than the original developers – perhaps a stewardship group convened for such purposes. In addition to the current decision situations, another question to pose is "When do other decision situations come online?"

5.6 Motivation

Last, and perhaps most important in the development of any information system is "From where does the motivation come to get things done?" Long-term environmental stewardship is a critical concern with environmental cleanup – when the cleanup problems are so significant that they will affect thousands of people over a long period of time. However, continued motivation for such an approach to transparency would come only if society "values" the importance of cleanup, such values for concern being safety, human health, sometimes cheaper now than in the future, ecosystem health, and perhaps others. Clearly, each of the cleanup decisions is important, but a priority ranking of when to bring what decisions on-line is critical.

6. Potential Next Steps

This report provides only a preliminary architecture as a basis for a conversation about what is possible in a DMS. The technology model level of detail was used as a starting point because we assume that the user needs and requirements as those provided by Drew (2001) are sufficient. This assumption is clearly are rather important and significant one that would be addressed if an enhanced DMS specification, design and implementation effort were to take place. All of the components of the architecture are dependent on "who is to be served" by an enhanced DMS. This question is open to further discussion at this time as we seek collaborators to move ahead.

7. References

Armstrong, M. P., 1993. Perspectives on the Development of Group Decision Support Systems for Locational Problem Solving, *International Journal of Geographical Systems*, 1(1):69-81

Bostrom, R. P., Watson, R. T., Kinney, S. T., 1992. *Computer Augmented Teamwork*, New York: Van Nostrand Reinhold.

Couclelis, H. and Monmonier, M. 1995. Using SUSS to Resolve NIMBY: How Spatial Understanding Support Systems Can Help with the "Not in My Back Yard" Syndrome, *Geographical Systems*, 2:83-101.

Drew, C. 2000. Decision Mapping to Promote Transparency of Long-Term Environmental Stewardship, unpublished Ph.D. proposal, Department of Geography, University of Washington, Seattle, WA.

Drew, C. 2001. Decision Mapping System and Transparency Project: Preliminary Design Report, CRESP Technical Report, Public Policy Project 6, Consortium for Risk Evaluation with Stakeholder Participation, University of Washington, Seattle, WA.

Drew, C. and Nyerges, T., 2001. Promoting Transparent Decisions for Long-term Environmental Stewardship: A case study of soil cleanup in the Hanford 100 Area, *Journal of Risk Research,* submitted.

Drew, C., Nyerges, T., McCarthy, K., and Moore, J. 2001. Using Decision Paths to Explore Three Environmental Cleanup Decisions: A Cross-Case Analysis, *International Journal of Environment and Pollution*, in press.

ESRI (Environmental Systems Research Institute) 2000. Using ArcIMS 3.0. Environmental Systems Research Institute. Redlands, CA.

ESRI, Environmental Systems Research Institute 2001. ArcGIS Data Models. http://www.esri.com/software/arcgisdatamodels/index.html, accessed September 24, 2001.

Harris, T.,Weiner, D., and Warner, T., and Levin, R. 1995. Pursuing Social Goals Through Participatory GIS: Redressing South Africa's Historical Political Ecology. In J. Pickles (ed.) *Ground Truth: The Social Implications of Geographic Information Systems*, New York, Guilford, 196-222.

Jankowski, P. and Nyerges, T. 2001. *Geographic Information Systems for Group Decision Making,* Taylor & Francis: London.

Jankowski, P. and Stasik, M. 1997. Spatial Understanding and Decision Support System: A Prototype for Public GIS, *Transactions in GIS,* 2(1):73-84.

Jankowski, P. and Nyerges, T. 2001. *Geographic Information Systems for Group Decision Making,* Taylor & Francis: London.

Moore, T. J., 1997. GIS, Society, and Decisions: A New Direction with SUDSS in Command, *Proceedings*, *AutoCarto 13*, v. 5, American Society for Photogrammetry and Remote Sensing, Seattle, Washington, pp. 254-266.

Moore, T., Nyerges, T. Jankowski, P. and Smith , A. 1998. An Issue Development, Exploration, and Analysis System (IDEAS) to Support Group-Based Environmental Decision-Making for Brownfields Redevelopment, Phase I Final Report, National Science Foundation, Small Business Innovation Research Program, 1998.

Nyerges, T. L., 1995. Cognitive Task Performance Using A Spatial Decision Support System for Groups, in *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*, T. Nyerges, D. M. Mark, R. Laurini, and M. Egenhofer (eds.), Proceedings of the NATO ARW, Mallorca, Spain, March 21-25, 1994, Dordrecht, Netherlands: Kluwer Academic Publishers, 311-323.

Nyerges, T., Barndt, M., Brooks, K., 1997. Public Participation Geographic Information Systems, *Proceedings, Auto-Carto 13*, Seattle, WA, American Congress on Surveying and Mapping, Bethesda, MD, pp. 224-233.

Nyerges, T. and Jankowski, P. 1997. Enhanced Adaptive Structuration Theory: A Theory of GIS-supported Collaborative Decision Making, Geographical Systems, 4(3):225-259.

Nyerges, T., Jankowski, P. and Drew, C., 2002. "Data Gathering Strategies for Social-Behavioral Research About Participatory Geographic Information Systems Use", *International Journal of Geographical Information Science*, in press.

Obermeyer, N. 1998. The Evolution of Public Participation GIS, *Cartography and Geographic Information Systems*, 25(2):65-66.

Rumbaugh, J., I. Jacobson, G. Booch 1999. *The Unified Modeling Language*, Reference Manual, Addison-Wesley, Reading, Massachusetts.

Shiffer, M., 1995. Geographic Interaction in the City Planning Context: Beyond the Multimedia Prototype, in *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*, T. Nyerges, D. M. Mark, R. Laurini, and M. Egenhofer (eds.), Proceedings of the NATO ARW, Mallorca, Spain, March 21-25, 1994, Dordrecht, Netherlands: Kluwer Academic Publishers, 295-310.

Zachman, J. 1987. A Framework for Information Systems Architecture, *IBM Systems Journal*, 26(3):276-292.

Acknowledgements

This paper was prepared with funding support of the U.S. Department of Energy, under Award No. DE-FG26-00NT40938. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the DOE.