Interoperability for GIS Document Management in Environmental Planning

Gilberto Zonta Pastorello Jr., Claudia Bauzer Medeiros,
Silvania Maria de Resende, and Henrique Aparecido da Rocha

Laboratory of Information Systems – Institute of Computing,
University of Campinas – CP6176, 13081-970 – Campinas, SP, Brazil
{gilberto, silvania, henrique}@lis.ic.unicamp.br
cmbm@ic.unicamp.br

Abstract. Environmental planning requires constant tracing and revision of activities. Planners must be provided with appropriate documentation tools to aid communication among them and support plan enactment, revision and evolution. Moreover, planners often work in distinct institutions, thus these supporting tools must interoperate in distributed environments and in a semantically coherent fashion. Since semantics are strongly related to use, documentation also enhances the ways in which users can cooperate. The emergence of the Semantic Web created the need for documenting Web data and processes, using specific standards. This paper addresses this problem, for two issues: (1) ways of documenting planning processes, in three different aspects: *what* was done, *how* it was done and *why* it was done that way; and (2) a framework that supports the management of those documents using Semantic Web standards.

1 Introduction

Environmental planning covers many aspects and geographical scales, ranging from a city section to the global level. It is a continuous process that requires to constantly monitor the region under study. Multidisciplinarity and dependence on cooperative work are characteristics of environmental planning activities.

During the development of environmental plans many steps are carried out. Among them can be singled out: (1) identification of problems to be considered in a given geographic area – the “diagnosis”; (2) development of strategies to solve or minimize these problems at short, medium and long term – the “plan”; (3) Implementation of the chosen strategies – plan “execution”; (4) plan revision and maintenance – “follow-up”. This process is strongly based on using geographical data and Geographical Information Systems (GIS).

Steps (1) and (2) are backed up by two kinds of document sets:

- A set of maps and related descriptive data which detail the characteristics of the studied region. Maps usually portray two types of situation: the current situation, which is the input to the planning activity; and the possible outcomes of plan execution (the desired final state);

S. Spaccapietra and E. Zimányi (Eds.): Journal on Data Semantics III, LNCS 3534, pp. 100–124, 2005.

– A set of directives which specify how to achieve the planning goals, enacting them using the maps as background.

Plan execution (3) is the implementation of the directives. At each stage there are several alternatives that should be discussed and revised by teams, considering, for instance, options on preservation or recovery of environmental resources to be balanced against economic exploitation constraints.

This process requires detailed documentation, but there is a lack of tools to support document management. As a consequence, if a similar problem occurs in another region, it is necessary to start from scratch. This hampers plan modification and detection of methodological errors. Documentation is also important for communication among designers, in order to aid plan maintenance and evolution. As the planning process grows in complexity, more people and technologies must be involved, augmenting the need for documentation. Moreover, documentation provides information on the use of given datasets, and the context in which they are used. Thus, it provides additional semantics to a given planning procedure.

Yet another factor to consider is the fact that Spatial Decision Support is moving from a closed, tightly controlled computational environment to an open, Web-based context. This brings up new research and development challenges. Web GIS can no longer be seen only under the perspective of GIS accessed via the Web. They must also consider that their users and data are distributed all over the world. Thus, the Web has created not only the need for GIS distribution and interoperability but also requires offering domain experts easy means of publishing and accessing distributed resources and documents.

This paper presents a computational framework to support cooperative environmental planning activities on the Semantic Web. This framework is centered on the notion that documentation is a key issue in fostering collaboration and reuse and attaching more semantics to data and procedures. In this context, documentation should describe not only the data used – e.g. a region's geophysical and economic context – but also the planning process itself. Based on these observations, the proposed framework supports management of three main kinds of documents on the Web: *what* was done, *how* the plan was produced, and *why* the plan was developed along given lines.

Part of the framework has already been implemented at the University of Campinas, where these documents have proven to be useful in a local context. This implementation led to the Decision Support System named WOODSS (Workflow-based spatial Decision Support System), see [21, 35]. It has been used to test and validate ideas related to environmental planning support and associated documents [33].

However, in order to support cooperation across the Web, semantics and interoperability issues must be considered. Answering this need, this paper extends the documentation paradigm to the Semantic Web in two ways. First, it adopts XML to represent these documents, thereby providing the basis for interoperability. Furthermore, it discusses the use of existing domain ontologies as the means to attach further semantics to documents, data and planning processes, levering cooperation and automatic execution of processes on the Web. We furthermore adopt Web Services for framework implementation. The result is a step towards fully interoperable Spatial Decision Support Systems.

The remainder of this paper is organized as follows. Section 2 presents some basic concepts and related work. Sections 3 and 4 specify the three kinds of documents, detailing internal database and Semantic Web representations. Section 5 presents the WOODSS system and implementation issues. Section 6 shows an application example. Finally, Section 7 presents conclusions and ongoing work.

2 Related Work and Basic Concepts

The main concerns in our work involve documentation of planning procedures and the Semantic Web. Related work is thus centered on these issues.

Documentation adopted by environmental planning experts is highly unstructured. It is usually maintained in very large textual files. Automated support for such documentation is limited to text processing tools. Also, domain experts largely ignore Computer Science advances in this area. Consequently, there are few studies on document management for environmental activities.

Our research takes as starting point one of the few works that deals with documentation within a geographic context, viz. [30]. This work proposes the management of *What*, *How* and *Why* documents associated with the changes occurring in a spatiotemporal database, to support a better understanding of the evolution of geographic phenomena in the context of urban development applications. Documentation and spatial objects are managed jointly in a single database, in order to document change reasons, procedures and originators.

Our documentation goals, as will be seen, require a finer grain of detail, due to the particularities of environmental activities. Specifically, our *What* documents consist of metadata as well as additional data stored in hypertext/hypermedia graphs. Furthermore, like [30], workflows are used to store *How* documents, and design rationale for *Why*. There follows a short survey on related work in issues for each of these documentation choices.

2.1 Hypermedia and Metadata

Hypermedia represents an approach for management of information where data are stored in a network of nodes connected by links. A node represents a concept or idea and contains some multimedia data, such as text, graphics, video or images. Links represent relationships between nodes. The content of a node is presented by activation of links.

Hypermedia technology is used in applications that manage dynamic documents as in digital libraries [28] or at the Web. Also it can be used in other contexts, e.g., version control [19] and integration of heterogeneous software development environments [2]. For a formal representation and comparison of different hypermedia data models see [46].

The Dexter model [17] is a widely adopted hypermedia reference model, where a hyperdocument consists of a set of *components*. A component includes a *con-*

tents specification, a general-purpose set of *attributes*, a *presentation specification* and a set of *anchors*. A component can be an *atom*, a *link* or a *composite*. The atomic component represents the hypermedia 'node' abstraction, containing generic data. Links are entities that represent relationships between components. The contents of a link component is a list of *specifiers*, each including a presentation specification as well as component and anchor identifiers.

DHM [26] is an object oriented open hypermedia system based on Dexter. Its data model extends Dexter's links, anchors and components/compositions. The model supports dangling links – links having zero or one endpoint – and anchoring is extended to include a distinction between marked and unmarked anchors.

Other models in the literature extend Dexter to include, for instance, adaptive techniques or semantic connectors. For our purpose, however, it suffices to use Dexter's basic model and some extensions proposed in DHM. Hypermedia serves as a basis for storing *What* documents, enhanced with metadata.

Metadata, in the sense of data that describe data, are useful in many contexts – documentation, semantics and support for data retrieval. In the GIS context, metadata are classified in three levels [10]: description of the studied domain; characteristics of exchanged data; and characteristics of the geographic information. Several metadata standards have been proposed for storing and exchanging geographic data. WOODSS' metadata [34] complement What-documentation and are based on the FGDC's Content Standard for Digital Geospatial Metadata (CSDGM) proposal [13]. They contain information on spatial and temporal characteristics, as well as lineage and quality information.

2.2 Design Rationale

During a design process, many alternatives can be adopted. Designers need to analyze each option and choose the more suitable one according to goals to be reached. Design rationale (DR) is an artificial intelligence technique that supports a formal representation of the reasons behind decisions taken in a design process. It allows keeping track of assumptions made during this process, and the discussions conducted within a design team – and sometimes across teams – to arrive at a given solution. DR is object of research mainly in Artificial Intelligence [6], Software Engineering [18] and Human-Computer Interfaces [24]. Our work uses an extension of these techniques for creating *Why-documents* in environmental planning activities.

DR usually adopts semi-formal methods based on directed graphs, where edges and nodes acquire specific semantic meaning. All these models have a set of basic elements that formalize the discussions around a given project – the questions posed, the alternatives that are raised in response to questions and the arguments for and against the alternatives. These elements, which can be interlinked, are represented in IBIS (Issue-Based Information Systems) [11], a pioneer effort to formalize DR, through the entities *Issues*, *Positions* and *Arguments*. The links can be of eight kinds and they have intuitive meaning. For example, a Position <Responds-to> an Issue; Arguments must be linked to their Positions with either <Supports> or <Objects-to> links; and so on.

Other models include PHI [16] that uses similar concepts to IBIS, and Design Space Analyses (DSA) [23]. Proteus [25] is a model for documenting and managing the rationale of software design. DR techniques have been used in other contexts, e. g. support for design reuse and collaborative design in design engineering projets [50].

2.3 Workflows and Scientific Workflows

A *workflow* denotes the controlled execution of multiple tasks in an environment of distributed processing elements. It can be defined as a set of tasks involved in a procedure along with their interdependencies, inputs and outputs. Each task is called an *activity*, which is a unit of work and can be executed by one or more agents, in a given *role*. An *agent* is a person or software component able to execute one or more activities.

Traditionally, workflows have been used for total or partial automation of business processes. *Scientific workflows* [37, 43] allow documenting and specifying scientific experiments and procedures. Scientific work documentation requires special treatment because it is characterized by a great degree of flexibility and presents a much higher amount of uncertainty and exceptions than business work. Scientific workflows extend business workflows functionality supporting the following aspects: *incompleteness*; *partial re-use*; *abandon/rewind and dynamic modification*; *tracing of invalid processes*; *specification from case*. For a description of these aspects see [1].

In business applications, the main motivation for introducing workflow management is the desire to "re-engineer" work to enhance efficiency. The motivation for workflow management in scientific applications, additionally, is to help to control experiments, and to make available to scientific users the information on how experiments were conducted [35]. A recent trend concerns the use of workflows across the Web, to support cooperative work organization (e.g., the special issue on internet-based workflows in [31], or the work of [8] on coordinating communication among workflows).

Environmental planning activities have the same peculiarities of scientific work procedures. Thus, we adopt the scientific workflow paradigm to document *How* these activities are performed. Examples of this kind of use are [5, 7, 8, 20, 31, 32], involving geospatial data for e-government, in situation of emergency planning and environmental disaster management.

2.4 Semantic Web Related Efforts and Standards

Our choices for document representation favor flexibility in document construction and ease in document exchange, by following specific standards. Such characteristics are important when it comes to cooperative processes, and become essential when we consider our ultimate goal, that is interoperability and reuse in the Web. The Semantic Web is being proposed as an evolution of today's Web to make the information available on the Web easily usable, with the aid of automatic tools. The World Wide Web Consortium is the association that leads the standards efforts on the Web and Semantic Web [42].

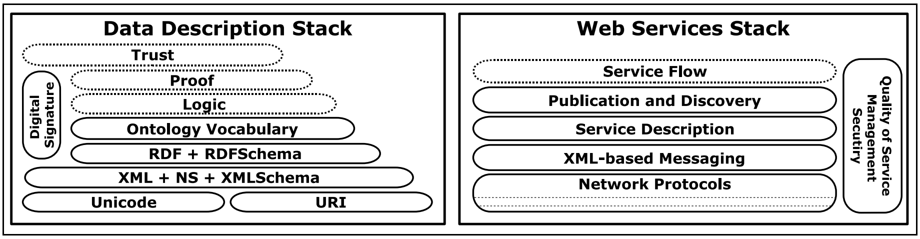


Fig. 1. The Semantic Web and Web Services Standards: Data and Services Description

The conceptual separation between data and services induces an implementation for the Semantic Web. On one side there are the data that should be semantically understood in the same way wherever they are used. On the other, pieces of software should provide a satisfactory degree of automation when handling these data. Such pieces of software often are Web Services. Figure 1 shows the proposal for data and services standards structure, portrayed in layers, where each layer supports the construction of the ones on top of it. Layers within dotted boxes do not yet have consensual standards.

At the data description part, Unicode encoding is used for processing textual data in any system, and URI, or Uniform Resource Identifier, to univocally identify an abstract or physical resource. Next comes the syntactical base for representing data in a semi-structured fashion, using XML and its associated standard for namespaces and definition of types, XMLSchema. RDF (Resource Description Framework) addresses semantics requirements. It forms a foundation for processing metadata and to express relationships. The Ontology Vocabulary layer uses an ontology language to formally describe the meaning and the terminology used in Web documents. OWL (Web Ontology Language) is likely to become the standard for this layer. The Digital Signature layer gives data a certificate that guarantees their origin. The Logic layer establishes a logical system through which the Proof layer can perform inferences about the data represented in lower layers. Digital Signature combined with Proof assures the validity of the information to be derived in the Trust layer.

The services stack defines distinct service layers. The XML-based Messaging layer provides a message formatting protocol, based upon usual network protocols, offering a high level abstraction for composing and exchanging messages formatted in an XML compliant language. SOAP (Simple Object Access Protocol) is the standard recommended by W3C for this layer. The Service Description layer provides a way to describe Web Services capabilities and communication interfaces. WSDL (Web Service Description Language) is the standard for this layer. Service Publication and Discovery using UDDI (Universal Description, Discovery and Integration) as a standard provide means to make Web Services reachable. The OWL-S language [39] is being proposed as a complement to service description, publication, discovery and composition standards and can even replace them at some degree. Quality of service, security and management are issues that must be considered at every layer of the Services stack.

The Service Flow layer is responsible for coordinating the composition of Web Services in order to achieve a specific functionality. Several standards have been proposed for this layer. They are of special interest to our work and are discussed in Section 2.5.

2.5 Workflow Interchange Standards

Workflows play a major role in constructing applications across the Web, helping to compose and coordinate Services. Currently, there are two main approaches being used to represent workflows on the Web. The first is to directly use an XML-based specification. The other favors functionality, by proposing means of composing services. Since we use workflows for *How-documentation*, and these workflows must support execution, we need to consider how to represent them for a distributed execution on the Web.

There are two major proposals of XML-based languages to represent workflows: XPDL (XML Process Definition Language) [45], and BPEL4WS (Business Process Execution Language for Web Services) [4]. The first was created explicitly to represent workflows in an accessible language. The latter was introduced to meet the requirements of service composition on the Web, using workflow concepts. These two viewpoints generated different, though overlapping, solutions.

XPDL aims at providing a "lingua franca" to represent workflows, enabling different Workflow Management Systems to use the same process specifications. BPEL4WS was introduced as a language to represent service flow coordination and is based on the merge of two other coordination standards, namely IBM's WSFL [22] and Microsoft's XLANG [38]. Recently, BPEL4WS was turned over to a committee of the Oasis-Open consortium [27], which changed its name to WSBPEL (Web Services Business Process Execution Language), and will be responsible for evolving the standard from now on.

Other languages include BPML (Business Process Modeling Language) [3] and the WSCI (Web Service Choreography Interface) [49]. These two languages have different scope than WSBPEL. Whereas BPML has a broader application context, WSCI is restricted to defining roles of services in a composition and needs not understand the whole process definition. Comparison and evaluation of these workflow representation proposals appears in [40, 48]. As discussed in Section 4, we adopt WSBPEL for publishing our *How-documents* on the Web.

3 Specification of Documents for Environmental Planning

Section 2 established the theoretical foundation for our proposal, discussing document management and Semantic Web issues. This Section presents the structures we propose for environmental planning documentation, namely, a hypermedia model to represent *What* documents, scientific workflows to represent *How-documents*, and design rationale structures to store *Why-documents*. The notation used to present the models is based on the entity relationship diagram for simplicity sake. All documents are stored in database tables to be published on the Web. Their integration is supported via additional entities, as well as by links within *What-documents*. For more details the reader is referred to [29, 33]. This Web representation uses XMLSchema (see Section 4).

3.1 Hypermedia Data Model: *What*

A *What* document describes the environmental plan itself – i.e., it supplies a general vision about what was done in the planning activity describing, for example, the plan objectives and methodology used for solution. The choice of a hypermedia model to document *What* data was based on two main factors: (1) it allows organizing documents in a non linear manner, thus facilitating user interaction and semantic links; (2) it supports incorporating multimedia data and thus remote sensing data, essential in environmental planning.

The hypermedia data model designed to document *What* is based on the Dexter Model [17] and some extensions proposed in DHM [26]. The Dexter Model was chosen because it is a reference standard used in many hypermedia systems and it has a well defined set of elements. Figure 2 shows the ER specification of the proposed model. Its main entities are Hyperdocument, Node, Anchor, Endpoint and Link. These entities have the standard semantics of hypermedia documents. Section 6 shows an example of their use within environmental planning.

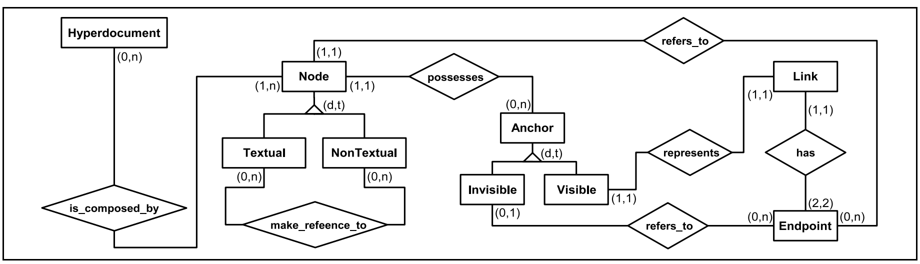


Fig. 2. Hypermedia Data Model for *What* Documentation

A *What* hyperdocument details a problem, and its links point to other (*What*, *Why* or *How*) documents and metadata. Thus, description of a given environmental problem (e.g., whether to allow cutting trees in a preserved area) can be linked to other relevant documented plans (e.g., describing how such an enterprise was successfully conducted in similar conditions).

3.2 Design Rationale Model: *Why*

The *Why* of the decisions in an environmental plan use design rationale. An important aspect in environmental planning is considering the risks presented by some solution alternatives. Risks can be decisive in the choice of an alternative to be implemented. During monitoring/maintenance of an already implemented plan, documentation of risks can be added to explain why a given solution does not work. Thus, in addition to usual design rationale elements, our model supports risk registering for each solution alternative. For instance, in the tree-cutting example, mentioned in Section 3.1, an obvious risk would be the impact on fauna and biodiversity. Figure 3 illustrates the proposed model.

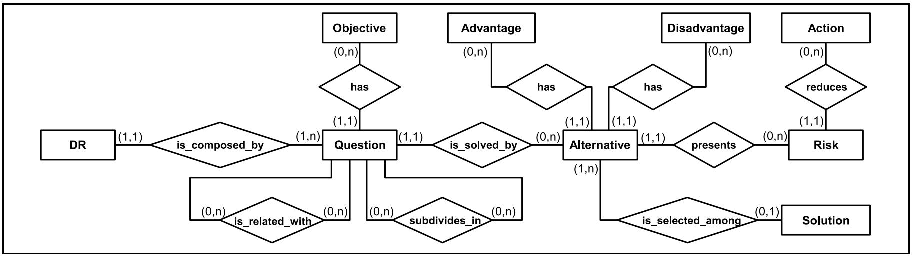


Fig. 3. Design Rationale Model for *Why* Documentation

A *Why-document* is formed by aggregation of questions raised during discussions of a given problem, but that are not necessarily interlinked. A *Question* formalizes a point raised during a design process, for which there are possible *Alternatives*. A *Solution* is the alternative selected for implementation, while an *Objective* is a requirement that should be satisfied by the solution. *Advantage* and *Disadvantage* record positive and negative points concerning an alternative. Any alternative can have *Risks*, and *Actions* can be carried out to reduce a risk.

The meaning of each relationship can be easily comprehended through its name. The relationship <subdivides_in> takes into consideration that complex *Questions* can be solved indirectly by decomposition, i.e., complex questions can be decomposed in more simple questions.

3.3 Scientific Workflow: *How*

The structure designed to represent a scientific workflow to document *How* is an adaptation of the workflow systems standard defined by the Workflow Management Coalition (WFMC). This standard, called Workflow Reference Model, supplies a common generic basis for development of interoperability scenarios between different workflow systems [44].

Figure 4 describes how we record this kind of document. The elements *Workflow*, *Activity*, *Atomic Activity*, *Sub-Workflow*, *Dependency*, *Data*, *Role* and *Software* appear in the WfMC reference model [44], and have the standard meanings. *Data Dependency*, *Temporal Dependency*, *Agent* and *User* are new elements introduced in our model to support the needs of environmental applications.

More specifically, a data dependency between two activities is established through exchange of data, with Activity B depending on an Activity A if output data of A constitute input data of B. Temporal dependency determines precedence of execution order of activities in time, e.g. Activity B depends on Activity A if the execution of B cannot start before the ending of A. The notion of sub-workflow allows document reuse – e.g., the plan for determining areas where to cut trees can embed procedures that have been implemented elsewhere.

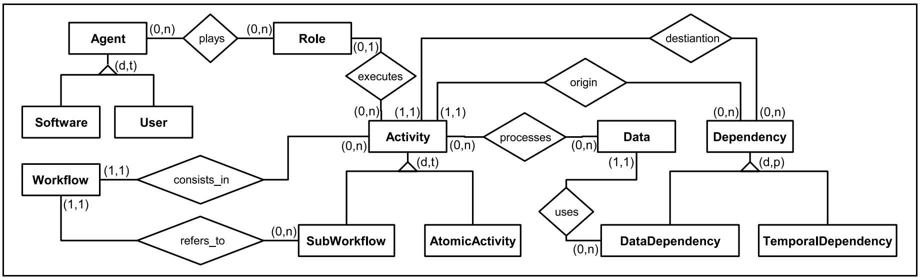


Fig. 4. Specification of a *How-document*

4 Publishing *What-*, *How-*, *Why-documents* on the Web

Section 3 shows how we store environmental planning documentation in a database, associating it with metadata. This Section shows how we publish our documents on the Web, enhancing their semantics with ontological associations. Following Semantic Web standards, we adopt XML to represent our documents, and its schema language, XMLSchema, to define their structure and syntactical constraints. This Section presents the schemata in XMLSchema for each of the document types. The specifications presented are partial because of space restrictions. Full schemas can be found in [29]. Section 6 shows examples of XML documents generated.

4.1 Domain Ontologies and Enviromental Documentation

Ontologies are shared elaborated concepts of knowledge about delimited domains [15]. They describe the meaning of terms, or instances, used in a particular domain, according to their defining concepts, or classes, and the semantic relationships among them. Thus, an ontology specifies the possible uses of data and processes, clarifying the usage scope, or context, for the application of these data.

Connecting documents and their components with ontologies improves their significance, especially geographic and environmental planning related ontologies such as introduced in [15, 14]. This connection can be implemented in a simple way by associating documents and URIs of Web available ontologies.

Our approach to combine domain ontologies and environmental documentation is based on the POESIA approach to handle cooperative processes in the Semantic Web [15]. POESIA relies heavily in two concepts: workflows to compose services, and domain ontologies to provide semantics.

The structure of a domain ontology is divided in dimensions that reflect distributed facets. For instance, for the tree cutting example, a spatial dimension defines classes and terms concerning spatial division concepts, a species dimension contains terms that refer to protected species in the area.

Since a term is an instance of an ontology node, terms are unambiguously defined by an ontology path expression, which specifies a unique path in the ontology structure to reach the node. This expression is specified by the concatenated sequence of *concept*(*term*) vertices visited within the path. As an example, `state(Rio).county(Campos)` is an unambiguous reference in the spatial dimension to a `county` called `Campos` in the `state` called `Rio`. An *ontological coverage* is a tuple of unambiguous references to terms of a POESIA ontology. Two examples of ontological coverages are:

(1) [`country(Brazil)`]
(2) [`country(Brazil).state(Rio)`, `species(Leontopithecus rosalia)`, `species(Caiman latirostris)`]

In *POESIA*, an ontological coverage determines, for one or more dimensions, the context in which the corresponding data and processes are valid. A term encompasses another if, and only if, it refers to a higher level term within the same dimension. This relation is represented by $\omega \models \sigma$, meaning that ω encompasses σ. Following this reasoning, an ontological coverage Γ encompasses another ontological coverage Δ, or $\Gamma \models \Delta$, if, and only if, for every term $\omega \in \Gamma$ there exists a term $\sigma \in \Delta$ such that $\omega \models \sigma$. In the example of ontological coverages, coverage (1) encompasses (2). Furthermore, (2) refers to endangered species (a kind of monkey and a specific alligator) found in Rio, Brazil, and involves two dimensions: territorial divisions and endangered species.

POESIA's specification of domain ontologies supports clear identification of the concepts involved in environmental planning activities. The notation used to denote relationships and terms is amenable to efficient algorithmic processing in XML database systems. Thus we propose their use in combination with *What-*, *How-* and *Why-documents*, enhancing the semantics of their contents. Encompassing relationships helps reuse – e.g., a solution given to a specific context [`country(Brazil)`] can be adapted to a context it encompasses [`country(Brazil).state(Rio)`]. Ontological path expressions can be attached to documents, thereby adding semantics to them.

4.2 *What-Document* Representation

What-documents are specified as hypermedia components, whose Nodes may be distributed on the Web. Their specification for Semantic Web purposes relies on XMLSchema. Domain ontologies can moreover be associated to *What-documents*. A direct mapping from the ER model of Figure 2 can be made to an XMLSchema description. Since this mapping is straightforward because of the similar nature of XML and hypermedia documents, we omit it here.

4.3 *Why-Document* Representation

An XMLSchema specification, partially shown in Table 1, is used for *Why-documents*, mapped from our internal database tables. Ontological references are provided via pointers to an ontological term. More specifically, lines:

“`<xsd:element name="ontologyURI" type="xsd:anyURI" minOccurs="0"/>`” and

“`<xsd:element name="ontologicalCoverage" type="xsd:string" minOccurs="0"/>`”

refer to the URI containing the ontology, followed by the corresponding path expression within the ontology, possibly going down to the term level.

Table 1. XMLSchema: *Why-documents* with ontological references

```
<?xml version = "1.0" encoding = "UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="dR" type="DRType"/>
<xsd:element name="question" type="QuestionType"/>


<xsd:complexType name="DRType">
    <xsd:sequence>
        <xsd:element name="question" type="QuestionType"
                     minOccurs="1" maxOccurs="unbounded"/>
        <xsd:element name="ontologyURI" type="xsd:anyURI"
                                        minOccurs="0"/>
    <xsd:element name="ontologicalCoverage"
                         type="xsd:string" minOccurs="0"/>
        ...
    </xsd:sequence>
    ...
    <xsd:attribute name="dRID" type="xsd:ID"/>
</xsd:complexType>

<xsd:complexType name="QuestionType">
    <xsd:sequence>
        <xsd:element name="drRef" minOccurs="1"/>
        <xsd:element name="isRelatedWithFK" minOccurs="0"
                                          maxOccurs="unbounded"/>
        <xsd:element name="subdividesInFK" minOccurs="0"
                                          maxOccurs="unbounded"/>
        <xsd:element name="objective" type="ObjectiveType"
                           minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="questionString" type="xsd:string"
                                 minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ontologyURI" type="xsd:anyURI"
                                            minOccurs="0"/>
    <xsd:element name="ontologicalCoverage"
                         type="xsd:string" minOccurs="0"/>
        ...
    </xsd:sequence>
    ...
    <xsd:attribute name="questionID" type="xsd:ID"/>
</xsd:complexType>
...
</xsd:schema>
```

4.4 *How-Document* Representation

How-documents use scientific workflows and link processes, activities and data to ontological coverages. Unlike *What-* and *Why-documents*, they are dynamic - i.e., they can be executed and this execution ensures reuse and adaptation of planning procedures. Thus, it is not enough specify them using XMLSchema. Rather, we must choose a language that allows their execution on the Semantic Web. The problem is that, as mentioned in Section 2.5, there are several standard proposals for workflows on the Web, notably XPDL and WSBPEL. We have chosen the latter because it offers more functionality, and better serves our needs.

This Section presents a brief comparison of these two standards that justifies our choice. For more thorough comparative studies the reader is referred to [40, 41, 47, 48, 36]. No proposal, however, offers all features needed by workflow representation standards, and more work needs to be done in this direction.

XPDL presents several problems [41]. The main issue is that the language lacks support for specifying synchronization constraints. Another issue is what happens when more than one source and/or sink is specified. It is clearly possible to create multiple sources and/or sinks in XPLD, but what is actually executed is not clear. Other features that we need are not supported. Among them we can single out: dynamically determining the number of instances of an activity; specifying choices from outside the document, i. e., from environment variables; the possibility of specifying states; and ways to cancel activities or entire workflows.

Even though WSBPEL has more features that suit our needs, it also presents shortcomings. One of them concerns problems in executing an activity following flow merges. Furthermore, WSBPEL is a loop-blocked language. Within its loop constructs (e. g. while loop), it is not possible to have an arbitrary exit point. This prevents changing the current executing loop block for another. In contrast, XPDL supports non-blocked, loop-blocked and full-blocked classes of workflows, following the definition in [45]. Hence, it is possible to define arbitrary exit points within cycles.

Table 2. WSBPEL extended XMLSchema for *How-documents*

```
<?xml version='1.0' encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema"
              xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                                                   xmlns:bpws=
  "http://schemas.xmlsoap.org/ws/2003/03/business-process/"
                                              targetNamespace=
  "http://schemas.xmlsoap.org/ws/2003/03/business-process/"
                            elementFormDefault="qualified">

<import namespace="http://schemas.xmlsoap.org/wsdl/"
        schemaLocation="http://schemas.xmlsoap.org/wsdl/"/>


<complexType name="tProcess">
    <complexContent>
        <extension base="bpws:tExtensibleElements">
            <sequence>
                <element name="partnerLinks"
                  type="bpws:tPartnerLinks" minOccurs="0"/>
                <element name="partners" type="bpws:tPartners"
                                                 minOccurs="0"/>
                <element name="variables"
                        type="bpws:tVariables" minOccurs="0"/>
                <element name="correlationSets"
                  type="bpws:tCorrelationSets" minOccurs="0"/>
                <element name="faultHandlers"
                    type="bpws:tFaultHandlers" minOccurs="0"/>
                <element name="compensationHandler"
              type="bpws:tCompensationHandler" minOccurs="0"/>
                <element name="eventHandlers"
                    type="bpws:tEventHandlers" minOccurs="0"/>
                <xsd:element name="ontologyURI"
                             type="xsd:anyURI" minOccurs="0"/>
                <xsd:element name="ontologicalCoverage"
                          type="xsd:string" minOccurs="0"/>
            </sequence>
            <attribute name="name" type="NCName"
                                              use="required"/>
            <attribute name="targetNamespace" type="anyURI"
                                              use="required"/>
            <attribute name="queryLanguage" type="anyURI"
      default="http://www.w3.org/TR/1999/REC-xpath-19991116"/>
            <attribute name="expressionLanguage" type="anyURI"
      default="http://www.w3.org/TR/1999/REC-xpath-19991116"/>
            <attribute name="suppressJoinFailure"
                               type="bpws:tBoolean" default="no"/>
            <attribute name="enableInstanceCompensation"
                               type="bpws:tBoolean" default="no"/>
            <attribute name="abstractProcess"
                               type="bpws:tBoolean" default="no"/>
        </extension>
    </complexContent>
</complexType>
...
<complexType name="tInvoke">
    <complexContent>
        <extension base="bpws:tActivity">
            <sequence>
                <element name="correlations"
                       type="bpws:tCorrelationsWithPattern"
                              minOccurs="0" maxOccurs="1"/>
                <element name="catch" type="bpws:tCatch"
                      minOccurs="0" maxOccurs="unbounded"/>
                <element name="catchAll"
                 type="bpws:tActivityOrCompensateContainer"
                                             minOccurs="0"/>
                <element name="compensationHandler"
                           type="bpws:tCompensationHandler"
                                             minOccurs="0"/>
                <xsd:element name="ontologyURI"
                          type="xsd:anyURI" minOccurs="0"/>
                <xsd:element name="ontologicalCoverage"
                          type="xsd:string" minOccurs="0"/>

            </sequence>
            <attribute name="partnerLink" type="NCName"
                                           use="required"/>
            <attribute name="portType" type="QName"
                                           use="required"/>
            <attribute name="operation" type="NCName"
                                           use="required"/>
            <attribute name="inputVariable" type="NCName"
                                           use="optional"/>
            <attribute name="outputVariable" type="NCName"
                                           use="optional"/>
        </extension>
    </complexContent>
</complexType>
```

Table 2 shows a partial WSBPEL specification within our framework. We assume, for space saving, that the definitions of the corresponding WSDL document are correctly specified.

5 Semantic Web Environmental Planning Support Tool

This Section discusses issues on implementing a system to support documentation of environmental planning activities and their use on the Web. Our proposal is based on the WOODSS system, which supports documentation management and is being ported to the Web.

5.1 The WOODSS System

The documentation ideas presented were implemented for a mono-user environment in WOODSS (WOrkflOw-based spatial Decision Support System) [35, 21], a software developed at the University of Campinas, Brazil. It was developed on top of Idrisi GIS [9] and tested in several environmental planning efforts.

WOODSS is centered on dynamically capturing user interactions with a GIS in real time, and documenting them by means of scientific workflows. It serves three purposes during environmental planning activities: (i) documentation, for reuse and semantics enhancement; (ii) support for decision making; and (iii) construction of a database that describes solutions to planning processes. This paper concerns documentation issues, and therefore only covers the first aspect. Details on other aspects are covered elsewhere [21, 35].

The dynamically generated scientific workflows correspond to the *How-documents* and are stored in a relational database. Users can manipulate, combine and retrieve these workflows, by accessing this database using WOODSS' graphical interface.

At the same time a workflow (the *How-document*) is constructed in WOODSS, planning experts can, at any time, enter data on *What-* and *Why-documents*, by accessing specific system menus. WOODSS also prompts the user for *What-documents* at the beginning and end of a planning session. Finally, users can also specify a *How-document* at a high level by using the graphical interface, without recurring to a GIS. This means that generic procedures can be stored in a database and made available, to be subsequently specialized for specific GIS implementations.

5.2 Extending WOODSS to the Semantic Web

In order to extend WOODSS to work in compliance with the Semantic Web standards we must work on the data and processes discussed in Section 2.4. Section 4 shows our data representation. We use Web Services to construct software modules to allow cooperative environmental planning on the Web using the documents proposed. This Section outlines how to solve these issues.

From the data point of view, our Web compliant data structures are based on XML. Next, semantic relationships among concepts are specified, which can be done in more than one level. The first level uses RDF for data, and RDFSchema for structure and relationship definitions. However a common vocabulary might be needed by some kinds of semantic relationships. OWL (Web Ontology Language) addresses this problem. We replace ontological specification in a language by references to ontologies within documents. More specifically, our solution is to provide additional semantics by links to ontology services, together with term paths. These services know how to interpret these paths. In particular, we use the OntoCover and OntoCarta [14] tools, developed at UNICAMP. OntoCover is a library implemented in Java™ that supports loading, manipulation and visualization of ontologies, making it easy to create references to ontological coverages. OntoCarta is a software being developed to aid navigation on maps, associating context to a spatial dimension ontology. WOODSS coupling to OntoCover associates processes and data with ontological coverages.

Services require the implementation of the layers shown in Section 2.4. Network layer implementations are commonly available. The XML-based messaging layer is supported with SOAP (Simple Object Access Protocol) compliant

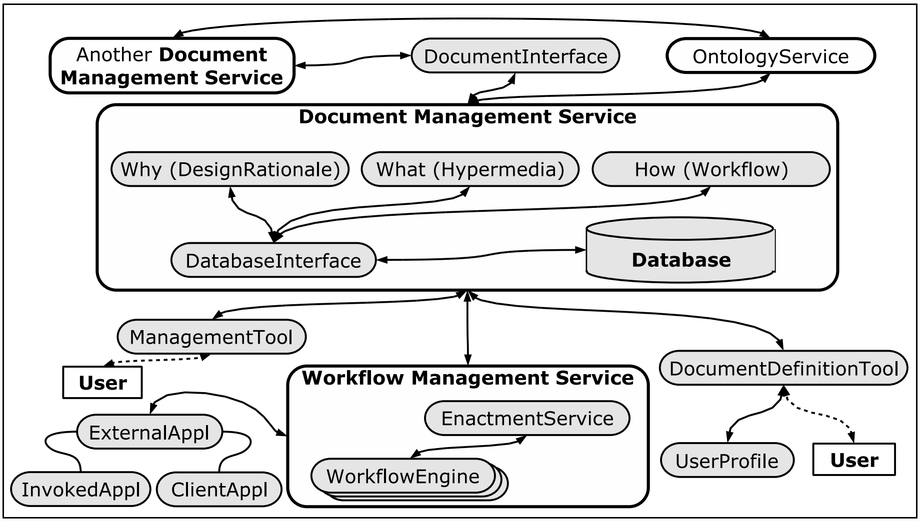


Fig. 5. Document Architecture based on Services

libraries. The service description, publication and discovery layers are provided by standard supporting environment; again, this brings no novelty.

The construction of a Web user interface for WOODSS is another issue to be considered. It involves usability concepts and multiple user management. This discussion is beyond the scope of this work.

Figure 5 shows a high level view of the architecture. It is centered on a Document Management Service that manages document specification and retrieval. The service encapsulates the three kinds of documents, storing them in a relational database. Linkage to ontologies is assured by an interface from the Document Management Service and an Ontology Service that encapsulates the description of a domain ontology. The Document Service also communicates with a Workflow Management Service that extracts the appropriate workflow specifications from it and executes them in workflow engines. Users interact with the Document Service in two ways: via a management tool that supports administrative tasks concerning documents; and via a document definition tool that can be tailored to different user profiles via UserProfile module.

The Workflow Service can invoke external applications (via the ExternalAppl box) and other workflow services. Finally, the Document Management Service can interact with other Document Management Services via a DocumentInterface specification. Each Service blob in the figure can be run at a distinct Web site. Thus, documents can be stored in different databases.

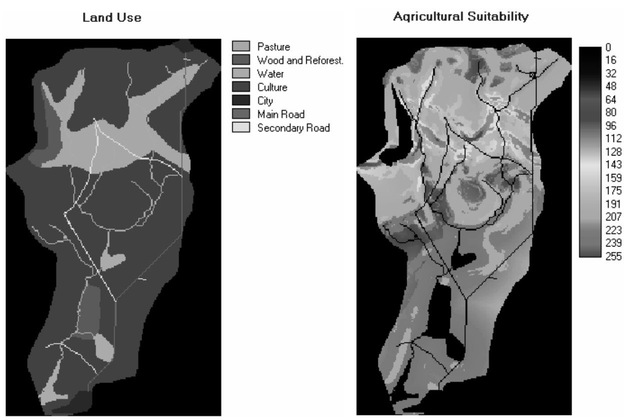
The mono-user version of WOODSS system already supports the functionality of the Document Management Service and its connection to ontologies, workflow management, document definition and management tools. The only external application is the Idrisi GIS. OntoCover and OntoCarta are also implemented and will be encapsulated within an Ontology Service. Thus, the core of our services are already implemented, showing the feasibility of our proposal.

6 Application Example

This Section presents an example of document management within cooperative environmental planning using our proposal.

6.1 Problem Overview

The goal of the problem was to develop an agricultural exploitation plan for a region in Brazil. Planning activities resulted in an agricultural suitability map, showing land suitability in a given region according to a set of relevant parameters. Planners wanted to find areas for agriculture practices within the region, while at the same time taking into account the need for preservation of environmental resources. Input data for solving the problem were maps concerning land use, hydrology, declivity and the result of computing a specific land use model. Figure 6(a) shows the land use map, where the goal region (Iracemápolis microbasin) has areas occupied by pasture, wood and reforestation, water, culture, cities, main and vicinal roads. The capacity of use map was generated by another planning process, previously documented in WOODSS. The declivity map has declivity scales within the area. For these inputs, the problem was solved with support of a GIS and from the solution a *How-document* specification was generated.



(a) (b)

Fig. 6. Microbasin of Iracemapolis: (a) Land Use (b) Agricultural Suitability

The result of the planning process is the agricultural suitability map illustrated in Figure 6(b). The best areas for agriculture are those that present the higher values of the scale of values. Areas in solid black cannot be used.

The planning procedure was executed with support of Idrisi GIS. The implementation process consisted on producing several maps with distinct weight

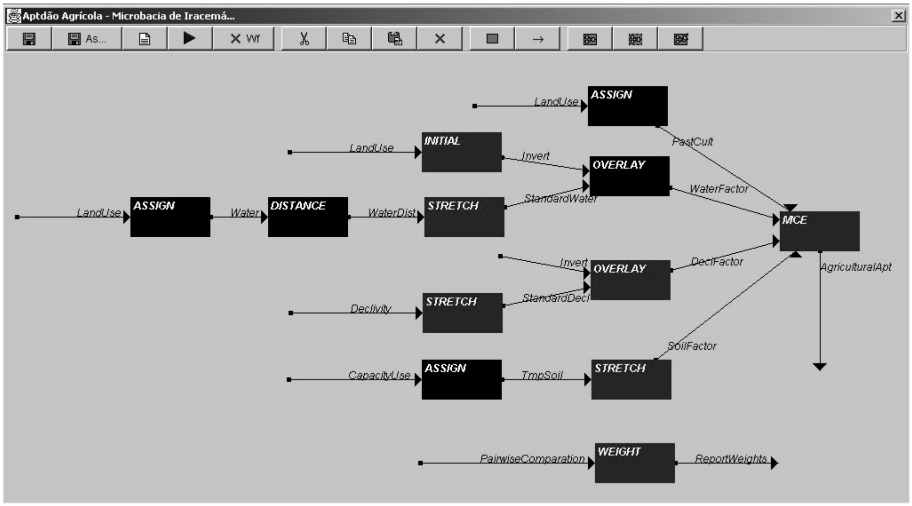


Fig. 7. WOODSS – *Workflow* for *How* Documentation of the example

factors, and overlaying them. Details about functions and the parameters used in each step can be found in [34]. Figure 7 shows the workflow corresponding to the implemented procedure, dynamically generated by WOODSS. In this workflow (executable *How-document*), activities are Idrisi functions and data are files processed by functions. There follows part of the documentation associated with the procedure, along with Semantic Web documents samples.

6.2 *What* Documentation

Figure 8 shows part of the problem's *What* documentation, represented by a hypermedia network of nodes-and-links. This linked structure can be arbitrarily extended to any level of detail (e.g., pointing to formulae and multimedia data). In this example, the main document node (left top corner) describes the general problem. This node is linked to another node that describes the methodology used to solve the problem. This second node contains three visible anchors:

– *next to water*, that points to another node that describes how distance from water was calculated;
– *classification in capacity of use system*, that points to a node describing land classification according to the capacity of use model;
– *lesser declivities*, that points to a node containing a textual description about the procedure used for computing declivity.

In the extended Semantic Web context, each hypermedia node can be in a distinct site, constructed by different users in an asynchronous fashion. Node contents are described in XML, following the XMLSchema specification of Section 4.

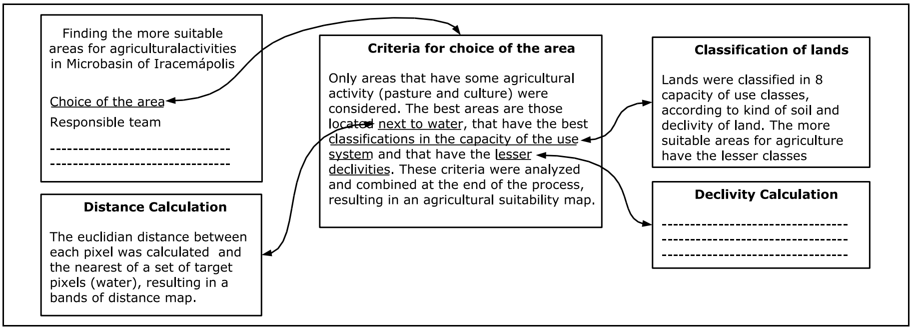


Fig. 8. Partial *What* Documentation for Production of a Solution

6.3 *Why* Documentation

Figure 9 shows part of the *Why* documentation associated with the problem. It describes discussions and decisions related to the choice of restrictions and factors to be considered in solving the decision problem – namely, to find suitable areas for agricultural practices while considering environmental factors. The document is shown as a directed graph. Capital bold letters indicate elements of our design rationale model. (**Q** = question, **SQ** = subquestion, **O** = objective, **A** = alternative, **Ad** = advantage, **D** = disadvantage and **R** = risk). Boxed alternatives were the ones chosen for solution of the corresponding question.

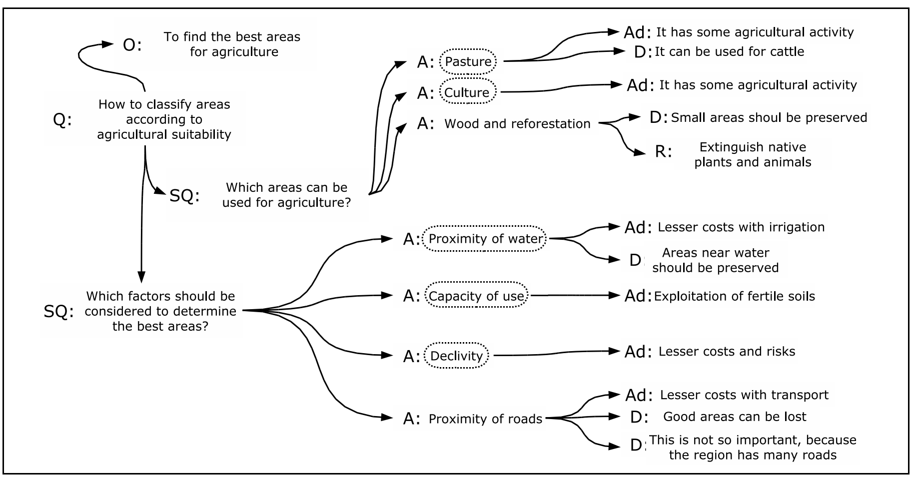


Fig. 9. Partial *Why* Documentation for Problem Solution

Discussion starts with a general question: "How to classify areas according to agricultural suitability?" The objective of this question is *to find the best areas for agriculture*. This complex question is divided into two subquestions:

– Which areas can be used for agricultural practices?
– Which factors should be considered to determine the best areas?

The first subquestion can be answered by three alternatives: Pasture, Culture and Wood; each with advantages, disadvantages and risks. The rest of the figure can be described in the same way.

Table 3. Example of *Why-document* partially translated to XML from Figure 9

```
<?xml version='1.0' encoding="UTF-8"?>

<dR dRID="iracemapolisAreaClassif">
    <question questionID="areaClassif">
        <drRef>iracemapolisAreaClassif</drRef>
        <subdividesInFK>agriUsable</subdividesInFK>
        <subdividesInFK>determFactors</subdividesInFK>
        <objective>To find the best areas for ariculture
        </objective>
        <questionString>How to classify areas according
                               to agricultural suitability?
    </questionString>
    <ontologyURI>
    http://lis.ic.unicamp.br/:8040/ontocover/assess-1034
    <ontologyURI/>
    <ontologicalCoverage>[country(Brazil).state(SaoPaulo),
                          species(Leontopithecus rosalia),
                              species(Caiman latirostris)]
    </ontologicalCoverage>
    </question>
    <question questionID="agriUsable">
    <drRef>iracemapolisAreaClassif</drRef>
    <questionString>Which areas can be used for
                                           agriculture?
    </questionString>
    <alternative>
        <altDescription>Pasture</altDescription>
        <advantage>It has some aricultural activity
        </advantage>
        <disadvantage>It can be used for cattle
        </disadvantage>
    </alternative>
    <alternative>
        <altDescription>Culture</altDescription>
        <disadvantage>It has some agricultural activity
        </disadvantage>
    </alternative>
    <alternative>
        <altDescription>Wood and reforestation
        </altDescription>
        <advantage>Short areas that should be preserved
        </advantage>
            <risk>Extinguishiment of native plants and animals
            </risk>
    </alternative>
    </question>
    <question questionID="determFactors">
    <drRef>iracemapolisAreaClassif</drRef>
    <questionString>Which factors should be considered
                          to determine the best areas?
    </questionString>
    <alternative>
        <altDescription>Proximity of water</altDescription>
        <advantage>Lesser costs with irrigation</advantage>
        ...
    </alternative>
    ...
    </question>
    <ontologyURI>
    http://lis.ic.unicamp.br/:8040/ontocover/assess-1034
    <ontologyURI/>
    <ontologicalCoverage>[country(Brazil).state(SaoPaulo),
                          species(Leontopithecus rosalia),
                              species(Caiman latirostris)]
    </ontologicalCoverage>
...
</dR>
```

Why-documents are usually centralized, but can be updated by users in different locations. The need for an XML representation of this kind of document goes in the directions of integration, reuse and attaching semantics to the data, which can be embedded in SOAP messages. Table 3 shows the XML representation for our example for the *Why-document*.

6.4 *How* Documentation

Figure 10 shows the *How-document* for the problem, representing the procedure used to solve the problem. This workflow is identical to the one generated by WOODSS (Figure 7); however, components (activities, data and dependencies) are annotated by experts with indications that facilitate the understanding of *How*. Documentation annotation is available in WOODSS.

For example, the longest sequence of activities (third from top to bottom in Figure 10) indicates that: (i) input is the land use map; (ii) the first activity

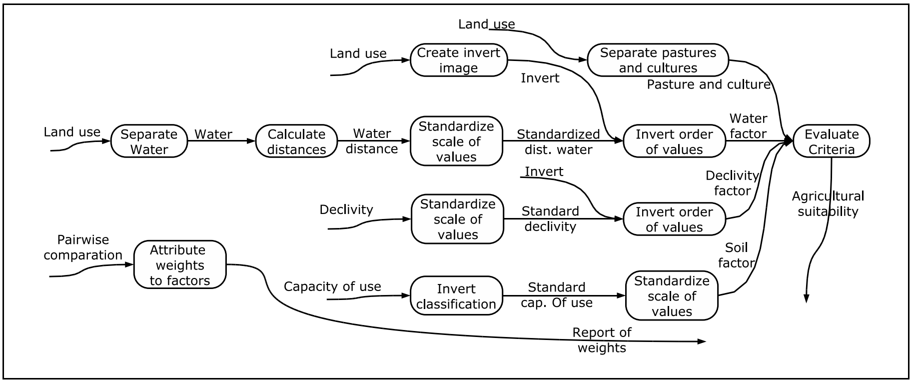


Fig. 10. *How* Documentation for the Problem

(implemented via ASSIGN in Idrisi GIS – see WOODSS workflow, Figure 7) separates water from other elements in the land use map with the water map being passed on to the next step; (iii) the goal of the second activity (DISTANCE in Idrisi GIS) is to compute distance buffers from each point of the region in relation to water; (iv) the third activity (STRETCH in Idrisi GIS) standardizes scales of values of the water distance map, allowing subsequent comparison of all considered factors.

Table 4. Example of *How-document* partially translated to XML from Figure 10

```
<!-- BPEL4WS process definition -->
<process name="agriculturalSuitability"
         targetNamespace="http://lis.ic.unicamp.br/woodss"
                                                      xmlns=
  "http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                            abstractProcess="yes">
...
<partnerLinks>
      <partnerLink name="idrisiCaller"
                        partnerLinkType="activityLinkType"
                                  myRole="opCaller"/>
      <partnerLink name="idrisiGIS"
                        partnerLinkType="activityLinkType"
                             partnerRole="opResponder"/>
      ...
</partnerLinks>
...
<flow>
    <flow>
    <sequence>
        <invoke  partnerLink="idrisiGIS"
                 portType="idrisiCallsPT"
                 operation="assignMap"
                 inputVariable="inMapPath"
                 outputVariable="outMapPath">
                 <target linkName="assign-to-gis"/>
                 <ontologyURI>
       http://lis.ic.unicamp.br/:8040/ontocover/assess-1034
         <ontologyURI/>
         <ontologicalCoverage>[country(Brazil)]
    </ontologicalCoverage>
                 ...
        </invoke>
            ...
            <invoke  partnerLink="idrisiGIS"
                     portType="idrisiCallsPT"
                     operation="distanceMap">
                     ...
            </invoke>
            ...
        </sequence>
        ...
        </flow>
        <invoke  partnerLink="idrisiGIS"
                 portType="idrisiCallsPT"
                 operation="overlayMaps"
                 inputVariable="inMapPath">
                 ...
        </invoke>
        ...
</flow>
<invoke  partnerLink="idrisiGIS"
         portType="idrisiCallsPT"
         operation="evaluateMCE"
         inputVariable="inMapPath">
         ...
</invoke>
...
<ontologyURI>
http://lis.ic.unicamp.br/:8040/ontocover/assess-1034
<ontologyURI/>
<ontologicalCoverage>
[country(Brazil).state(SaoPaulo)]
</ontologicalCoverage>
</process>
```

On the Web context, activities or parts of the workflow can be executed in distinct sites, using various GIS tools. Again, this can be supported by mapping the workflow definition to WSBPEL, as explained in Section 4, and annotating the workflow using XML. Furthermore, each activity and data dependency can refer to an ontology node. Providing this in the Web requires replacing ontology references by [URI, path expression], where the URI points to the ontology server and the path expression to the term within the ontology, as shown in Table 4.

We must point out the essential difference between our proposal and model builder tools such as those provided by ESRI^{TM}[12] software packages. Similar to WOODSS, these packages capture user activities and show them as “workflows” that can be re-executed. However, WOODSS stores these specifications within documentation database tables, thus fostering interoperability and reuse. Therefore, our proposal supports a generic implementation, regardless of the target GIS. First, the document database can be shared and updated by several users simultaneously. Second, the database can store specifications generated for any GIS, since it implements the *Why-document* model of Section 3. A single document database can therefore house models specified within distinct software packages – the only additional requirement is to develop specific modules to encode and decode the commands for each GIS. Our implementation has just one such module – for Idrisi. Extending it to other GIS requires as many additional modules, but data are stored in one database. Finally, generic *How* specifications (such as those of Figure 10) can be defined graphically and stored in the document database, being linked to *What* and *Why* documents, ontologies and metadata. Those generic documents can be exchanged among GIS Web users of any GIS, to be subsequently refined into specific implementations.

7 Concluding Remarks

This paper proposed a framework to support documentation of environmental planning activities in the Semantic Web. It presented three kinds of documents generated during environmental planning: description of the problem to be solved and the associated plan (*What*), description of the process used to obtain the plan (*How*) and description of the reasons behind the planning decisions (*Why*). *What* documents were represented through hypermedia and metadata, *How* documents through scientific workflows and *Why* documents through design rationale.

A mono-user version of this proposal has already been implemented as part of a spatial decision support system – WOODSS – developed at UNICAMP. WOODSS is being extended to meet Semantic Web standards, including connection to ontology terms.

The main contributions are centered on proposing specific document structures for supporting cooperative environmental planning on the Web and an architecture based on Web Services to manage these documents. Documents and processes are linked to each other and associated with geographical metadata and domain ontology terms. Thus, the documents become not only a means of supporting cooperation on the Web, but also of lending more semantics to it.

Another contribution is showing a practical application of the proposal within the Semantic Web context.

Ongoing work involves implementation and theoretical issues. At present, we are implementing the modules responsible for managing the three kinds of documents for the Semantic Web. Issues on the system's user interface should also be considered, given the Web context. This means the interface must support distinct kinds of user profiles and cultures that cooperate on the Web.

Uncertainty is a very important issue in any planning procedure. The present stage of our work does not consider documenting this kind of factor, except via user textual entries in *Why* documentation. Thus, an extension is to provide support to registering probabilities associated with possible outcomes, and use this to help the decision process.

Another extension concerns additional documentation means – e.g., using voice records and video of meetings. These, for example, could be used to generate *Why* documents. Finally, document integration exists at the database level, but must be better reflected at the interface level, to help users query across documents with less navigational effort.

Acknowledgements

This work was developed partially financed by FAPESP, CNPq, the SAI project - Advanced Information Systems - of PRONEX-MCT, as well as WebMaps and AgroFlow CNPq projects. We also thank A. Santanchè, L. Digiampietri and the reviewers for their insightful comments.

References

1. A. Ailamaki, Y. Ioannidis, , and M. Livny. Scientific Workflow Management by Database Management. In *Proc. of 10th IEEE Int. Conf. on Scientific and Statistical Database Management*, pages 190–201, 1998.
2. K. M. Anderson, R. N. Taylor, and E. J. Whitehead Jr. Chimera: hypermedia for heterogeneous software development enviroments. *ACM Transactions on Information Systems (TOIS)*, 18(3):211–245, 2000.
3. A. Arkin. Business Process Modeling Language. Technical report, BPMI.org, 2002. www.bpmi.org (as of Oct 2004).
4. BPEL4WS. Business Process Execution Language for Web Services Version 1.1. Technical report, BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003. www-106.ibm.com/developerworks/webservices/library/ws-bpel/ (as of Oct 2004).
5. T. X. Bui and S. R. Sankaran. Design Considerations for a Virtual Information Center for Humanitarian Assistance/Disaster Relief using Workflow Modeling. *Decision Support Systems*, 31(2):165–179, 2001.
6. J. E. Burge and D. C. Brown. Reasoning with Design Rationale. In *Artificial Intelligence in Design'00*, pages 611–629, 2000.
7. M. A. Casanova, T. A. S. Coelho, M. T. M. Carvalho, E. T. L. Corseuil, H. Nobrega, F. M. Dias, and C. H. Levy. The Design of XPAE - An Emergency Plan Definition Language. In *IV Brazilian Geoinformatics Symp.*, pages 25–32, 2002.

8. F. Casati and A. Discenza. Supporting Workflow Cooperation Within and Across Organizations. In *ACM Symp. Applied Computing*, volume 1, pages 196–202, 2000.
9. Clark Labs. Geographic Analysis and Image Processing Software. www.clarklabs.org (as of Oct 2004).
10. S. Coloudre, T. Libourel, and L. Spéry. Metadata and GIS: a classification of metadata for GIS. In *1st Int. Conf. and Exhibition on Geographic Information*, 1998.
11. J. Conklin and M. Begeman. gIBIS: A Hypertext Tool for exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 6(4):303–331, 1988.
12. ESRI. GIS and Mapping Service. www.esri.com (as of Oct 2004).
13. Federal Geographic Data Committee. Content Standard for Digital Geospatial Metadata (CSDGM). www.fgdc.gov/metadata/contstan.html (as of Oct 2004).
14. R. Fileto. *The POESIA Approach for Services and Data Integration On the Semantic Web*. PhD thesis, IC–UNICAMP, Campinas–SP, 2003.
15. R. Fileto, L. Liu, C. Pu, E. D. Assad, and C. B. Medeiros. POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture. *The VLDB Journal*, 12(4):352–367, 2003.
16. G. Fischer, A. Lemke, R. McCall, and A. Morch. *Design Rationale Concepts, Techniques, and Use*, chapter Making Argumentation Serve Design, pages 267–294. Lawrence Erlbaum Associates, 1995.
17. F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, 1994.
18. G. P. Heliades and E. A. Edmonds. Notation and Nature of Task in Comprehending Design Rationale. *Knowledge Based Systems*, 13(4):215–224, 2000.
19. D. L. Hicks, J. J. Legget, P. J. Nürneberg, and J. L. Schnase. A Hypermedia Version Control Framework. *ACM Trans. on Information Systems*, 16(2):127–160, 1998.
20. R. D. Holowczak, S. A. Chun, F. J. Artigas, and V. Atluri. Customized geospatial workflows for e-government services. In *Proc. of the 9th ACM Int. Symp. on Advances in Geographic Information Systems*, pages 64–69, 2001.
21. D. Kaster, C. B. Medeiros, and H. Rocha. Supporting Modeling and Problem Solving from Precedent Experiences: The Role of Workflows and Case-Based Reasoning. *Environmental Modeling and Software*, 2004. Accepted for publication.
22. F. Leymann. Web Services Flow Language (WSFL 1.0). Technical report, IBM, 2001. www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf (as of Oct 2004).
23. A. MacLean, R. M. Young, V. Bellotti, and T. Moran. Questions, Options, and Criteria: Elements of design space analysis. *Human-Computer Interaction*, 6(3&4):201–250, 1991.
24. A. MacLean, R. M. Young, and T. P. Moran. Design Rationale: The Argument Behind the Artifact. In *Proc. of the SICCHI Conf. on Wings for the mind*, pages 247–252, 1989.
25. S. R. Monk, I. Sommerville, J. M. Pendaries, and B. Durin. Supporting Design Rationale for System Evolution. In *European Software Engineering Conference (ESEC95)*, pages 307–323, 1995.
26. K. Grønbæk and R. H. Trigg. Design Issues for a Dexter-based Hypermedia System. *Communications of the ACM*, 37(2):40–49, 1994.
27. OASIS Open. OASIS Web Services Business Process Execution Language Technical Committee. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel (as of Oct 2004).

28. J. Oliveira, M. Gonçalves, and C. B. Medeiros. A Framework for Designing and Implementing the User Interface of a Geographic Digital Library. *International Journal of Digital Libraries*, 2(2-3):190–206, 1999.
29. G. Z. Pastorello Jr. Publishing and Integrating Scientific Workflows on the Web (in Portuguese). Master's thesis, IC–UNICAMP, Campinas–SP, 2004.
30. A. Peerbocus, C. B. Medeiros, A. Voisard, and G. Jomier. A System for Change Documentation based on a Spatiotemporal Database. *Geoinformatica*, 8(2):173–204, 2004.
31. C. Petrie and S. Sarin. Internet-based Workflows – Special issue. *Internet Computing*, May-June 2000.
32. I. Rauschert, P. Agrawal, R. Sharma, S. Fuhrmann, I. Brewer, and A. MacEachren. Designing a human-centered, multimodal GIS interface to support emergency management. In *Proc. of the 10th ACM Int. Symp. on Advances in Geographic Information Systems*, pages 119–124, 2002.
33. S. M. Resende. Database-centered Documentation of Environmental Planning Activities (in Portuguese). Master's thesis, IC–UNICAMP, Campinas–SP, 2003.
34. H. A. Rocha. Metadata for Scientific Workflows in Environmental Planning Support (in Portuguese). Master's thesis, IC–UNICAMP, Campinas–SP, 2003.
35. L. Seffino, C. B. Medeiros, J. Rocha, and B. Yi. WOODSS - A Spatial Decision Support System based on Workflows. *Decision Support Systems*, 27(1–2):125–123, 1999.
36. R. Shapiro. A Technical Comparison of XPDL, BPML and BPEL4WS. xml.coverpages.org/Shapiro-XPDL.pdf (as of Oct 2004).
37. M. Singh and M. Vouk. Scientific Workflows: Scientific Workflow Meets Transactional Workflow. In *NSF Workshop on Workflow and Process Automation in Information Systems: State of the Art and Future Directions*, 1996.
38. S. Thatte. XLANG – Web Services for Business Process Design. Technical report, Microsoft, 2001. www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm (as of Oct 2004).
39. The OWL Services Coalition. OWL-S 1.0 Release. www.daml.org/services/owl-s/1.0/ (as of Oct 2004).
40. W. M. P. van der Aalst. Don't go with the flow: Web services composition standards exposed. *IEEE Intelligent Systems*, 18(1):72–76, 2003.
41. W. M. P. van der Aalst. Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. Technical report, Queensland University of Technology, 2003. QUT FIT-TR-2003-06.
42. W3C. The World Wide Web Consortium. www.w3.org (as of Oct 2004).
43. J. Wainer, M. Weske, G. Vossen, and C. B. Medeiros. Scientific Workflow Systems. In *Proc. of the NSF Workshop on Workflow and Process Automation Information Systems*, 1996.
44. WfMC. The Workflow Reference Model. Technical report, Workflow Management Coalition, 1995. TC-1003.
45. WfMC. Workflow Process Definition Interface – XML Process Definition Language. Technical report, Workflow Management Coalition, 2002. TC-1025.
46. E. J. Whitehead Jr. Uniform Comparison of Data Models Using Containment Modeling. In *Proc. of 13th Conf. on Hypertext and Hypermedia*, pages 182–191, 2002.
47. P. Wohed, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede. Analysis of Web Services Composition Languages: The Case of BPEL4WS. In *Proc. of the 29th EUROMICRO Conf.*, pages 298–305, 2003.

48. Workflow Patterns. tmitwww.tm.tue.nl/research/patterns/ (as of Oct 2004).
49. WSCI. Web Service Choreography Interface 1.0. Technical report, W3C, BEA Systems, Intalio, SAP AG , Sun Microsystems, 2002. www.w3.org/TR/wsci/ (as of Oct 2004).
50. W. Xin and X. Guangleng. Supporting design reuse based on integrated design rationale. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 3, pages 1909–1912, 2001.