Hydrol. Earth Syst. Sci. Discuss., 4, S978–S986, 2007
www.hydrol-earth-syst-sci-discuss.net/4/S978/2007/

Hydrology and Earth System Sciences Discussions

Interactive comment on "Web services for distributed and interoperable hydro-information systems" *by* J. Horak et al.

J. Horak et al.

Received and published: 11 September 2007

We are thankful to the anonymous referee for his/her valuable comments and questions, which have helped us to discover parts, which are difficult to understand, weakly explained or not documented. The topic is multidisciplinary and it aggravates how to find a balance between the length of paper and comprehensive explanation of all relevant aspects. The answers to comments and questions are provided below; original comments are in italic. The corresponding changes and improvements are incorporated in the final version of the paper.

General comments

This paper manages to convey the feeling that the authors have, in T-DSS, constructed in interesting system. The use of a distributed system architecture to address certain difficult realities of trans-boundary water management is itself a worthy subject, while the particular architecture of T-DSS and the specifics of implementation are also of interest. The paper is however too short to do these subjects justice. In four sides, the authors attempt to provide background, survey related work, introduce the T-DSS system, and present examples of the integration of modelling components. In the end none of these aspects is adequately treated, and a clear picture of the use of T-DSS in a real, trans-boundary decision context is not established.

T-DSS is not intended to be applied only for trans-boundary water management. The paper should attract the developers of hydroinformation system to utilisation of web services as well as end-users to give them an idea about possibilities of such a distributed system application. Unfortunately it is difficult to provide detail description of all related aspects of the multidisciplinary topic.
In the modification of the final version we have extended several parts to provide better view of architecture aspects, possible case studies, end-users and future implementation.

The paper touches in the introduction on the critical issue of the relationship between the architecture of software systems and the social (including institutional) context in which that software must be deployed and used. I would prefer to see this important matter given a more thorough treatment.

Improvements are made in several relevant places in the text. It refers to better characterisation of local and global distributed systems, notices in description of case studies and extension of the conclusion.

HESSD

4, S978–S986, 2007

Specific comments

The main body of the paper is a bit of a "jargon soup", and will I think be difficult or impossible for many HESS readers to understand. The higher level issues are likely to be of greater interest than for example details of the implementation of particular simulation services (e.g. that ModFlow is mounted on a Linux system and the Perl language is used to expose it as a service).

The technical details are provided to document specific abilities of web services based system (in this example to provide an evidence of multiplatform integration and application of various programming languages). We think, it corresponds to the main objectives of the paper mentioned above. Unfortunately we feel a common lack of experiences of web service based (hydro-)information systems to be able to practically document higher level issues.
The final version is revised and extended to improve the understanding with respect to this comment. We have tried to eliminate ”jargon” by improving the description or by deleting unnecessary specific terms.

On the other hand, those with the technical background to understand the details are likely to be interested in specifics such as the message formats used to communicate with services. SOAP defines a standard "envelope"; it says nothing about the contents of this envelope. I would not suggest that any attempt is made to describe these in this paper, but it would be useful if references were provided to any available documentation of these (are existing standards used, or ad-hoc XML scemata? how general are they? does each modelling service use a different message format?).

The description of web services is revised in the final paper and references to respective specifications are provided. Specifications represent a basic standardisation level

HESSD
4, S978–S986, 2007

and are sufficient for the web service implementation. Nevertheless, the experiences show, that more standardization is needed, especially for exchange format. A short comment to web services application is incorporated to the conclusion.

A discussion of the work needed to turn T-DSS into a practical tool for integrated water management is needed. There will surely be both technical and social (including institutional) problems to solve. Who will set up compatible modelling services, how will use of these services be controlled, and how will their running costs be recovered, for example? The cost recovery question will surely be answered differently by different service providers.

The development of distributed information systems relay on the coordination of individual systems and cooperation of relevant subjects. Nevertheless due to the flexibility of web services individual services can be easily substituted. The T-DSS can provide e.g. for individual user/groups different modelling services according to their license policy and the existence of individual agreements between data/service provider and the end-user. These issues are addressed in the conclusion of the final version.

The authors state that "T-DSS incorporates a framework for building web based applications". Can any examples of such applications be provided? Is T-DSS a decision support system, or a framework for building such systems? The name suggests the first interpretation, while the description, particularly including this statement about web based application, is more consistent with the latter.

T-DSS is a DSS. T-DSS incorporates a framework, which is called ArteGIS. ArteGIS provides specialised web mapping components.

HESSD
4, S978–S986, 2007

EGU

This passage seems to be less important, and to avoid ambiguity, it is deleted from the final version.

The second page of the paper begins with the statement, "The architecture of the RODOS DSS is N-tier and Plug-in based, with an Object Relational Mapping solution for system data management and GIS subsystem." This sentence carries very little information even to those who are familiar with the terms "n-tier", "plug-in", and "object relational mapping" (none of which terms require capitalisation). For most of the readership of HESS, who will not be familiar with software-architectural concepts and jargon, it will be entirely meaningless. The authors then inform us That RODOS utilises some class called IModel for model management, but that "the class and its role are not fully documented". Since it seems that the authors have no real idea what the IModel class does, it is not clear why they decided to mention it.

The introduction of the architecture of distributed system (chapter 2) contains more information about basic concepts of architecture, including n-tier one. RODOS DSS represents another local system solution different from OpenMI application that is why it is shortly documented.
The description of RODOS DSS is revised in the final text.

These are examples of a more general problem. The authors make many statements which carry little meaning. What is/are "document attached info"? "WMS"? "Info panels"? What does the second paragraph of section 2 mean? A word has surely been omitted from the first sentence, but the second is also problematic. Solutions to what problem? How, why, when should parts of a distributed system be interconnected "more tightly"?

HESSD
4, S978–S986, 2007

We are sorry to make unclear statements and terms, taken from the T-DSS development. The text is revised and modified for better description of these features. WMS (web map service) is described in chapter 3.

In the discussion of OpenMI, we are told that "OpenMI creates an independent multi-platform environment above modelling systems". Independent of what? Multi-platform in what way? Just two sentences later we are told that "OpenMI does not possess sufficient independence ...". It is still not clear what this independence is from, but this statemenet seems to conflict with the earlier. Then "it is impossible to connect wrappers prepared in C# and Java": so OpenMI isn't multi-platform; there are simply two OpenMIs. What is "variable marking"?

OpenMI specification is independent (not vendor, not platform meaning kind of hardware and operation system, and not programming language depending), but existing implementation tools (called OpenMI environment) cannot be specified as independent.
The text is revised in the final version.

While the authors will understandably not wish to reproduce the findings of the TANDEM project (ref Tyler et al., 2004) in full, an indication of why those findings are relevant to the discussion is certainly needed. Potential users, we are told, indicated a desire for a system which is "freely available". It would be interesting to know why. Is this really to do simply with the cost of licenses, does it derive from a rational analysis of whole life costs, or does it have to do for example with flexibility and lack of vendor lock-in (so "free" in the sense used by the Free Software Foundation: free as in "free speech" rather than "free beer").

HESSD
4, S978–S986, 2007

T-DSS (and other web service based information system) is not tied with free of charge software. The software selection reflects criteria of required functions for end-users, minimal technical requirements (the existence of distant control mechanism) and license requirements (the license policy has to enable an utilization in the internet).
If the software is provided free of charge (and with low license restrictions), it is an important advantage for developers of applications like T-DSS; it is evitable side effect. The benefits can be seen in:

- better license conditions (e.g. no limitations of the number of installations for testing and operating; no special restrictions of software exploitation)
- lower dependency of the total system cost on external prices, where no prices for modelling software are copied to the final product. It means that a software developer can better manage the pricing policy.
- ability to satisfy a specific end-users policy like administrative rules, which lead to a preference of systems with low investment costs and acceptable higher service costs or personal costs

From the end-user's point of view, the cost of the system is the critical issue. The explanation of the reasons, why the cost is so highlighted and why free software is preferred by a part of end-users, is out of the range of the paper. The simple answer does not exist - it is mixture of "objective" as well as psychological reasons, which cannot be easily labelled as irrational, because they are rational e.g. from the personal or administrative point of view.

Was the simpler REST style of web services considered? Is the complexity of WS- justified?*

HESSD
4, S978–S986, 2007

The key to the REST methodology is to write Web services using the URI as an interface. The REST offers to apply a technology, which is already well known and widely used.
Nevertheless, the REST methodology has been evaluated as insufficient for the systems, where a large volume of data is exchanged. E.g. particular Web Services, running in the frame of T-DSS, often receive spatial data and XML based documents. Using REST can be useful where the simple data types are used (like integer id=6 or string name=Josef). The problem is in the case of serialized objects or complex data type in general (both used in the frame of T-DSS). Using the RESTs URI for sending such data is not appropriate for this case.

Fig 1 suggests that a "business logic" server coordinates all communication in the system, except for a direct link between data store and spatial data processing service. The term "business logic" is a fuzzy one in its original context of general purpose (business supporting) computing, and it seems inappropriate here. Whatever term is used, an explanation of the role(s) of this component of the system is needed.

The business logic basically describes the functional algorithms, which handle information exchange between a database and a user interface. But the term has several meanings including modelling business objects, prescribes how business objects interact with one another and enforces the routes and the methods by which business objects are accessed and updated.
The application of this term is not necessary in our paper and it is eliminated from the final version.

Technical corrections

The figures are too small. Fig 2 in particular uses much too small a font.

HESSD
4, S978–S986, 2007

Figure 2 is changed.

HESSD
4, S978–S986, 2007