Design Considerations for Participatory Gis

Piotr Jankowski
Department of Geography
San Diego State University
piotr@typhoon.sdsu.edu

SUMMARY

Development of a participatory GIS (PGIS) can be described as the process of creating, within the constraints imposed by the development environment, a system providing end-users with maximum functionality that can be effectively applied to a decision task. Although it may at first appear that constraints upon such a task are minimal, in practice there are many factors that need to be considered early in the process. This paper is an attempt to identify several of these constraints, and briefly describe how they were encountered in the process of developing a prototype system called WaterGroup that was used in a collaborative water resource planning.

KEYWORDS: *GIS, public participation, spatial decision support systems*

INTRODUCTION

Traditionally, information system development proceeded through a series of stages known as the system development life cycle. This process typically begins with a requirements definition phase, followed by logical and then physical designs, and finally an implementation phase. More recently, increasing system complexity, uncertain requirements, and reduced cycle times have tended to steer developers toward a more iterative process in which the final system evolves from basic core functionality toward a fully functional system.

In an idealized world, where all requirements are known up front and there are no physical constraints placed upon the final system, a sequential approach to design can prove to be efficient. But when responding to a resource constrained, uncertain world with an iterative approach, early iterations impose constraints upon those that follow, and the developer is no longer free to "start from scratch" when designing and implementing functionality. Attempting to force fit sequential design concepts upon an iterative approach can result in an inefficient development process and ineffective information system.

This paper might be better titled "Consider a*ll* your constraints up front". It is an attempt to touch upon a variety of constraints that may well be germane to other PGIS applications, and an effort to illustrate how these constraints were (mostly) dealt with in the context of a prototype system called WaterGroup. WaterGroup is a prototype PGIS designed to assist stakeholders with collaborative water resource management in the Boise River basin of southwest Idaho. Papers detailing the context of the decision problem, the experimental setting in which the prototype was utilized, and the analyses conducted on the experimental results will be forthcoming.

REQUIREMENTS DEFINITION

Problems subject to PGIS support are often ill-defined or partially structured (Jankowski and Nyerges, 2001). Frequently the problem solving process taken dictates the final solution. Although the desired endpoint is typically some type of decision, the nature of the decision as well as the steps taken to get to it cannot be defined in advance. Rather than designing a system to reach a specific endpoint, it is sometimes preferable to think in terms of actions that might be taken to help along the way. These actions could then be realized in the software as discrete functions. Participants could then choose whether to perform a given action, or in what order to perform actions, based upon the specific context of the problem solving process in which they are involved. This discretization points toward modular development, where potential actions are realized by functions or modules that can be invoked as needed (or be utilized in future systems/applications).

For the WaterGroup prototype, potential participant actions were grouped into functions including (2D and 3D) static visualization, solution option generation, what-if analysis, voting and group communication. Although development began with the intent of making each function group modular, much of the modularity was abandoned in favor of efficient coding. The need to define variables in each module and pass these between modules was alleviated by defining variables once with a public scope so that multiple functions could access them. The rationale for this decision was that many of the functions did not make sense as stand-alone modules, and developer resources were limited.

One other constraint that might be planned for in terms of requirements definition: For WaterGroup, the researchers were physically separated and working within different organizations. The logistics of scheduling common time/place meetings forced much of the requirements definition work to be conducted asynchronously via "chain" emails. Although this approach is workable, it may not produce the quality achievable with same place/same time interaction. It seems that many PGIS applications might be subjected to similar constraints, and might take extra efforts to maximize synchronous communication.

DECISION PROBLEM CHARACTERISTICS

As mentioned above, although the desired endpoint of the process is frequently a decision, the nature of the process is uncertain. This uncertainty can lead to a situation where data are nonexistent or simply lacking in the PGIS. Frequently at the start of a decision process participants don't know "what they don't know". Only by going through the decision process are they able to identify areas where data needs to be included in the system. And often data that could assist participants is simply not available. In such cases is can be helpful to attempt to identify proxy measures that could be used to approximate missing data. But this runs the risk that the process will be sidetracked into discussions centered on whether the proxy is a valid measure.

In other cases, a desired endpoint may be envisioned by the PGIS system sponsors, or some potential paths to a decision might be placed off-limits. Realizing process constraints within the PGIS design is relatively simple, but this runs the risk of alienating participants by not allowing them to perform actions that they might want to be allowed to and reducing their perceived locus of control. Implementing such groundrules in the software might well produce a more (time) efficient decision process, but at the same time reduce participant satisfaction with the decision result.

A third consideration involves the complexity of the solution that will result. Many if not most problems subject to PGIS support are not amenable to simple solution. Creation of a complex solution is a complex task. Even if the software is capable of creating a complex solution, participants, especially those with limited computer skills, may not be capable of utilizing the computer to generate such a solution. Every attempt should be made to design PGIS interfaces such that the entire range of participants is able to participate fully. In addition, decision processes frequently involve negotiation among participants. Designing a PGIS that covers all possible participants and also facilitates negotiation may require significant resources.

For WaterGroup, the decision process the participants were involved in was in its early phases. Although substantial visualization data were available, quantitative data that would be useful in generating decision options and what-if analyses were limited. Proxy measures were created in lieu of actual data, and for the most part appeared acceptable to the participants. Since the participants were not being asked to “sign up” to solution options they created, sidetracked discussions were minimal.

In the decision process for which WaterGroup was used, participants were asked to determine timing for the management of groundwater wells within the Boise River basin (southwestern Idaho, USA). A constraint imposed in the software was that “all wells (except possibly the ones that pump very little water) must (at some point) be managed”. Several participants wanted to be able to create solution options in which only some (or none) of the wells would be subject to management. Even though the group would likely have come to the same conclusion themselves, the implementation in the software sidetracked participant discussions somewhat.

Creation of solution options with WaterGroup involved defining the years in which each well would be subjected to management during a 10-year phase-in period. Proxy measures of flow rate and impact upon surface water flow were used to choose wells to be subjected for a given year. The process of building solution options as implemented in the software was fairly complex, and those participants with limited computer skills might not have been fully able to contribute their ideas to the group. No explicit negotiation capabilities were included in WaterGroup.

Participants

The interface design for a PGIS should be customized as much as possible for the participants who will use the system. In some cases the target end users will be limited to a small group with similar computer skills and knowledge of the problem domain. But this should be considered the exception rather than the rule. In most cases participants will have widely varying computer expertise and extensive but disparate, focused knowledge of the problem domain. Design considerations for such groups are numerous.

In terms of computer skills, the interface should be designed to accommodate multiple levels of end users. For those with limited skills, this translates into point and click graphical interfaces and wizards for complex tasks. For more skilled users, the interface should include keyboard shortcuts and the ability to customize. And where resources are available, help functionality should be provided.

In some cases the complexity of tasks that must be performed by participants will exceed the capabilities of those less familiar with computers. An approach used to deal with this situation is to provide “helpers” or chauffeurs to help guide the participants through the difficult tasks. If this approach is taken, helper selection and training requirements must not be overlooked.

One of the tenets of participatory decision-making is that a diversity of participants is able to bring unique expertise and perspectives to the table. From the standpoint of the PGIS design, this means that the end-users of the system will, in aggregate, be capable of validating and verifying the data contained in the system. Unfortunately, participant access prior to final implementation is frequently not possible. As a result, inaccurate data or invalid assumptions made in the design may not surface in time for correction prior to the system being fielded. In order to avoid the potential of sidetracking decision-making discussions into validity debates, consideration should be given to as much system testing, utilizing the best available representative end-users, as possible.

For WaterGroup, no participant access was available during the development. From previous experiments undertaken with the participants, it was known that as a group their computer skills ranged from minimal to extensive. The prototype was designed so that individual participants would each have access to a computer and would be able to individually, or with the assistance of a helper, exercise the full functionality of the system. Resource constraints dictated an interface that provided no wizards, minimal shortcuts, and no help system. The resulting interface was judged too complex for beginners, and as a workaround individual access was replaced with small group (3 or 4 per machine) access to the prototype. This change in access in turn invalidated the voting functionality designed into the prototype, as each vote (one per machine) represented the consensus opinion of several participants.

WaterGroup helpers were recruited and received limited training. The helpers had extensive problem domain knowledge, but their exposure to the application interfaces utilized was limited. The helpers performed admirably, but there were cases where they were unable to adequately manipulate the interface, thus causing some frustration within the group.

Data validity concerns were not really an issue with WaterGroup. Even though the participants were extremely knowledgeable about their pieces of the problem, the data contained in the system were not actual but representative data. This was made clear to the participants early on, and they were able to accept the constraint and move on with the decision task. Note again however, that they were not being asked to “sign up” to any of decisions they made, and more concerns may have been raised if the decision consequences were higher.

RESOURCES

In the theoretical world of system development the designer starts with a blank page. But in the field the page frequently is pulled from the recycle box. The designer is faced with a multitude of resource constraints from the beginning of the project, and it is important to consider the constraints fully prior to embarking upon system design.

With unlimited resources the designer could custom code an entire system. With constrained resources it makes more sense to utilize existing functionality wherever possible. If the physical system is constrained to a particular platform, the scope of the search for functionality is narrowed and the designer can reasonably easily search for application software that can fit the bill. If the application software that is available to the designer is also constrained, the search can be narrowed even further.

Once existing software has been identified, a choice needs to be made about how best to provide for interprocess communication. As each application used will run in it’s own process space, there needs to be a means to transfer data among the applications. Otherwise, the final system might impose too high a process cost (e.g. save a file and open in another application; cut here, paste there; put it on a floppy and move it to another machine) upon the participants.

The designer needs to consider several other constraints. How much time and how many coding personnel are available? What programming languages do they know and how much custom code must be written? Will there be access to a physical test environment to validate and verify functionality? Are test subjects available that can adequately validate the system?

The designer also must consider training and implementation issues. Who will be trained and to what extent? What type of documentation will be required? Will the system be used in an ongoing basis, how much maintenance and upgrade resources are needed? Will there be a need for ongoing end-user support?

Always present are the considerations related to the GIS portion of the PGIS. Which data will be used? How much new data will need to be acquired? What preprocessing requirements are necessary? Who will be responsible for the spatial data? Will the data be updated or remain static?

For WaterGroup, the design was constrained to be use on a collection of 15 wireless laptops running Windows 2000. The machines had not been purchased at the time of development. Available software was limited to the ESRI suite of products and Microsoft Office. Interprocess communication was achieved using named pipes and the Windows 32 bit API. One person was available for coding and he was most familiar with Visual Basic. There would be no access to a test environment or test subjects. Helpers were recruited by the project sponsor and trained via 2-1 hour telephonic sessions with a pre-use 1 hour follow-up.

Substantial amounts of data were available for the GIS. However, much of the data were limited to imagery and visualization data, and less contained quantitative data that could be directly applied to the decision problem. Representative data were generated by the project sponsor a short time prior to fielding the system. Follow-on use of the system will likely employ additional and updated quantitative data with a different structure.

CONCLUSION

The listing of considerations included in this paper was by no means exhaustive, and in different problem domains additional considerations will certainly be present. But regardless of the nature of the problem upon which the PGIS designer is focused, careful up-front consideration of problem constraints is critical to the success of the development, and in turn, the end-user process.

BIBLIOGRAPHY

Jankowski, P., and T. Nyerges. 2001. *GIS for Group Decision Making*. Taylor & Francis, London