Chapter 7. Reflections, Critiques, and Future Directions

This thesis began with the concern that planning analysis is too expensive due to systemic problems with the information technologies in use today. No matter how small or straightforward the analysis is, it seems that much of the project time is spent gathering data and preparing it for use, and once a project is complete, its results can rarely be updated without incurring costs approaching those of the original analysis effort. And yet the sheer amount of data about the urban environment increases yearly as we install traffic counters on our roads, and air quality sensors on our rooftops. Making use of these data will not only require better information management techniques, but also better ways for experts to engage in collaborative analysis. However, the current state of technology makes it difficult to imagine that any but the largest projects and/or agencies will have the resources to marshal the expertise of various planning disciplines and their detailed data sources to analyze urban problems.

A look at the literature on geographic information sharing offers little help. That field tends to hold technology as fixed, which leads to solutions where organizational behavior issues are the focus. We do, however, make two important findings. One is that the most universal determinant of geographic information sharing success is the cost-benefit ratio of implementation. In most situations, the cost of sharing and collaboration is high, which creates a need for large benefits to all participating organizations. The second is that organizations often function as independent "trading partners" rather than one entity with different departments, or agencies. In this institutional context we attempt to implement one of two technologies; either we use some type of "enterprise" system, which centralizes information management too much for the comfort of most business partners, or a Web site based system, which is good at disseminating read-only data, but not at fostering collaboration.

What would happen if we instead adapt technology to fit the organizational behavior we observe in practice? This is the question we set out to answer in this thesis. It led to a research agenda based on Web services, a relatively new paradigm for developing multi-participant computing systems where the parties involved are "loosely coupled," meaning that their collaborative interactions do not require changes to the systems designed to achieve their core internal goals. The Web services paradigm was chosen because it is flexible enough to allow a discipline like planning support systems to build its own specialized tools, but defined enough so that the basic enabling software works for multiple industries, and is therefore a relatively cheap commodity.

The flexibility of the Web services architecture requires that our analytic tools must be re-developed within this new framework, and that was the focus of the second half of this work. Using the MassGIS buildout analysis as a lens through which we could look at the key information management challenges in PSS, we built up a Web services-based PSS framework called PAMML. PAMML consists of a vocabulary for describing the components, or building blocks, or information exchange and processing, along with a suite of Web services that can execute the requests articulated in the vocabulary. We showed how data sharing, stakeholder participation, collaboration, and iteration could be approached from this framework. We showed that the costs could be extremely low for small organizations with simple needs, but they could still interact with larger organizations with much more sophisticated systems.

Finally, we used the tools we created to re-develop the buildout analysis using PAMML and discussed some potential user interfaces that might drive a PAMML-based system. While this began to show what planning support systems might "look like" from a user's perspective, this thesis has been primarily about the development of the framework. This is unusual in that it is more common for technology research in urban planning to be about an implementation (software design and development) of an existing technology framework. This means that the technology presented here has been at a more abstract level, which creates a tension as the reader may have been expecting to see a concrete software application as the final product of this work. Fully developing a software implementation in this way would put the emphasis, and therefore the evaluation, of PAMML in the wrong place. This thesis concludes with some discussion of the difficulty in evaluating a work of this kind, some of the accomplishments made despite this difficulty, and some as yet unmentioned areas of urban planning that PAMML has a chance to significantly impact.

Critiquing the PAMML vocabulary

The vocabulary developed in this thesis has concentrated on spatial analysis and distributed processing. It is far from being a complete planning, or analysis, language. There are two main reasons for this. First, it would be impossible to cover the entire field of planning, or analysis in one research effort. That is rightly the work of a research community. Second, XML languages are designed to integrate multiple vocabularies, allowing people to specialize in certain areas. For example, many XML languages will require a way to describe someone's address. XML was designed so that everyone can reference and use the postal service's address definition, instead of inventing their own.

This is the strategy envisioned here for implementing a number of features that were not discussed. For example, the ability to construct database queries was not a feature of the PAMML framework describe here. This is not because database query is not a critical feature in PSS; in fact, the next level of sophistication in the treatment of data sharing would have been to discuss the need for data users to get incremental updates of large data sets, and synchronize their personal changes with the official repositories. These features were not discussed because they are so central to IT in general that the solution must come from the database query specialists, not the planning community. There is a great deal of work underway in this area,[1] but at this time, no standard query vocabulary has been widely adopted. This argument applies to many areas that would seem central to the development of a planning analysis vocabulary, such as metadata, where the Federal Geographic Data Committee (FGDC) leads efforts to standardize the way in which data developers describe a data set's provenance.

So, while this work has shown that PAMML addresses a wide range of physical planning issues, the language is not complete. But it can not be judged harshly on the basis of what features are missing, because this lack of completeness is by design. One could question how well PAMML is able to complete itself through integration with other vocabularies, but this is largely taken care of by the design of XML itself. XML

---

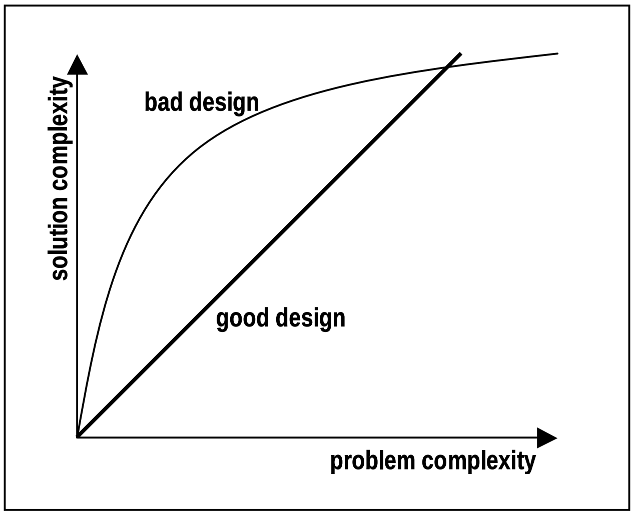
[1] Most notably XSQL (*http://xsql.sourceforge.net*) and XQuery (*http://www.w3.org/XML/Query*), a specification from the World Wide Web Consortium.

guarantees that vocabulary integration is possible, but it does not speak to the efficacy of this integration. But although that criterion is important, it is so subjective that it is difficult to discuss here.

Another way to think about evaluating the language is to look at the vocabulary from a semantic perspective. Is it too wordy? Can a term have multiple, confusing meanings? Are the meanings of terms easily recognizable to the community they serve? These issues have been addressed through careful adoption of well-vetted concepts within the spatial planning community. In terms of data modeling, we believe vector and raster and tabular data (along with some numeric types), are the key basic types to consider. For spatial operations, we believe that map algebra and set theory operations (union, intersect, etc.) provide the foundation upon which most spatial analysis is based. If one disagrees with this assessment, then PAMML will seem poorly conceived. However, these data types and operations seem to be well-accepted within our community, and therefore are not confusing, or verbose.

One risk, however, is that the language has been specified at too coarse a level of detail. For example, there are no requirements regarding the type of geometry within a vector data set. We said that this should be one of the seven types described in the OpenGIS Simple Features for SQL specification, but must it be one type, like polygon, or can a vector object in PAMML consist of a mix of different types, like polygons and lines. PAMML simply provides the syntax with which a user can specify either choice, which seems to be the strategy employed by most specification efforts, so we neither resolve, nor exacerbate any existing debates on this issue.

In our research and professional work on specifications and language definitions, the most important factor of success by far has been community adoption. If a language is able to garner widespread use, it serves its purpose. Because XML ensures that a language designer can not break the most important rules of information modeling, the success of an XML-based dialect has little to do with vocabulary syntax. The most



important factor seems to be the ability to solve the most basic problems simply and tersely. And another important factor is to make sure solutions scale smoothly in relation to the problem solved. In other words, if problem complexity and solution complexity were the axes of a graph, a good system design would map problems and solutions along a straight line.

If a large group of users can quickly solve their most basic problems, then they are likely to adopt the technology. This creates a large user base, and that is the key to any successful Web services framework. Unlike traditional desktop software, service-oriented frameworks have little value until a community of users exist that take advantage of the system. These users will fix any problems with syntax and meaning, and develop innovative new grammars on top of the original language that address new challenges that will arise, but the community must be developed in the first place, which leads us to our final critique—that these services are best tested in a multi-participant environment.

Exploring the nature of Web services as contracts

This research has primarily been about defining a technology framework rooted in Web services, however few concrete Web services were described. There are many reasons for this omission. First of all, the syntax of Web Services Description Language (WSDL) conveys little information to a human being. Second, every operation or object in the PAMML language could be expressed directly as a service, so from that perspective many services are described here. More important than these two considerations is the fact that the Web services are less important than the paradigm shift of moving to a service-oriented architecture (SOA) from a data-oriented one. This conceptual shift transfers the emphasis of information-intensive planning from data management to process management. In the buildout case, for example, once the planners set up data sharing agreements with local municipalities, a service-oriented framework like PAMML makes it as easy for the analysis software to always use the most up-to-date local information as it is to use zombie data off a hard drive.

This works in theory, but the theory has not been tested in practice. Concrete implementations of the services we describe in the abstract are important because that is where the “contract” between business partners is defined. In the first chapter we postulated that collaborative initiatives that use technologies that in some way capture the social or legal contractual relationship between parties may succeed more often. We expect to see this result due to the ability to better align expectations, in the form of "real" contracts, with performance, the information sharing system, thereby reducing the chance that either party will feel that they were not receiving the agreed-upon benefits. This postulate has not been tested here. It requires a level of empirical research beyond the scope of this work. PAMML reduces costs in many other ways, so this research stands on its own, but work on contracts has the potential to increase the strength of our argument.

When we speak of a service-oriented architecture, we are talking not only of the interaction between two parties, but of a whole network of content producers and consumers. A town assessing office may be a consumer when they acquire property information from private developers, but they are a producer when they then share their property database with a state agency. And if the state then sells this information (in an aggregated form, of course) to private firms, what is the town's relationship to the final end user? Do they have a contract with each other? And as a practical matter, what happens when someone cannot keep their server running? These are critical implementation questions whose answers will determine whether or not Web services, and service-oriented architectures in general, will work. Although as planners we must be concerned about this issue, there is little we can do. It is a concern for the entire information technology community. Many standards communities are working to define these multi-participant service relationships[2], yet no clear strategy has emerged. We can not even be sure that the effort in this area is warranted. WSDL already gives us the

[2] With names like Web services choreography language, Web services orchestration language, Web services flow language, Business process execution language, and Web services modeling language.

tools to define the relationships between a service consumer and producer. Why must multi-service systems require a different structure? This has yet to be proven, and may or may not become an important concern for planners.

Further implications for the planning profession

This thesis has concentrated on the value of the PAMML framework to improve the way in which planners engage in information management and analysis, two areas very central to the discipline of information technology. While some attention has been paid to the requirements of participatory decision making, for the most part the discussion has been restricted to those areas of planning practice traditionally associated with IT. However, if the PAMML framework were to become the backbone of planning support systems, this would create an opportunity for other practices to redefine the way they engage in spatial analysis. Two of these are explored here.

Democratizing urban design

One significant implication of this work is the potential to create, articulate and implement urban design patterns using PAMML. As Alexander says, "[a design pattern] describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" (Alexander, 1977). In the computer science domain, like architecture, design patterns are generic problem-solving models. They are the building blocks of analysis, like mathematical theorems.

Urban planning provides us with a rich history of design patterns that are beautifully articulated, but are difficult to apply outside the designer's original context. Looking back at Olmstead's emerald necklaces of the 19th century, that designer was able to implement his vision of how to integrate green space into a metropolis, but despite the popularity of the idea and the implementation, there have been few emerald necklaces designed after the Olmstead era. The same can be said of Kevin Lynch's "image-able city" (Lynch), Christopher Alexander's "pattern language" (Alexander 1977), or Alan Jacobs' "great streets" (Jacobs).

Urban design, as it is currently conceived, is inherently an expensive endeavor. Good urban design usually requires the attention of a team of professional planners over the course of weeks or months. People's time does not come cheaply, and therefore only the wealthiest communities, or the most important projects, receive the long-term attention of professional urban designers. The rest must make due with "commodity" design tools, resulting in communities whose aesthetic is driven mainly by the interactions of zoning regulations, road construction manuals, and the profit-maximizing tendencies of private developers.

We believe that an urban design language can be articulated using the basic spatial syntax found in PAMML. And if this is the case, then any design theory expressed in PAMML has the potential to be implemented computationally, instead of by professional designers, thereby greatly reducing the cost of design. One recent example of this principle at work is Bill Hillier's Space Syntax theory. This design theory has been described computationally and packaged into many different software programs (*http://www.spacesyntax.org/software/index.htm*), including ArcView (*http://www.casa.ucl.ac.uk/venue/space_syntax.html*), the most common GIS package in the world. It does not seem a coincidence that in little over a decade the technique has been used in localities all over the world, guided by people other than the original designer. Of course we do not expect the design process or the end product to be as rich as if it were performed by human designers, but the reach of one's ideas is exponentially broader.

Enabling community statistical systems

More than anything else, this work aims to change the paradigm of analysis from the current one-time major effort to produce a document to many small efforts that produce a continuous information flow. "Making plans for urban development is something you do constantly, not once" (Hopkins, 1999). The underlying assumptions that go into a plan, such as economic conditions and development activity constantly change, yet most plans are static. This is a necessary compromise based on the cost of marshalling the resources required to prepare useful plans. The framework suggested here allows plans to become dynamic tools—more like monitoring and early warning instruments than rule books. This may sound threatening to those who consider plans to be embodiments of a community's vision about their place, but Hopkins notes that plans are really the strategic implementation of visions, not the visions themselves. In this new type of plan, the community's vision is still present. It simply manifests itself in a different form, such as the point at which development triggers a moratorium or an infrastructure investment.

This paradigm implies that the goal of analysis and modeling should change from report generation to situation monitoring and performance measurement. The need to support this effort is evident in many places. The National Neighborhood Indicators Partnership is an effort to build “advanced information systems with integrated and recurrently updated information on neighborhood conditions in their cities (*http://www.urban.org/nnip/concept.html*).” This is the most explicit example of this change in focus, but the trend presents itself in many other places. The Heinz Center's ***Report on the State of the Nation's Ecosystems*** (2002) recommends that environmental quality be monitored and reported on in a consistent, constant way, in the manner of well-known federal economic indicators such as durable goods orders, housing production, consumer spending, etc. Indirectly related efforts include local government efforts to define a strategy for integrating the Internet into their mission. The National Civic League addresses this in the 8th revision of their Model City Charter. A joint project of the National Association of Counties and the National League of Cities seeks to support the ability of towns to automate government transactions over the Web through their “Totally Web Government” program.

Information technology developments that change the nature of planning tools also affect the planning process itself (Schuur, 1994). It is difficult to imagine that current planning support systems can facilitate the development of community statistical systems in a scalable, cost-effective way. Hopefully PAMML Web services can change the nature of planning tools and help elevate the discourse of urban planners among the voices competing to shape our society.

In this work we have used a standard framework for building multi-organization, distributed computing systems to redefine the information architecture upon which planning support systems are built. As an initial proof of its efficacy, we designed and prototyped a system that solved many of the issues found in spatial analysis and physical planning. This has been an important exercise, as physical planning is one of the core areas of PSS, but the real importance of PAMML will be seen in its ability to integrate the work of previously separate fields. For example, the computational expression of urban design, as discussed above, might be built using intermediate spatial analysis tools built by geographers, transportation planners, and environmental experts, but our current technology paradigms make it hard to imagine this type, or depth, of collaboration at any cost. Hopefully, the PAMML framework allows us to begin to envision a new era in planning support systems with fewer limits to our ability to collaboratively and continuously plan for the future.