

What is “Architecture”?

David D. Clark

V 4.0 of 28 November 2005

Introduction

The CISE directorate of the NSF has announced a new focus area as a part of their recent solicitation for research proposals. This focus area is called FIND, and it challenges the research community to conceive of the global networks we might desire in 10 or 15 years, and to propose research necessary to realize this network. The solicitation talks about “research that leads to architectural proposals”. It is worth a brief consideration of what “architecture” means in this context, and what kind of research this description might actually imply. The computer science discipline makes much use of the term “architecture”, but of course we borrowed the term; what “architecture” properly describes is the design of a building or other physical edifice, as a member of the AIA would be quick to point out. To understand what architecture means in CS, it helps to go back to that original use.

An architect, in the original sense, is not concerned with the creation of building materials—making glass with better thermal qualities or girders with higher yield strength. An architect is concerned with how the range of building materials can be put together in desirable ways to achieve a building suited to its purpose. It is the design process that assembles the parts into a useful, pleasing and cost-effective (some times) whole that is called “architecture”.

By analogy, in computer science we use the term architecture to describe how parts are put together to achieve some overall goal. The term can be used in lots of contexts. Chip designers talk about the architecture of a chip. “Chip architecture” does not refer to improvements in VLSI technology—making the line widths narrower, for example. That problem, like making stronger girders, is technology research: making better parts to facilitate new architectural conceptions. Chip architecture refers to the process of putting various functional components together in specific relationships to achieve a particular goal: a signal processor, an inexpensive “system on a chip”, a high-performance pipelined processor, and so on. One can talk about the architecture of almost any system: in the Internet context the Internet itself has an architecture, as does the Web and the email system. Designers of physical level communications systems will talk about architecture in terms of the physical design options. For example, a cable television distribution system has a *tree architecture*, while SONET systems often have a *ring architecture*.

What are the important points to note about architecture?

First, architecture only makes sense in the context of a set of objectives. One cannot just “design a building”; one must know whether it is an apartment building, a museum or a prison. This fact is equally true of computer systems: a chip, a network or an application

is designed for a purpose. A system designed to “do anything” is nonsense. In this respect, one signal of a realistic architecture is what it admits that it *cannot* do, as well as what it does. At the same time, most CS designers strive for generality—it has been a very powerful and successful approach to design. The Internet strove (successfully, I would claim) for a high degree of generality, but there are limits: it was not conceived as a hard real time system, or for circumstances with round trip latencies measured in hours.

Second, architecture is about how the parts fit and work together. In computer terms, this means that architecture is about modularity and decomposition, interfaces and dependencies, and exploiting reusable parts, as well as fundamental design principles and approaches.

Third, built objects often come to be defined by their architecture. The architect helps the client refine the requirements, proposes approaches, expresses design preferences, adds form and context to function, and in that process comes to create the essence of the object. We all might share some idea of “place of worship”, but it is the architect who gives us the concept of “cathedral”. The essence of “cathedral” is architecture, and so it is with many computer elements. When we describe “the Internet”, and focus on what it is that defines it—what separates it from the telephone network or a cable television distribution network, we are in fact describing its architecture.

Fourth, architecture (like most human endeavors) benefits from what has already been tried and learned. It is a discipline to be studied and learned. Architects study existing buildings, and CS students study existing systems. As teachers, it is our job not just to show these systems, but to discuss the *why* and *oops* they illustrate. In (building) architecture school, students are taught about design patterns. These are reusable patterns or principles that lead to successful designs with less need to create from scratch. Similarly, different aspects of CS have their own design patterns: chip designers have methods for solving specific problems such as clock distribution and power reduction, network protocol designers are taught about layering and abstraction, and more specific patterns such as the end to end arguments.

So we should think of architecture as first a process, second a conception, and finally as a discipline.

The scope of the architectural challenge

The upcoming NSF initiative is titled FIND, Future Internet Design. From the name, one might erroneously conclude that the scope of the challenge is limited to a narrow conception of the Internet—a new IP layer. In fact, as the solicitation makes clear, the scope of the solicitation is quite broad. It uses as examples concepts that are today thought of as “lower” than IP, and concepts such as location management, identity management and information management, which might today be “above” IP. These examples beg the question of what the limits to FIND actually are.

If the Internet is defined narrowly by the IP layer, it is conceptualized broadly by a set of features, capabilities and services that are defined in common across the net, and are

generally useful, even if not mandatory. The DNS is not part of the specification of IP, but most would consider it to be a part of the Internet. TCP is not mandatory (one can choose to use alternatives), but it is clearly part of the architecture. Individual applications have an application-specific architecture in their own right, as do technologies such as Ethernet. FIND is less concerned with specific network-layer technologies or individual applications. The Future Internet, like the Internet of today, will sit on top of multiple heterogeneous technologies and support multiple applications. Within that space sits the focus of FIND: the architecture of features, capabilities and services that should be globally defined in common across the network, and that are generally useful across multiple technologies, or in support of multiple applications. An approach to traffic engineering might be generally useful across several technologies, and so could be within scope. Similarly, an architecture for location management, identity management or information management might well serve as a cross-application, general service. If so, work on these ideas, if proposed in a way that implies “thinking architecturally”, would be in scope.

Of course, part of the FIND process will include discussion and debate about what features will be generally useful in a Future Internet. So a proposal that identifies a candidate requirement, proposes some new mechanism as worthy of architectural specification, and proposes an approach to designing this mechanism, is well within scope. The argument about what should be architected is as much a part of the project as the architectural approach itself.

Finally, it is important not to focus on layer definitions to decide what is in or out of scope. While traffic engineering may be “below” IP in the current architecture, and identity management might be “above”, there is no reason to think this is true in the future. In fact, in the future we might decide to depend less on layering as a design pattern, and use some other approach to architecture. Even the concept of layers is up for debate in FIND.

Perhaps the project should not be called Future Internet, since the use of “Internet” might cause confusion. Perhaps instead some new acronym should be invented: perhaps GNEMGUCS, or “Global network mechanisms generally useful as core services”. “Internet” is perhaps easier to say and more familiar. To mirror the old quote about Fortran, we don’t know what the Internet will look like in ten years, but for marketing reasons it will probably be called the Internet.

Thinking architecturally

What sorts of proposals would make sense in the context of FIND? Few if any proposers will be in a position to put forward a complete architectural proposal. Most of us don't "design a new network architecture" on a daily basis. We work on parts, mechanisms, and approaches –architectural building blocks. These might be such things as schemes for addressing and forwarding, routing, congestion control, traffic filtering, and so on. And FIND solicits that sort of work. The difference is the context in which the work is positioned.

Most work on mechanism and components is positioned in a context--a framework of expectations about what the technology is good for, why it is worth developing, and what the other components are into which it must fit. For most of us today, the context is the existing Internet. There are good reasons for this:

- The current Internet is where the action is, and this context improves the chance that individual research is relevant.
- Starting with the current Internet gives us a very concrete context to work with, and this is useful.

On the other hand:

- Using the current Internet as a context can be constraining, and point us at the present more than the future.

The FIND solicitation does not preclude research on components and technology. It invites that sort of work. But FIND challenges us to do our work on components and technology in a different context. Instead of the context of the current Internet, the work must be in the context of a Future Internet that does not yet exist. Since the Future Internet does not exist yet, this context is not at all concrete, but very abstract. It exists, to the extent that it exists at all, at the level of goals and requirements (about which we may not all agree), design principles and patterns (about which we may not all agree), and other parallel research projects (that are going on at the same time.) This challenge has some implications. Specifically, a researcher who is successful here must be able to function at two levels at once, and link them together.

- They must be able to think in depth about their particular topic of interest--the specific focus of their research.
- They must be able to think about this context--this Future Internet--in its very abstract and evolving form.
- They must be able to link the two. They must be able to explain why their particular topic of interest is important to this Future Internet Architecture: how this research will inform the debate and shape this architecture. The important ideas are the one that FIND should fund. (There are many little ideas that might well *be a part of* a future architecture, but would not be important to shaping its creation.)

So a lot of current networking research can be fit into this new context, but this is not just a process of cosmetic adjustment. It is a deeper adjustment. Because this context is so abstract, it will both invite and force us to think more about fundamentals and less about how to conform to the concrete existing Internet. And this is the challenge of FIND.