CityWare:
Information Technology Planning And Urban Planning

Marc Demarest
marc@noumenal.com

October 1998

1. Introduction: Blindness and Insight

Because I am an architect, I am interested in the form of things, and particularly in the form of the physical environment, natural and man-made.

Joseph R. Passonneau, "A Planning Inventory for the Metropolis," in *Planning for a Nation of Cities* (1966).

The philosopher Paul de Man once observed that the strategies we employ to organize our thinking are at best mixed blessings. For every *insight* such strategies yield into the nature of the problem at hand, de Man observed, these strategies produce *blindnesses*: the inability to have other possibly more important insights, the inability to uncover some fundamental flaw or misconception, the inability to think past our own constructions.

The crucial problem, de Man pointed out, was that this duality -- the simultaneity of blindness and insight -- is a fundamental condition of thinking itself, since we are compelled by language itself to employ strategies -- metaphors, similes, figures of speech, models, paradigms -- when thinking about anything, from computing to thought itself.

What I want to suggest in this essay is that we have available to us today a powerful strategy for thinking through the nature of enterprise computing: the metaphor of **the firm or enterprise as a city**, designed expressly to collect, manufacture and deploy information through an immediate and intuitive connection between information and the workers who both comprise and make good on the competitive advantage of the enterprise.

This use of the metaphor of the firm as a city is a deliberate attempt to do three things:

- root out all the insights that a well-developed history of urbanism and urban planning can provide us about the design, construction and maintenance of cities

- escape the unenlightening, often confusing images associated with computer technology and the design practices associated with that technology: in short, to get clear of the mess we have made of our own discipline

- begin to expose the qualitative or ethical dimension of the discipline we practice: what “good” IT architectures might be.

I believe that, by employing this metaphor, we can begin to dig our way out of two fundamental problems that plague IT architecture and IT architects today:

- the term *architecture*, and the terms and concepts that group themselves under that word, have become so hopelessly muddled (and perhaps polluted) that we can no longer communicate with one another about the materials, the heuristics, or the ethics of the trade we practice

- because we are not clear about what we do, how we do it, or why doing what we do offers material value to the firms and organizations we work for, we cannot answer the question that is the *sine qua non* of late twentieth century commerce*: what value do we add?*

As we do this, we need to keep de Man's warning within view. For every insight the metaphor of the city provides us with respect to enterprise computing, it masks a difficulty, a problem, a flaw. Our success, in the long term, is dependent on our ability to know when the metaphor is no longer stable; when it has ceased to enlighten and when it begins to mislead us.

marc@noumenal.com

2. The House That IT Built

The reality of the building does not consist in the four walls and roof but in the space within to be lived in.

Lao Tsu, *Tao Te Ching*

Bill Inmon's classic text, *Building The Data Warehouse*, begins with this observation:

> The world of information systems is an "immature" world. One has to be careful using that word in public because it normally has a negative connotation. But from a historical perspective, it is true....the information processing profession is historically immature because it has existed only since the early 1960s. One of the manifestations of the information processing profession's youth is the insistence on dwelling on detail. There is a notion that if we get the details right, the end result will somehow take care of itself and we will achieve success. It's like saying that if we know how to lay concrete, how to drill, and how to install nuts and bolts, we don't have to worry about the shape or use of the bridge we are building. Such an attitude would drive a more mature civil engineer crazy.[1]

We find a similar emphasis on the relationship between information technologists and the building trades in the introduction to Steven Spewak's book on enterprise IT architectural planning. Discussing the Bay Area home of Sara Winchester, widow of the Winchester rifle magnate, Spewak remarks that:

> Tours are now given through the Winchester House. The grounds are beautiful and the custom-made stained glass, porcelain fixtures, and woodwork are remarkable. However, the highlights of the tour are such odd features as stairways that rise into ceilings, doors and windows blocked by walls, more passageways and halls than rooms, a three-story chimney that falls short of the roof, and many rooms serving the same purpose. The information systems portfolio of most companies resembles the Winchester House in many ways....There was no overall set of blueprints that showed what Mrs. Winchester wanted[2] her house to be....The only way to break out of the mode of continuous custom building and replacing of systems that result in such costly odd features is to create enterprise-wide architectures and plans for implementing them.[3]

To have the same essential metaphor -- the information technology trade is like the building trades -- seems natural to those of us who work in information technology. We commonly employ bits and pieces of building metaphors in our specialist vocabularies, and perhaps more importantly, use a word -- in many different contexts, with many different and perhaps incompatible meanings -- that is in both origin and historically borrowed from the building trades: *architecture*.

---

[1] W.H. Inmon. *Building The Data Warehouse*. New York: Wiley-QED, 1993. p.1 That we routinely use a specific kind of building to describe part of the firm's decision support systems (DSS) problem set ought to have been, for us, a clue that we were working with the IT analog of an interurban system – the data distribution network, designed to approximate its real-world goods distribution cousin. But it did not.

[2] What Mrs. Winchester wanted was a house sufficiently byzantine to prevent the ghosts of the people killed by her husband's product (firearms) from finding her within the house. One wonders if Spewak is suggesting a similar psychotic impetus for modern commercial IT architectures.

[3] Steven H. Spewak with Steven C, Hill. *Enterprise Architecture Planning*. New York: Wiley-QED, 1992. p. xix.

marc@noumenal.com

That the information technology profession has, and continues to, borrow *concepts and practices* from other older disciplines is not surprising. In fact, this is precisely how younger disciplines evolve: by leveraging the accumulated experience and theoretical understanding of older disciplines whose models and practices can be translated into the new problem areas of the younger discipline. Although the information technology profession sometimes behaves as though its problem set is unique and without historical predecessors, we can see, in the use of concepts, models and metaphors drawn from the building trades, the information technologist's subtle admission that we have, after all, been here before.

It is also not surprising that we should find, under the high-technology specialist vocabularies of the information technology profession, the older and more familiar metaphors of the building trades, particularly -- as Spewak suggests -- **the metaphor of the house**. First of all, our language is essentially metaphoric -- we explain things by means of analogic comparison, and the more complex the thing, the more immediately we resort to explanations *in terms of* something else. Gareth Morgan, for example, has demonstrated the extent to which our understanding of organizations, and in particular the commercial firm, and our general rules and precepts for managing firms and other groups of people, are in fact drawn from the reigning metaphors of the day: the organization-as-machine, the organization-as-organism, the organization-as-collective consciousness, and so forth.[4] Today's emphasis on the organization as a complex political phenomenon is no doubt the result of our collective preoccupation, in other spheres, with the breakdown of conventional modes of political participation.

The information technology profession's subtle dependence on the images, models, concepts and practices of the building trade are good signs, signs of an immature discipline struggling to develop itself by patterning itself on more stable kinds of knowledge. We should take heart in this dependency. But, as the quotations from Inmon and Spewak demonstrate, something is changing in the metaphoric space in which information technologists think and plan and dream. The metaphor of the house -- and the idea of the information technologist as a builder of houses – is, or rather ought to be, giving way to a new metaphor that is as yet undeveloped: *the metaphor of the enterprise as a city, and the IT organization as city planners, city builders.*

[4] Gareth Morgan. *Images Of Organization*. Newbury Park, California: Sage, 1986.

3. The Architecture Of The House That IT Built

Looking at cities can give a special pleasure, however commonplace the sight may be. Like a piece of architecture, a city is a construction in space, but one of vast scale, a thing perceived only in the course of long spans of time.

Kevin Lynch, The Image of the City (1960).

A recent popular text on downsizing information systems relegated its discussion of architecture -- the central conceptual framework for both IT and the building trades -- to its eleventh chapter, some 170 pages into its discussion of downsizing issues. The text begins:

> A client/server computing architecture is a computing architecture in which applications are partitioned between clients and servers.[5]

We would have to look pretty hard to find a better example of a vacuous tautology, and yet we can hardly blame the authors. If terms like *architecture* and *client/server computing* once meant something precise and broadly agreed-upon, the terms today are the playthings of IT marketing organizations, who spin their definition of the terms to match the capabilities of their firms' products and services, evacuating the terms of all referential meaning in the process.

And sure enough, this text demonstrates the problem, at least with respect to the term *architecture*, conclusively. Under the heading vendor architectures,[6] we are introduced to:

- the by-now legendary block diagram of the Distributed Computing Environment (DCE), with its neat boxes labeled "distributed file services," "PC integration," "security," "management" and "other fundamental services (future)"
- Hewlett-Packard's New Wave Computing, described as "object management technology and object-oriented networked services...[using a] distributed object management facility (DOMF)..."
- IBM's Open Distributed Computing Structure (ODCS) which we are told is implemented "through various products"
- NCR Corporation's Open Cooperative Computing Environment (OCCA), centered around "NCR Cooperation for MS-DOS servers [which] is an object-oriented application-integration environment consisting of more than 50 software modules that implement OCCA in multivendor systems. Rhapsody, a similar product from

---

[5] Steven Guengrich and George Schussel. *Rightsizing Information Systems: Proven Methods For Reengineering Complex Systems* (2 ed.). Indianapolis, Indiana: SAMS Publishing, 1994. p. 170.

[6] Ibid., pp. 179-182 inclusive.

marc@noumenal.com

recent NCR partner AT&T, focuses on workgroup features, combining an enterprise-wide orientation with workgroup features. In addition, NCR Desktop, the interface for NCR Cooperation applications, provids the object management facility and task-automation technology of HP NWC and NCR's remote method."

- Microsoft's Open Database Connectivity (ODBC), which we are told is "Microsoft's architecture for data access across a heterogeneous environment."

- Apple's AOCE (no expansion of the acronym is given to us), which we are told is "a framework for creation of collaborative or workgroup applications and also a platform that uses hardware and software resources to their best advantage"

- the Open Environment Corporation's three-tiered application design model.

Under the heading *architecture*, then, we are invited to consider as essentially comparable:

- a distributed operating environment (DCE) only parts of which are commercially available

- a defunct graphical user interface technology (New Wave) whose last viable commercial incarnation was as a replacement for the Microsoft Windows Program Manager (Dashboard)

- a collection of unnamed products stitched together with marketing collateral (ODCS)

- two highly-integrated proprietary (and commercially defunct) workflow products (Cooperation and Rhapsody)

- a programmatic interface specification for (a) decision support and small-scale online transaction processing (OBDC) and (b) client/server middleware whose future has been called into question by its vendor

- a vendor's desktop object model & compound document architecture (AOCE)

- a theory about client/server application partitioning backed up with DCE- and RAD-based development tools (OEC).

Is it really the case that each of these qualifies as a specific instance of the class of things known as *architecture*? If this were so, then the word *architecture* would clearly have no meaning, and therefore no use for us, and the IT profession would abandon it. But that is not the case. We know that:

- all of the products and technologies listed above have something to do with *architecture,* and may in any given case be both *the results of architecture* and *the building materials for architecture*

- other, more disciplined writers, thinkers and practitioners -- one thinks of, say, John Zachmann -- use the word *architecture* in far more precise and useful ways.

marc@noumenal.com

The point I want to draw from this is not -- **emphatically not** -- that the word *architecture* and the concepts that underpin it have no more practical value for the IT profession. What I want to suggest is that:

- the rank confusion of the example above is typical of the widely divergent and imprecise uses of the word *architecture* in IT theory and practice today

- the confusion is a sign of decaying models and changing metaphors

- we need to understand how the history of the concept architecture in the IT profession could create the confusion in the example, in order to understand how we take our disciplinary understanding of IT architecture forward.

marc@noumenal.com

4. A History Of Architecture As An IT Concept

When we deal with cities we are dealing with life at its most complex and intense. Because this is so, there is a basic aesthetic limitation on what can be done with cities: *A city cannot be a work of art*.

Jane Jacobs, *The Death and Life of Great American Cities* (1961).

Broadly speaking, the history of the term *architecture*, as a central concept in the information technology discipline, can be divided into three successive periods, during which architecture was focused on fundamentally different objectives:

- architecture as programming discipline
- architecture as technical specification
- architecture as design (of something).

Phase One: Architecture As Programming Discipline

When IBM and DEC[7] introduced the term *architecture* into the IT discipline in the 1970s, they both attached very precise meaning and implication to their use of the term. IBM defined architecture as:

> ...definitions of interfaces, formats and protocols to be used between the components. These should be of sufficient clarity and robustness to permit the asynchronous development and ongoing re-implementation of the components. Here, we introduce two critical corollaries of the architectural approach: asynchronous development, which permits the execution of complex projects and the ability to re-implement modules without affecting the user...[8]

while Digital suggested that:

> At Digital, to create an architecture is to envision all the components of a solution, then very carefully define the relationships, or interfaces, between each component. If an architecture is designed well, the interfaces remain constant and provide the stability to improve any part without affecting the whole.[9]

Here we find the oldest and in some senses the most fundamental notion of architecture. *Architecture* is a set of programmatic interfaces that are stable relative to the implementation of a particular technology. This architecture, we are not surprised to find, is **an architecture for the IT programmer**: for the builder of applications. Good architectures allow builders to work independently of one another, to develop

---

[7] Historically speaking, it is not clear that IBM was the first IT firm to use the word architecture. But certainly pride-of-place belongs to them because they in fact installed the term in the data centers of the world and brought it into wide usage.

[8] IBM Corporation, "Defining Architecture"

[9] Digital Equipment Corporation, "Guide to Information Systems"

marc@noumenal.com

components and applications that last, or are at least portable, as the technology underneath them changes.[10] The programmatic interface guaranteed both the integration of independently-developed components and the longevity of those components.

This definition of architecture was well suited to proprietary technology paradigms, in which one vendor (or at best a small group of vendors) provides the vast majority of the technology used by the IT organization. IBM and DEC clearly intended to shift technologies underneath applications developed by their customers. Both firms needed some way to both:

- encourage their customers to develop many and vast applications (in the process binding themselves to IBM and DEC technology because switching costs would be exhorbinant)
- prevent the ugly possibility that, as the technology underlying customer applications was changed out, IT programmers would be caught in a vicious and unproductive cycle, constantly rewriting the same application over and over again.[11]

What this original conception of architecture produced was: **components** that could be combined into applications to be deployed on top of a set of technologies the capability and change rate of which was controlled by the same vendor that published the architecture.

Phase Two: Architecture As Technical Specification

In the 1970s and 1980s, the strict definition of architecture as “programmatic interface specifications for the IT application programmer” began to expand, in several directions at once. Multiple factors drove this expansion:

- the discovery that complex systems, implemented at great cost over fairly long periods of time, were not likely to meet the expectations of “the business,” which operated with a cycle time different and shorter than that of the IT organization, and which operated with a higher level of implicit expectation than IT designers were comfortable with. This is the now-famous “requirements analysis” problem, which suggested that more than a stable programmatic interface was required to produce useful applications.
- the expansion of computing resources, geographically as well as logically, and the at least partial integration of computers and telecommunications, which required design, planning and management of IT across physical space

[10] One would have to be absolutely blind not to see that, in this incarnation, architecture was fundamentally for the convenience of the IT vendor, not the IT organization.

[11] An uncharitable interpretation might add “and therefore fail to purchase more, newer technology for other, new applications.”

marc@noumenal.com

- the widespread commercial adoption of database management systems, which replaced flat file storage mechanisms and carried with them significant design and implementation considerations

- the arrival of the first generation of open systems technology, which shifted the "integration burden" from the (usually single source) IT vendor to the (now multi-vendor) IT organization, and led to the emergence of a model of the IT function as integrator of (supposedly) work-alike components.

- the advent of robust commercially-available software packages as inexpensive alternatives to the in-house development of "command-and-control" OLTP applications like financials and manufacturing applications, which forced the IT organization even further out of its traditional role as in-house software development operation

These forces created a new notion of architecture, seldom explicitly defined, but nonetheless in widespread use throughout the 1980s and perhaps best exemplified by the work done by ISO and CCITT on the OSI reference model and its associated specifications. This new notion of architecture, in broad terms, focused IT architects on what were seen as two critical elements: data and technology components.[12] If the IT organization was now responsible for:

- the development **and** the integration and deployment of applications

- built on database technology and the notion of managed data stores

- accessed and used across the firm

- by means of increasingly complex sets of technologies from multiple vendors

- integrated into a functional whole

then, clearly, the focus of architecture would have to change to encompass and say something about the new focus areas.

In the area of data, the problem of architecture came to be seen as one of physical and logical design of database schema for particular applications. Increasingly robust modeling techniques, culminating in entity-relationship diagramming and (lately) object-oriented modeling methods produced, by the end of the 1980s, a clear operating notion among IT professionals that data had its own unique (and central) architecture: data architecture was a specialist discipline unto itself.

In the area of technology, the problem of architecture came to be seen as one of physical and logical topologies for the "wide area computing environment," and the central tools of the architect are those now-infamous boxes- and-wires diagrams, showing the location of each system, the applications and data each system hosted,

[12] Sometimes these two elements were expanded to three: data, *applications* and technology.

marc@noumenal.com

and the media and protocols connecting each system to other systems and to user communities.

And what architecture -- of both the data and technical strains -- produced was: **systems**, complexes of data, application logic and technology that performed particular functions for the business.

This period, retrospectively, was a period in which **architecture served the IT department**, and *architecture* during this period was exemplified by:

- Laundry lists of technologies that were “in” or “out”
- “Boxes-and-wires” diagrams, which were the *sine qua non* of technology architectures
- In-house “standardization” efforts, in which one component or other of the technology mix was isolated, and a single (or a few) vendor for that component chosen
- Huge “specifications” and “guidelines”, based almost exclusively on arguments about technical merits of various component alternatives, defined the technology mix for the firm
- Methodologies used like cookbooks, and argued over like religious texts.

In the span of less than 15 years, then, architecture had moved from a narrowly defined role -- programming discipline -- to a far broader one. This new role, if not poorly defined, was certainly difficult, complex and contentious.

Phase Three: Architecture As Design

By the end of the 1980s, several new forces began to act on the operating notions of architecture in the IT profession.

- First of all, the first generation of open systems -- promising work-alike best-total-cost plug-and-play options for each technology component in a firm’s technology mix -- failed to deliver on its promises.
    - Common base technologies did not work alike. UNIX, for example, did not consolidate; in fact UNIX variants of increasing variance, abounded and did not interoperate well or at all.
    - *De jure* standards efforts failed. The OSI initiatives failed utterly outside certain markets effectively controlled by governmental organizations, which were themselves increasingly uncomfortable with OSI as an open systems paradigm.

marc@noumenal.com

- Irresponsible, radical downsizing initiatives, focused on unplugging and removing the mainframe from the data center and replacing the mainframe with commodity technologies, failed in droves, wounding some firms and damaging others irreparably.

- The cost of integrating dozens (or hundreds) of open systems components into stable configurations proved not less expensive than pre-integrated proprietary systems, but significantly (possible an order of magnitude) more expensive than such proprietary systems.

- Computing resources did not, as promised, become free. Despite the marketing clatter about infinite essentially free CPU cycles, memory and disk space, not only did the demands placed on CPUs, memory and disk by software rise faster than those technologies' capabilities.[13]

- The network became less a piece of telecommunications infrastructure than the vital fibres that knit the communities that comprised the firm together. Absent the claims of vendors that "the network is the computer" and the spurious promise of "fully distributed computing" the plain fact is that, during the 1980s, the LAN took over the world.

- The Internet came of age as a commercial computing utility.

- Semi-structured and multimedia data became mission-critical

- The typical firm became porous: open to IT-based interconnections upstream (with suppliers), sidestream (with business partners) and downstream (with channels and customers)

- The PC transformed itself from a desktop annoyance into the center of enterprise IT planning.

- Franchise vendors – vendors who define standards and make or break markets in the area surrouding their core technologies -- began to dominate key technology areas.

- IT, once seen as an efficiency mechanism of marginal interest to business strategy, was recast by management science as the basis -- perhaps the only basis -- for a firm's competitive advantage.

---

[13] For compelling experiential proof of this, I invite anyone who has owned more than 2 generations of desktop technology to consider two data points: the sizes of hard disk seen as 'bare minimum' across those generations and the relative size -- in bytes of binary storage and memory requirements -- of the last two generations of word processors used. In 1987, I had a CTOS workstation with a 10 MB hard drive that seemed unfillable; the PC on which I am writing this has a 2 GB drive that is 90% utilized. In 1987, I wrote papers on a dual-floppy PC using a version of WordStar that fit in 512K and on a single 5.35-inch floppy. The version of Microsoft Word I am using requires more than 15MB of binary storage space. Ultimately, I would argue that Moore's Law and other "laws" that suggest that computing resources must become free because they are increasing so fast fail to take into account what we might call **Gates' Law**: (a) the fact that higher level resources immediately balloon to consume all new capacity and (b) the amount of data we generate and store is increasing at a rate faster than that of CPU power, disk space and memory capacity combined.

marc@noumenal.com

- Increasingly robust and reliable data sets indicated that the requirements definition problem tackled in the 1980s had not only not been solved, but was worse than ever. IT, as the phrase went, "was not linked to the business."

All of these forces combined to call into question the focus of earlier definitions of architecture: the emphasis on the **system** as the product of architecture. Instead, the new argument ran, the emphasis of architecture ought to be on:

- planning and linkage, ensuring that technology choices, data and applications returned quantifiable business value to the firm

- leveraging an IT marketplace far more inventive and responsive than in-house development organizations

- the enterprise: architecting IT solutions that integrated systems and resources and responded to increasingly rapid and radical changes in business needs.

The single most important harbinger of this new notion -- that, fundamentally, **architecture is for the business**, and provides an enterprise, not a system, view of data and technology -- was John Zachman's framework, which first appeared in 1987, but was not, I would argue, in broad practical use until the early 1990s. Zachman saw quite rightly that there were multiple views (he defined six) of multiple architectural elements (he defined three, essentially data, application and (network) technology), and attempted to unify these views of these elements into a tops-down model of architecture driven by business requirements.

Retrospective: From Materiel to Systems To… What?

Seen retrospectively, the evolution of the word *architecture* in the IT profession went through three phases from its introduction until roughly the present day.

- In the 1970s, architecture described a discipline for IT programmers (and IT vendors) that produced (supposedly) reliable, rehostable components (materiel). Architecture dealt with components used to build things.

- In the 1980s, architecture described a discipline for IT designers (and IT vendors) that produced (supposedly) well integrated, performant systems. Architecture dealt with built things.

- In the 1990s, architecture has come to suggest – or perhaps ought to suggest – a discipline for IT planners (and IT vendors) that produces robust, resilience constellations of systems and components in support of the firm's business objectives. Architecture, today, deals with the relationships and interconnections among sets of built and bought things.

Seen this way, I would suggest that all we have to do is look back on the source of the word architecture – the trades of the built environment – to see that we have, in the IT profession, rediscovered the transition from building materiel to building design to urban

marc@noumenal.com

planning: to the design and construction, over long periods of time, of complex urban environments designed specifically to support an undefinable and complex set of human activities.

IT architecture is becoming, in other words, the analog of urban planning.

But then, the enterprise has always been like a city.

marc@noumenal.com

5. IT Architecture And Urban Planning

The town is a tool.

Le Courbusier, *The City of Tomorrow and Its Planning* (1929)

The idea that IT architecture is like urban planning begins with a simple observation: enterprises, regardless of the businesses in which they are engaged, function like cities: they exist as a complex of locations in space, populated by human beings, engaged in work of various sorts.

People occupy the enterprises in which they are engaged. To the extent that a company succeeds in creating a sustaining environment, its employees are city dwellers: they bind themselves to their enterprise, working and playing largely within its environs. To the extent that a company remains the source of a paycheck, its employees are commuters: they arrive, work, and depart to return to their lives, which are lived elsewhere. Some companies have intuitively or explicitly recognized themselves as cities: within the walls of their physical plants, one can find athletic centers, hair dressers, restaurants, and other essential services designed to make the experience of work more complete, more focused -- more like the experience of a city.

Both cities and business enterprises provide infrastructure for the activities they support. Both provide highways and byways, scenic routes and, unfortunately, dead ends, but where the city moves people, the enterprise moves product, and information, which are **increasingly the same thing**. In fact, it is the use of information technology as the infrastructure of the modern enterprise that makes the enterprise most like a city. We commonly speak of hardware and software *architectures*: of *building* systems and networks, of information highways and superhighways. And why not? What, after all, is a corporate network if not a private turnpike? What is the intelligent desktop if not the enterprise's answer to the worker's demand for a home? What is a database management system if not a restaurant, a vehicle registration center, a hall of records or a grocery store?

The History Of IT In The Enterprise Is Like The History Of The Modern City

The modern city arose out of the problems caused by the application of high technology -- primarily mechanical power -- to older, human-scale industries in the early years of the nineteenth century, first in the United Kingdom and later throughout Europe and America. The general evolutionary pattern was one of:

1. **concentration**: the placement of high technology in factories refocused the lives of workers, concentrating what had before been piece-work cottage crafts

marc@noumenal.com

2. **explosive growth**: around the factory grew the town. Around the town grew the city. Beside the town ran the canal, and then the railway. Populations soared. The industry that gave birth to the city became utterly dependent on it, and on other cities, both for sources of labor and for markets in which to place product.

3. **migration**: the city, as its infrastructure became increasingly dysfunctional, was evacuated by those most able to leave, who set up anti-cities outside the city limits: suburbs

4. **intervention**: at this stage in the evolution of the modern city, the first urban planners intervened in the evolution of the city, ameliorating the worst excesses of the city: slums were razed, sewage systems were installed, police forces were chartered and put on the streets, mass transportation was made available. The city was stabilized, and began to grow again, overtaking its suburbs, making them part of the urban space.

The implementation of information technology in the modern enterprise has followed a surprisingly similar trajectory. The 1950s and 1960s constitute the era of concentration, as human-scale occupations -- book-keeping, order processing and the like -- were automated, and the core of the enterprise became dependent on information technology. During the 1970s and early 1980s, the growth of IT within the typical enterprise was indeed explosive; in not a few cases, a worker's environment was entirely automated by the beginning of the 1980s.

During the mid-1980s, the knowledge worker deserted the mainframe city for the suburban anti-city of the personal computer, which succeeded precisely because it was unlike the city/mainframe: familiar, friendly, and, most importantly, under the knowledge worker's control. The remainder of the 1980s constitute a largely ineffective attempt, on the part of IS, to stop the most important segment of the enterprise population -- the knowledge worker -- from migrating to the desktop for computing services (just as urban planners spent equally unfruitful time in the 1960s trying to stem urban flight).

The 1990s represent, potentially, the period of intervention. Clearly, the popularity of client/server computing structures can be linked to a desire, on the part of IS, to reintegrate the suburban desktop into the urban space dominated by the data center, and a related desire, on the part of the knowledge worker, to have the data center behave more like the neat and manageable suburbs she chooses to inhabit.

The Future Of The Enterprise Is Like The Future Of The City

We are commonly asked to entertain descriptions of the current world market as a post-industrial or post-capitalist market: one in which, as Manual Castells writes, is characterized by

> the emergence of information processing as the core, fundamental activity conditioning the effectiveness and productivity of all processes of production, distribution, consumption and management. [14]

[14] Manual Castells. *The Informational City: Information Technology, Economic Restructuring and the Urban-Regional Process*. London: Basil Blackwell, 1989. P. 12.

and is fundamentally a market in which companies increasingly discover that information is

- the raw material of the enterprise
- the 'means of production': the tools used to direct the enterprise
- the product of the enterprise.

The fundamental contradiction in this post-industrial or post-capitalist market is, as Castells points out, that:

> on the one hand, support activities, and particularly the handling of information, are at the core of productivity increases in the whole economy; on the other hand...many of the information-processing activities are...prone to low labor productivity... [between 1948 and 1982] the overhead costs of [information technology intensive] production skyrocketed.[15]

Castells cites statistics worldwide to indicate persuasively that, while non-capital productivity and output per hour per employee soared in the period 1948 to 1982, the worldwide output per unit of capital remained, throughout the period, at 1948 levels, primarily due to the high cost and (relatively speaking) low productivity of IT investments.

Why is this the case? Alan Webber, writing in the *Harvard Business Review*, argues that:

> Knowledge only flows through the technology (enterprises deploy); it actually resides in people -- in knowledge workers and the organizations they inhabit....[management's] job is to create an environment that allows knowledge workers to learn -- from their own experience, from each other, and from customers, suppliers and business partners.[16]

More mundane examples exist to underscore both Castells' and Webber's arguments. The billions of dollars spent in the 1970s and 1980s to solve the information capture problem (OLTP systems) are being augmented, in the cash-poor, customer-hungry 1990s, by hundreds of millions of additional dollars spent on the information analysis problem (DSS systems), simply because the managers of the enterprise have discovered -- and none too soon -- that capturing information, in and of itself, is largely a useless endeavor: someone must be able to use that information, which is after all only history, only the past, to make intelligent decisions about the future. OLTP technology, on the whole, has not fundamentally increased the productivity of capital, nor enhanced the competitive positions of organizations with respect to their adversaries, nor brought the enterprise significantly closer to either its suppliers or its customers.

---

[15] Castells, p. 135.

[16] Alan Webber. What's So New About The New Economy? Harvard Business Review. 1994.

The mistake IT organizations are making today is the one Webber calls out so clearly: mistaking what in fact creates knowledge for the enterprise. Decision support systems -- however legible the schema, however complete the data they contain -- can do nothing more that feed the enterprise's real decision machines: the knowledge workers, and the business teams they build.

Other industry voices are expressing similar concerns. For example, the Iacocca Institute has issued a report on twenty-first century manufacturing that calls for 'agile manufacturing' in which business teams are created across companies, their suppliers, their customers and even among adversaries to build and deliver products to rapidly-emerging (and rapidly-closing) markets. These 'agile manufacturing environments' require massive software components -- called out by the Iacocca Institute report -- that in effect create temporary 'urban' environments for these cross-boundary business teams, providing them with complete support environments electronically across time and space.

So, at the very time when city planners are consumed with the 'phenomenology of the city,' with designing cities that maximize 'ease of use' and 'pleasure' for their citizens, we find economists, sociologists and IT theorists arguing for a radically recentering of IT focus at the worker both as consumer and as producer of information, and so competitive advantage.

IT Organizations (Ought To) Build IT Cities

But what is our task, really?

The first thing we should understand is that urban planning as a formal human function first came into being in response to the wild, uncontrolled proliferation of urban landscapes during the Industrial Revolution. At the height of the Industrial Revolution, when the modern science of city planning was first codified and practiced, urban planners faced two interwoven problems: the problem of urban renewal and the problem of urban planning.

In Europe, large cities were falling apart. Allowed to evolve organically since their foundings during the Middle Ages, the capitals of Europe had, one by one, grown into chaos; they could no longer support their populations, provide adequate services, or even, in some cases, sustain human life. City governments could not control their urban citizens (Paris was twice replanned in the nineteenth century simply to limit the possibility of spontaneous rioting), and, perhaps, more importantly, urban dwellers could no longer negotiate the urban network: increasingly, the very people the city depended on -- people with money and jobs -- left the chaos of the cities for neatly-planned and platted suburban towns: for navigable environments with services easily available.

City planners, studying this phenomenon carefully, argued that cities could not be allowed to simply evolve, organically and on their own, without guidance supplied by long-term urban growth plans, and without substantial investment in reworking older urban areas to (a) ameliorate the harsher immediate problems of slums, inadequate

marc@noumenal.com

water supplies and sewage disposal and (b) open up the closed, crowded Old World urban landscape to new ideas about road planning, coordinate system design and service clustering.

In the New World -- particularly the United States -- a different kind of opportunity presented itself to Old World-educated urban planners. Vast open spaces were being settled, and those settlers had need of cities: focal points for supplies, transportation and services, and, spiritually, the sign on the landscape that civilization had arrived. Urban planners imperatively felt that, in these new open spaces, they had the opportunity not simply to rework dysfunctional urban environments, as was the case with Old World cities, but to start afresh: to design, from the ground up as it were, urban environments that were calculated to provide a high quality of life to their citizens and to grow in a controlled, managed way -- in perpetuity -- in the name of maintaining and enhancing that high quality of life.

We are presented with identical opportunities. Our Old World is the world of legacy systems and first-generation open systems (many of them now in fact if not in name legacy systems): a world defined by dozens of conflicting industry and international standards for everything from kernel interfaces to SQL dialect conventions. This Old World open systems city is profoundly dysfunctional; our internal customers came to this city, with our help, from yet older and more isolated proprietary cities that were not as dysfunctional as our UNIX metropolis; they were simply expensive, parochial and in some cases confining. We promised them the splendours of Rome: more features. more services, more choices, and at a lower cost of living.

What we gave them was something substantially less than we promised.

We cannot raze Rome; we cannot start over again. What we can do is follow the examples of the nineteenth-century town planners: we can ameliorate worst of the immediate problems, and we can define a program for selected urban renewal to extent the usefulness of these cities.

Our new world is the Microsoft-Oracle-Internet influenced enterprise. Today, this enterprise is an untilled field, a vast, wind-blown prairie: there is nothing to undo, nothing to rework. Like the settlers of the American West, we are utterly free to stake claims to large amounts of territory, and design new cities from the ground up: cities that, in their design and growth strategies, reflect the lessons we have learned building the cities of the Old World.

Our greatest challenge is to keep two things in view simultaneously: the fact that our most significant opportunities to create something truly new and different -- something that is revolutionary and not simply evolutionary, something that secures the futures of the firms we work for and with, rather than just prolonging their existences -- are to be found in the New World. But our settlers -- or the vast majority of them -- are still living in the Old World hamlets, townships and metropoli we helped them build.

marc@noumenal.com

6. Building IT Cities

Every citizen has had long associations with some part of his city, and his image of the city is soaked in memories and meanings.

Kevin Lynch, *The Image of the City* (1960)

Today, the term *enterprise architecture* is in danger of going the way of *open systems*, *client/server computing*, and *free*. To be perfectly blunt, it appears to me that two fundamental things have happened:

- the idea of architecture as a holy grail that can cure the ills of today's underfunded, overtaxed IT organization has been picked up by the popular business management press

- IT vendors and consultants have retooled old -- sometimes ancient and discredited -- system implementation methodologies under the heading *enterprise architecture*, and are out selling the same wine in new bottles, a tactic sure to discredit utterly the idea of enterprise architecture almost immediately.

Factors Calling For A New Methodology

The major factors that call into question existing IT methods and practices (M&P) and demand new methods, practices and frameworks are, briefly:

1. dis-integration: of markets, customer blocks, and firms. The firm today is losing its traditional markets, its customers, and its boundaries. It is blending into, on the one hand, its upstream suppliers and on the other, its downstream channel and customer base. Traditional methodologies are unable to cross firm boundaries.

2. strategic management: rather than a mystical process that delivered, semi-annually, a set of documents or pontifications, strategy and business management are now seen as adaptive, ongoing processes within a firm that seldom reach more than provisional conclusions. There is no stable, durable, predictable "output" from the "management" process that a disconnected IT operation can use as "input" for an equally disconnected IT planning activity

3. process-centric management: instead of functions and tasks, the essential unit of orchestration for cutting-edge forms today is the business process, which is not only cross-functional and multi-task, but which actually extends beyond firm boundaries.

4. new usage models and technologies: DSS, messaging, workflow and electronic commerce are rising in the minds of IT and business professionals alike. OLTP is falling as both a primary focus of effort and a perceived source of value.

marc@noumenal.com

5. new data types and sources. The traditional methods and practices within an IT organization focused exclusively on transactionally-generated, internal structured alphanumeric data sets, and this is where most IT expenditures are focused today, in one way or another: on structured alphanumeric data generated internally by the firm. Increasingly, the firm requires semi-structured (multimedia, compound object) data, and structured and semi-structured data from outside the firm, in order to operate effectively in commercial markets. No traditional M&P practice deals effectively with these kinds or sources or data.

6. process-based competition: more and more organizations, across vertical market segments and geographies, are competing on the basis of HOW each organization performs a process the inputs and outputs of which have largely become a commodity.

7. returned business value (RBV) as key IT metric: after 20 years of studies that demonstrated conclusively that more IT not only did not guarantee more productivity, but, paradoxically, almost guaranteed static productivity, new studies from MIT and elsewhere have pinpointed the fact that the reason for the lack of returned business value from IT was a historical failure to automate business processes (instead of functions and tasks) and an unthinking reliance on narrowly-set financial metrics at the expense of more relevant second- and third-order risk-based strategic valuations.

8. reallocation of IT dollars: more and more of the 'new application' budget is in the hands of business units rather than IS organizations. These business units have the process and 'customer-in' religion and treat data as what it is: dirt, the raw material of knowledge work.

9. the new for a core competency in the IT function in the form of a flexible, extensible infrastructure. In the future, the infrastructure of the centralized IT function will move up and outwards, to include the database, middleware componentry, and fundamental toolkits for developing desktop applications. If the IT function cannot provide this, users will demand its replacement with a group that can.

10. varying rates of renovation and obsolescence: by 2000, the average lifespan of an infrastructural component will be 3 years, and the average lifespan of a business process will be 1.5 years. At that time, a "typical" company will get half of its infrastructure through value-added public service providers. Today, traditional IT methods and practices makes no distinction between infrastructure and process.

11. new ethical implications of IT. Once a technology deployed on or against a specific class of worker within the firm – the so-called production worker – IT is now directed against all classes of employee within the firm, and against all aspects of work itself, raising profound ethical questions about the proper and improper uses of IT within the firm and within society as a whole. Existing M&P is by and large value-free.

The basic problem is simple: today, there is no such thing as a typical company or a methodological approach that will allow for a 'one size fits all' approach. Simply put,

marc@noumenal.com

each firm's business is more unique than "similar to" another customer's business, and M&P must allow latitude to pay close attention to important uniqueness.

Framing The New Methodology

New methods and practices must be developed inside a framework that answers three key questions:

- what role does an IT architect play in the definition, design, development and deployment of information technology within the firm and across the firm's boundaries?
- what objects of study, inquiry, definition or control does an IT architect focus on, and what language does an IT architect use to describe her work?
- how does an IT architect evaluate her own work in the context of her peers' work, and in the context of the firm's market objectives?

Within this framework, I believe that drawing an explicit link between IT architecture and urban planning is fruitful for IT professionals searching for new methods and practices on top of which to build their own repertoire of processes, heuristics and skills for several reasons:

1. the enterprise, as I have tried to suggest, is just like a city, from an IT perspective: collections of structures or systems knit together with complex networks of communications, supporting multiple systems of communication and commerce, many of which were not known or understood when the city was initially designed.
2. urban planners have a more sophisticated and more complete understanding of their discipline than we do of ours. We can, quite frankly, steal many of their models, processes and evaluation criteria, and import them into our discipline, thereby leveraging the planet's 4000 years of experience designing, building and rebuilding cities.
3. urban planners have a more robust system for evaluating, quantitatively and qualitatively, the success and failure of their work than we do of ours. These systems have both commercial and ethical dimensions, and ours do not.

Planners' And Architects' Evaluation Criteria

Kevin Lynch, the dean of American urban planning theorists, develops a set of urban performance criteria in his book Good City Form that have compelling IT analogs. Those criteria, and their IT analogs, are listed in the chart below.

Kevin Lynch's Urban Performance Dimension	Lynch's Definition for Urban Performance	Enterprise IT Architecture Performance Analog
1. Vitality	"The degree to which the form of the	The degree to which the enterprise IT architecture

marc@noumenal.com

	settlement supports the vital functions, the biological requirements and capabilities of human beings – above all how it protects the survival of the species."	explicitly supports the firm's stated strategies to deliver a defined level of returned value to shareholders or owners, employees and customers, as well as other stakeholders. The degree to which the enterprise IT architecture, as implemented, supports the day-to-day work activities of the employees required to make good on the firm's stated business strategy and objectives. The degree to which the enterprise IT architecture, as implemented, supports the product, service and process innovations required by the firm or its employees without substantial lags in time between (a) the recognition of the need for innovation and (b) the ability of the IT architecture as implemented to help produce such innovation.
2. Sense	"The degree to which the settlement can be clearly perceived and mentally differentiated and structured in time and space by its residents and the degree to which that mental structure connects with their values and concepts…"	The degree to which the firm's IT architecture, as implemented, reflects the firm's cultural values. The degree to which the IT architecture's dominant navigational metaphors (provide simple access to required resources, arrange those services and resources in appropriate contexts, and reflect the firm's cultural values.
3. Fit	"The degree to which the form and capacity of spaces, channels and equipment in a settlement match the pattern and quality of actions that people customarily engage in, or want to engage in…"	The degree to which the firm's IT architecture, in design and in implementation, explicitly support the firm's business processes and the collaborative work of process teams as opposed to the automation, routinization and scrutiny of the tasks of isolated individuals.
4. Access	"The ability to reach other people, activities, resources, services, information, or places including the quantity and diversity of the elements which can be reached."	The degree to which the firm's IT architecture, in design and in implementation, guarantees all firm resources (subject to security considerations) are accessible to employees when needed as needed, in the context of a unified navigational model.
5. Control	"The degree to which the use and access to spaces and activities, and their creation, repair, modification and management are controlled by those who work, use or reside in them."	The degree to which the firm can explicitly control all resources made available by the IT architecture as implemented, including the passive monitoring of those resources, active management of those resources, and explicit control over access to those resources. The degree to which the firm's IT architecture, in design and in implementation, assimilates new resources and services that are introduced without substantial rework or extension of the firm's infrastructure. The degree to which the firm's IT architecture, in design and in implementation, guarantees that new resources and services, once introduced, collaborate with existing resources and services without substantial rework or extension to either new or existing resources or services themselves.
6. Efficiency	"The cost, in terms of other valued things, of creating and maintaining the settlement, for	The extent to which the IT architecture predicts accurately the first, second and third order costs of

marc@noumenal.com

	any given level of attainment of the environmental dimensions (1-5) listed above."	its implementation, the first, second and third order benefits that will accrue as the result of its implementation, the timeframes required for benefits to accrue, and the risks associated with those costs, benefits and time scales. The extent to which the IT architecture contains within it a comprehensive model for evaluating the returned business value (RBV) of any new IT project considered as an extension to the IT architecture.
7. Justice	"The way in which the environmental benefits and costs are distributed among persons, according to some particular principle such as equity, need, intrinsic worth, ability to pay, effort expended, potential contribution, or power."	The extent to which the IT architecture provides the mechanisms for the firm to allocate access to and control over resources based on its value system. The extent to which the IT architecture, in design and in implementation, strikes the appropriate balance between the empowerment of the work force and the fiduciary and legal need of the firm to control resources and personnel, legislate behavior and protect intellectual property. The extent to which the IT architecture, when it infringes upon an employee's rights or privileges, does so explicitly and in the context of an explicit policy that existed before the act of infringement.

marc@noumenal.com