4-10-2008

Information Architecture: From Structural Notion to MeaningfulCommunicative Concept

Antony Bryant
Leeds Metropolitan University, a.bryant@leedsmet.ac.uk

Rik Maes
University of Amsterdam, maestro@uva.nl

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Information Architecture: From Structural Notion to Meaningful Communicative Concept

Antony Bryant
Leeds Metropolitan University, United Kingdom
Rik Maes
University of Amsterdam, The Netherlands

Abstract
The notion of â architectureâ is widely used in the context of IS/ICT1. It is often to be found in couplings such as business architecture, knowledge architecture, strategic architecture, governance architecture, IS architecture, IS competence architecture, IT architecture, network architecture, computer architecture; the list could easily be extended. Why is the word actually used in this manner? What does it add? Could it simply be replaced by simpler, perhaps less resonant terms such as structure or framework? Our paper examines the ways in which the term architecture (in particular information architecture) and the architectural metaphor have come to be accepted and used in the context of IS/ICT. In so doing we draw attention to the important cognitive, perceptual and communicative aspects of the metaphor which have all-too-often been ignored at the expense of the structural, constructional and tectonic aspects. It is important that the cognitive features of the architectural metaphor are brought to the fore in order to highlight the full range of issues central to IS/ICT practice; which itself is a critical aspect of organizational contexts.

Keywords: IT, postmodernity, organizations, systems

Permanent URL: http://sprouts.aisnet.org/5-9

Reference: Bryant, A., Maes, R. (2005). "Information Architecture: From Structural Notion to Meaningful Communicative Concept," University of Amsterdam, Netherlands . *Sprouts: Working Papers on Information Systems*, 5(9). http://sprouts.aisnet.org/5-9

1. ZACHMAN'S ARCHITECTURE

One of the landmark uses of the architectural metaphor originated with John Zachman's paper in 1987, which introduced the concept and the model of the 'Information Systems Architecture' [ISA]. Zachman was not claiming to be the first in IS/ICT to use this metaphor; indeed one of his stated objectives for writing the paper was to offer some clear and rational basis for employing the term given its widespread but inconsistent use. His aims were quite specific as the abstract to his article states.

With the increasing size and complexity of the implementations of information systems, it is necessary to use some logical construct (*or architecture*) for defining and controlling the interfaces and the integration of all of the components of the system. *Stress added*

By the mid 1980s it was clear that information systems were critical components of their business or organizational context. In order to operate effectively such systems have to become embedded in their host environment; and in order for this to be accomplished and sustained a wide range of different perspectives and orientations have to be taken into account as part of the development and operational activities. This cannot be accomplished simply by expanding the schedule of meetings and consultations; rather it requires a qualitative revision of the process of developing such systems, accounting for the many different interests and perspectives involved.

For Zachman the way in which to resolve this issue was to use the model of architecture. He advocated starting with an initial model that was in effect no more than a back-of-an-envelope sketch; primarily devised as a basis for negotiation between the client and the specialist (architect). The purpose of this rough-and-ready diagram was at least as much to confirm to the owner that the architect understood what was in the owner's mind, as it was to provide a basis for the more technical aspects of the project. As such it served to focus negotiation between these different parties, assisting in reaching confirmation and contractual agreement.

This confirmation and harmony were particularly important to the information systems professional since the mismatch between the envisaged system on the part of the clients and users, and the finished product was an increasingly embarrassing issue. One of the main

factors underlying this discrepancy was the nature of software itself. So it was no wonder that software-based systems were often expensive failures, with research in the 1980s indicating that as much as 80% of government expenditure on software systems was wasted on systems that either were never completed or were not fit-for-purpose when delivered.

One attraction of the architectural metaphor is that it appears ready-made as a basis for inter-related activities demanding constant oversight, highly technical specification, and separation of concerns. Moreover there is an underlying assumption that ultimately client and user requirements must be satisfied. This must involve more than any limited and mechanistic views of requirements.

Zachman bases his argument on the distinction between three cognitive or perceptual domains; *owner*, *designer*, and *builder*; and the necessity to establish dialogue between them. It is imperative that some form of negotiation can be established between them. This must be a prime concern of any IS project, but Zachman somewhat weakens this point by entwining it together with a more product-centred argument. The result is that he presents the Information Systems Architecture [ISA] as offering a developmental path for different systems abstractions or models; and the models are increasingly seen in terms of representations or products. The consequence is that the cognitive/perceptual thread is lost amongst the model and product based one. Zachman specifically warns against this, and so must have recognized the likelihood of such partial understanding of the ISA. Each model or representation had to be seen to be different in nature from the others, not simply displaying a level of detail greater than previous ones.

A glance at the current uses of ISA and its associated website[2] indicates that since 1987 the structural aspects have eclipsed the perceptual ones. The framework is widely used, and there are many project offices where the 6x6 grid is pinned up, each cell populated with a model; each column seen as an increase in detail reading from top to bottom. The framework has become part of the IS body of knowledge; with high visibility amongst practitioners, even more than amongst academics and researchers. If Zachman's objective of clarifying the ways

[2] There is a website dedicated to Zachman's framework – The Zachman Institute for Framework Advancement (ZIFA) http://www.zifa.com/

in which the IS community apply the concept of architecture has been in any way successful, then it is an ironic accomplishment, since it undermines the main strength of his initial argument. The structural has won out over the conceptual.

2. THE ARCHITECTURE METAPHOR

The architectural metaphor was also used in the context of computer hardware, and the term *computer architecture* predates Zachman by at least a decade. By the 1990s this even extended to ***software*** architecture, 'an emerging discipline' derived from a combination of concepts drawn from the field of *hardware*, and those from the field of *design* as applied to software systems. In his introduction to a text on software architecture, Barry Boehm made the point that the concept of architecture in this new context emphasizes the necessity for intermediate or bridging abstractions that connect 'characteristics of systems users needs to characteristics of systems that software engineers can build'. (vii) This echoes Zachman's cognitive argument, reinforcing what must now be regarded as the neglected but critical aspect of Zachman's 1987 paper.

The combination of the common misinterpretation of the ISA, and the use of the term with regard to hardware, meant that by the late 1990s the architectural metaphor was associated with structure and configuration. Computer architecture itself had, however, taken on a wider ambit: Physical architectures now encompassed *networks* rather than individual machines; and so evoked concerns with configuration, composition and components.

We do not wish to argue that such considerations are unimportant or incorrect, but we do wish to rescue and reconstruct the cognitive and perceptual aspects of the architectural metaphor which have been increasingly over-whelmed by tendencies that focus on structural and *con*structional aspects.

3. REMEDYING INFORMATION ARCHITECTURE

One of the appeals of the architectural metaphor, specifically alluded to by Zachman, seems to have been the prominence it gave to the ways in which architectural practices and artefacts coped with the issue of how best to facilitate an understanding between different parties to any complex (construction) project. Such parties will often be non-like-minded, and consequently it is no trivial matter to reach a consensus. It can be argued that professional practice ought to include recognition that the true professional should have a better understanding of client requirements than clients do themselves. Yet there is a fine line between specialists who know their patients' or clients' needs better than the latter do themselves, and those who simply lord it over their clients. The paradox is one between insight on one hand, and arrogance on the other. There may be no easy way between the two. Some professions have tried to establish procedures to deal with this matter; and IS/ICT practitioners must do likewise.

Zachman's work appears to offer a way to achieve this. Without explicitly stating the argument as such, he intimates that practitioners with an understanding of the rationale and motivation underlying the ISA will be more likely to act with the insight and empathy required to encompass and reconcile the range of perspectives and cognitive abstractions given in the ISA itself. This in fact necessitates a major accomplishment on the part of *information architects*, since they must be capable of overcoming their own cognitive parochialism in recognizing the limitations of other people's. They must then seek ways in which these disparate perceptions can be reconciled so that they satisfy client and user requirements – real and perceived. This must form a critical part in the formation and development of IS/ICT practice.

This reinforces the importance of the cognitive ramification of the architectural metaphor, explicitly centring on Boehm's 'bridging abstractions'. If information architecture is to mean no more than information *configuration*, then it might be better to dispense with the term 'architecture' in this context altogether. If the role and responsibility of an information architect is to mean anything, then it must encompass the ability to transcend perceptual abstractions in the sense alluded to by Boehm.

In fact it appears that architects themselves have battled with similar arguments about the nature of their practices, and their discussions can shed light on the issue for IS/ICT. Roland Barthes (1967), referring to the work of Kevin Lynch, draws attention to the ways in which urban architects have to use terms that relate to those understood and used by city dwellers. Barthes specifically points out that architects have to recognize and build upon the ways in which urbanites develop and impose schemas of discrete units that break-up and render the city itself as comprehensible and intelligible. In terms that are strikingly similar to those used by the patterns movement in software – itself derived from architecture – Barthes gives examples such as *path*, *edge*, *district*, *node*, and *landmark* which act 'like phonemes or semantemes' – i.e. fundamental units of meaning. He argues that this is a *Gestalt* as opposed to a structural conception of architectural practice, and that in general there is a conflict between the semantic view and the functional one.

This concept of a *Gestalt* or cognitive practice parallels Zachman's initial rationale. It also evokes considerations similar to those of Boehm's bridging abstractions. There is a common recognition that people's perceptions are constitutive of their experienced and understood reality. In consequence modelling a context involves a negotiated dialogue that takes account of these perceptions. Practitioners – whether architects or information architects – have to adapt their practices accordingly, transcending their own perspectives in order to preside over processes seeking to achieve some form of consensus; even if this is unlikely to be a completely satisfactory process.

Umberto Eco (1973) develops these arguments in his work on *Function & Sign: Semiotics of Architecture*. He offers a very wide definition of architecture – which, with some minor modification, could readily include Information Systems (i.e. if we exclude the phrase 'three-dimensional')

> any type of design producing three-dimensional constructions destined to permit the fulfilment of some function connected with life in society (p182).

Like Barthes, Eco argues that architecture operates communicatively as well as functionally. He notes that as soon as people start to generalize from one occurrence to another – e.g. from the first cave in which they take shelter to a second cave they come across later – they generate models of generic concepts. At first these will operate on an individual or private level; but eventually such concepts will start to be used interactively and will then become *iconic principles*, and objects of communicative intercourse: Indeed they will make communication possible *per se*.

It must be pointed out that some of the most fruitful and some of the most exasperating forms of communication occur when people with differing *iconic principles* enter into dialogue; particularly if they utilize the same *sign vehicles* (see below) but find that they are pointing in different directions. This is endemic to architectural practice, and even more so in IS/ICT. The adoption of engineering terms exemplifies this; hence the continuous and sustained disputes around the meaning of terms such as *requirements*, *maintenance*, and *prototyping* when taken from their earlier contexts and re-interpreted in new ones. With regard to ICT, its constituent *sign vehicles* have undergone enormous re-design and re-grounding in the past 40 years or so. In the 1970s and 1980s teaching people about the promise and potential of computer-based technology – often termed 'computer appreciation' – involved use of sign vehicles drawn from more mundane aspects of people's existence. Thus there was extensive use of analogies and metaphors based around *desk-tops*, *files*, *documents*, and the like. In the intervening period ICT has itself become part of our everyday taken-for-granted existence; and has even taken on a defining role, so that many old concepts as well as new ones are couched in terms derived for computer-based technology. There is therefore a great deal of truth to say that we live in a computer age since so many of our *iconic* principles are computer-based, or more correctly ICT-based.

Eco argues that artefacts and models 'communicate the function to be fulfilled' (183), and that they signify this 'even when they are not being used'. Eco coins the phrase *sign vehicle*.

> Thus what our semiotic framework would recognize in the architectural sign is *the presence of a sign vehicle whose denoted meaning is the function it makes possible* … (184 – stress in original).

The sign vehicle is 'observable and describable apart from the meanings we attribute to them'. This distinction between the sign vehicle and the meaning

> permits us to recognize in architectural signs *sign vehicles capable of being described and catalogued*, which can denote precise functions provided one interprets them in the light of certain codes (184 – underlined part indicates stress added)

Eco offers the example of a flight of stairs. IT denotes the possibility of 'going up'. but the link between form and function operates through a codified connection as well as 'a conventional conception of how one fulfils the function with the form'. Thus someone confronted by an elevator for the first time would have no idea how the form fulfils the function of 'going up'. The consequence is that one cannot simply state the maxim that *form follows function*; it has to be revised as follows

> *the form of the object must, besides making the function possible, denote that function clearly enough to make it practicable as well as desirable* … Then all the ingenuity of the architect or designer cannot make the new form functional (and cannot give form to a new function) *without the support of existing processes of codification* … (186).

This has significant implications for IS/ICT and the way in which the architectural metaphor has developed. Zachman's motivation in 1987 might be seen to emanate from a period where IT (we use this term explicitly) was being used as a new, indeed revolutionary form for pre-existing functions. Thus the ISA can be seen as a response to the need to make the new forms 'functional' within existing processes of codification. In the intervening period these 'existing processes' have developed sufficiently so that previously esoteric IS/ICT-based terms are now part of common parlance – e.g. input, database, computer network, internet and the like. To some extent this explains why the meaning of the term architecture in the IS/ICT context has altered. Yet Eco's central point, that the architect has to fulfil a primarily communicative role, must not be allowed to disappear simply because the technology is now part of our everyday life and habits.

Eco develops this argument in stressing the distinction between *denotation* and *connotation.* Denotation is a far more direct or explicit aspect of signification; connotation is more diffuse and implicit. Thus a cave may have come to denote 'shelter', but in time it may also have 'begun to connote family or group, security, familiar surroundings, etc' (187). This 'symbolic function' is no less functional or useful than its primary, denotative one. Connotative functions might be termed 'secondary', but not because they are less important: They are based on the semiotic mechanism, 'in the sense that the secondary functions rest on the denotation of the primary function'. (188)

In the context of IS/ICT, architecture came to be incorporated to denote 'architecture-as-structure'; this was its primary function at least in the sense of being the original meaning. But it has failed to develop beyond this to form the basis for any widespread awareness of the secondary connotation of 'architecture-as-bridge-across-abstractions'. With regard to information architecture this may because the concept of itself has not really achieved stability. Developments in the use and range of ICT have not given any firm foundation for an understanding of the term; and the development of the internet and world wide web have engendered a range of further uses and fluidity in understanding. This fluidity encompasses those for whom IA relates to the *external presentation of information,* particularly websites, and those for whom IA refers to the *internal organisation* of (enterprise) information. For Eco denotation and connotation only operate within a context where they can draw support from 'existing processes of codification'. If there is no stability or at least temporary stasis then it is difficult to envisage how processes of codification might become established and so operate effectively. Thus it is hardly surprising that the concept of architecture has failed to develop connotatively.

In fact the ideas of *codification* and *codes* are themselves problematic, particularly with regard to visual representations. If the term *code* is used with regard to linguistic codes, then related terms such as *syntactics, semantics, meta-language* have some clear relevance; but this is not necessarily the case in non-linguistic cases. There are inherent difficulties and paradoxes in applying concepts drawn from linguistic representation to non-linguistic ones. Eco warns against those who move from a phrase such as 'the semantics of architecture' to search for equivalents of concepts such as 'words', 'phrases' etc. Furthermore, he also argues

that just because architecture '*can be described in terms of geometry does not indicate that architecture as such is founded on a geometric code*' (192/3). What is actually needed is an understanding of the base units of the architectural code, which is specific to itself; and Eco suggests three categories – technical, syntactic, and semantic.

The technical codes include engineering aspects such as 'beams, flooring systems, … etc.' The syntactic ones encompass 'typological codes concerning articulation into spatial types' and other syntactic conventions such as 'a stairway does not as a rule go through a window'. Semantic codes are the 'significant units of architecture, or the relations established between individual architectural sign vehicles … and their denotative and connotative meanings'. Eco outlines a scheme for these semantic codes, which we might now see as similar to Alexander's concept of patterns.[3] Moreover Eco points out that as these units are elaborated they become 'codifications of *already worked-out solutions*', which results in their acting as structural constraints on possibilities and potentialities. This resonates with the ways in which proffered solutions in the realm of *information architecture* operate to present standardized implementations; rather than acting as a basis for innovation and experiment. 'They establish not generative possibilities but ready-made solutions … fossilized forms'.

This concept of structure as standardized and constraining differs significantly from the Chomskyan view of (linguistic) structures as *generative* and hence intrinsically promoting innovation and novelty. This view of structure as constraint may have a stronger rationale in the realm of IS/ICT than in architecture *per se*; after all it may be better to have a mass produced system that works within the confines of a tried-and-tested solution, than an idiosyncratic one that does not. The danger, however, is that this will blind people to the potential of new developments. As computer technology was developing in the 1960s and 1970s, Stafford Beer pointed out the significant fallacy in simply using this technology to add rigour or speed to existing activities in an enterprise –

[3] C Alexander, *A Pattern Language*, OUP 1977 – in fact Alexander's concepts have been somewhat misapplied within the context of IS/ICT.

> *'the question which asks how to use the computer in the enterprise, is, in short, the wrong question. A better formulation is to ask how the enterprise should be run given that computers exist. The best version of all is the question asking,* what, given computers, the enterprise now is. *(Beer, 1981, originally 1974 - stress in original)*

Generalizing this dictum, Beer is advocating a position that continually challenges the status quo with the potentialities of developing technologies. On the other hand Eco is arguing that the possibilities of innovation are not limitless, but are constrained by the necessity of some anchoring in 'existing rhetorical and ideological expectations'. Without this anchoring innovations would have no links to existing codes and so would preclude effective communication. The architect, however brilliant in terms of technical insight and innovation, has to retain a link with prevalent codes and concepts as a basis to communicate ideas and persuade others. Eco argues that this means that the architect 'is continually obliged to be something other than an architect … forced to become something of a sociologist, a psychologist, an anthropologist, a semiotician …'. This is the counterpart to the point made earlier that the implicit assumption in Zachman's ISA is that the practitioner can and must strive to overcome the cognitive parochialism associated with any specific row of the ISA. Eco advocates that architects are 'obliged … to think in terms of the totality' and must do this regardless of the extent to which they 'have become a technician, a specialist, someone intent on specific operations rather than general questions.' He offers the slogan that 'the architect should be designing for variable primary functions and open secondary functions'.

This readily translates to the realm of IS/ICT, the role of the information architect, and the production of an information architecture. In all cases the outcomes are significantly different from currently accepted ideas. The predominant meaning of the term architecture in the IS/ICT realm is close to that of standardized problem characterizations codified in such a way that they are amenable to predetermined solutions. More critically, those who adopt or acquire the label of information architects tend to see themselves as narrowly confined specialists with a focus on specific (technical) operations – whether these are defined in terms of internal or external meanings of the term (see above). To adopt Eco's terms, the information architect must recognize the necessity to become something other than an

information architect in the restricted and technicist sense. The complexities of the enterprise – internally-oriented or externally-oriented – cannot be forced into standardized formats or codes, although these formats and codes can offer a useful starting point for discussion or negotiation between cognitively alien participants.

The mandatory starting point must be the experiences and cognitive schemas of the users, clients, or general participants within the context of the enterprise. In the urban setting Barthes emphasizes that one has to begin with the ways in which urbanites develop and impose schemas of discrete units that break-up and render the city itself as comprehensible and intelligible. In similar fashion, the information architect has to start from the schemas and cognitive components 'developed and imposed' by those who comprise, act within, benefit or suffer from, and sustain the system – however that system is conceptualized. This will involve the practitioner in a role that will encompass non-technical activities and practices, sometimes involving subversions and subterfuge. Without such a realization the tendency will be simply for the practitioner to perpetuate the status quo; thus shoring up existing structures of dominance and concomitant 'regimes of truth', which will preclude any consideration of possibilities of doing things differently or doing different things. Account has to be taken of Beer's advocacy for re-conceiving and renegotiating the nature of the enterprise.

If there is no attempt to 'think in terms of the totality', then the resulting architecture will often bear little resemblance to the reality of the enterprise. In Baudrillard's terms this is constructing a hyper-reality where the model is not simply inaccurate; but where it actually precedes and engenders the territory. The specific identity of the system is lost, overwhelmed by the 'precession of simulacra' derived from codified ready-made solutions that bear no resemblance to the specific context and are unintelligible to those involved with the system. Those professing to practice and perfect information architecture must ensure that they continually evade this failing, fostering a sustained regard for the totality.

REFERENCES

Barthes, R., *Semiology and the Urban,* 1967; reprinted in N. Leach (ed) *Rethinking Architecture,* Routledge, 1997, pp166-172

Beer, S., *The Brain of the Firm*, Wiley, 2nd edition 1981Brooks, F.P., *No silver bullet: essence and accidents of software engineering,* IEEE Computer, 1987, 20:4

Bryant, A., & Maes, R., *The Role of the Information Architect: Conquering Cognitive Parochialism,* 2005; currently available via the Research In Progress (2005) website at www.imresearch.org

Eco, U., *Function and Sign: Semiotics of Architecture,* 1973, reprinted in N. Leach (ed) *Rethinking Architecture*, Routledge, 1997 pp182-201

Shaw, M., & Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*; Prentice Hall, 1996

Sowa, J. F. & Zachman, J. A. *Extending and formalizing the framework for information systems architecture*, IBM Systems Journal 31:3, 1992

Zachman, J., *A Framework for Information Systems Architecture*, IBM Systems Journal, 26:3, pp454-470, 1987

INDEX

---

1. ZACHMAN'S ARCHITECTURE..... 4

2. THE ARCHITECTURE METAPHOR 6

3. REMEDYING INFORMATION ARCHITECTURE 7

REFERENCES..... 15

Information Architecture: From Structural Notion to Meaningful Communicative Concept

Antony Bryant and Rik Maes

Antony Bryant (Leeds Metropolitan University – a.bryant@leedsmet.ac.uk)
Rik Maes (University of Amsterdam – maestro@uva.nl)

STREAM

IT and Postmodernity for Organizations and Systems

ABSTRACT

The notion of 'architecture' is widely used in the context of IS/ICT[1]. It is often to be found in couplings such as *business architecture, knowledge architecture, strategic architecture, governance architecture, IS architecture, IS competence architecture, IT architecture, network architecture, computer architecture*; the list could easily be extended. Why is the word actually used in this manner? What does it add? Could it simply be replaced by simpler, perhaps less resonant terms such as *structure* or *framework*?

Our paper examines the ways in which the term *architecture* (in particular information architecture) and the architectural metaphor have come to be accepted and used in the context of IS/ICT. In so doing we draw attention to the important cognitive, perceptual and communicative aspects of the metaphor which have all-too-often been ignored at the expense of the structural, *con*structional and tectonic aspects.

It is important that the cognitive features of the architectural metaphor are brought to the fore in order to highlight the full range of issues central to IS/ICT practice; which itself is a critical aspect of organizational contexts.

[1] Although phrases such as IS/ICT realm, domain of IS/ICT etc seem somewhat cumbersome, our use is deliberate since it invokes both the hardware (IT) aspects as well as the more critical, but less tangible systemic ones (IS).

芽|Sprouts

Working Papers on Information Systems | ISSN 1535-6078

Office:
Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org

芽|Sprouts