Slide notes – preICEIMT09 presentation

Slide 1: Evolving Enterprise Architecture
This presentation and the forthcoming paper are the result of my participation in a workshop in September with a new group at Penn State. While the focus of the group is enterprise architecture pedagogy at the undergraduate, master, and practitioner levels, they are also engaging on a research track as well. Sandeep is a co-leader of the Penn State IST EA effort. Work he has done with Smolander concerning the use of enterprise architecture descriptions led to this presentation.

Since this is a workshop, we will present a range of topics centered on the theme of architecture as something that evolves during the enterprise life cycle and touch upon aspects of architecture that are critical to our understanding of that evolutionary process. Our purpose is to evoke discussion of these critical aspects as they relate to the place of enterprise architecture in the broader context of business management.

Slide 2: Evolutions
An enterprise undergoes many changes that are the consequence of: A, B, C. These aspects form a knot that can be almost impossible to untangle. Our claim is that enterprise architecture is a means to untangle that knot so long as it evolves as the knot evolves. So when and how does architecture of an enterprise get created?

Slide 3: Architecting
We use the term 'architecting' for the activities involved in specifying and using architecture.

Famous architects have all built practices to pursue their concepts of architecture. In the US, Frank Lloyd Wright built both very distinctive homes, like Falling Water, and a very successful school of practice that still exists. Christopher Alexander in his work on design patterns for civil architecture has inspired effort to understand the significant architectural patterns of software systems. Edsger Dijkstra led the way in early attempts to understand the separation of concerns necessary to build complicated software systems. And Eberhardt Rechtin led some of our most important system architecting efforts in the late twentieth century.

We have codified architecting practice in standards such as: 15288, 19439, and TOGAF.

For all of these practitioners and standards, architecting involves the transitions from…

IEC/ISO/WD 42010 defines architecting this way: process of… The importance of this definition is that architecting is a process throughout life cycle. It does not stop just because the architecture is itself well formed.

Slide 4: Layers of Architecting
We can identify at least three layers of architecting that occur in any enterprise effort. The first is what most people consider when talking about architecture, whether that is civil architecture or very complicated enterprise architecture. There is also a second layer of architecting that enables creation of the first. It is at this layer that the famous

architects identified earlier built their practice. And then there is a third layer that spans practices as we gain more knowledge of what is possible and what works the best in many situations.

Slide 5: 'Meta-" is relative
Without our endorsement of this particular model structuring, the OMG has defined a four layer stack of models. Notice that each layer below the Meta-Object Facility layer is an instance of the one above. The User Concepts layer is expressed as a UML model composed from the instances of UML Concepts above it. In this stack there are three (3) meta- situations, M1 is meta- to M0, M2 is meta- to M1, and M3 is meta- to M2.

Slide 6: 1st layer evolution of utility
As we focus on the 1st layer architecture description, it becomes obvious that as the description takes shape over time there are at least two distinct stakeholder communities involved. One community has concerns that are more abstract in nature, i.e. less quantifiable in terms of operational mechanisms, that they expect the architecture to address. This community is often the first to become involved since they generally fund the creation of the architecture description. The second community is most often the one expected to take the more abstract expression of the enterprise architecture and flesh it out so that those charged with actually doing the detail design work upon which the implementation effort depends can proceed in confidence.

Sometimes the order in which these two communities encounter the architecture description is reversed as when the effort is directed at describing the AS-IS architecture form the bottom up.

In any case, the person in the middle, i.e. the one actually doing the substantive work of crafting the architecture description, is the architect. The architect may belong to one of these communities but may be more successful in rendering a valid architecture description for the enterprise if they are independent of both. Independence also aides the architecting process as the architecture description evolves with the successive participation of more stakeholders.

Slide 7: Stakeholder community grows
Since architecting is a process throughout the life cycle, for a particular enterprise project we might expect it to begin with a market opportunity assessment that identifies a possibility for sales of a new product or service capability. That opportunity is passed along for development of the product or service concept. The AD is crafted as guidance to the detail designers and carried forward to implementation, operation and support. To be effective the architect needs to be involved all along the evolutionary path of the enterprise architecture so that the growing list of stakeholder concerns is properly reflected in the architecture description.

Slide 8: Other architecting layers
In addition to the layers already presented, the architecting process will produce additional rounds of life cycle architecting as components become specified in the

architecture. Each component also has an architecture for which the architecting activities must evolve a distinct architecture description.

In fact every dimension of enterprise architecture has unique stakeholder concern driven aspects to characterize in an AD.

We often use frameworks to inform architecting participants about prospective stakeholders as the AD evolves.

Slide 9: Models and architecture
Enterprise architecture and enterprise models have an intimate relationship. For many they are two sides of the same coin. For others enterprise architecture is found in a subset of enterprise models. And for others, enterprise architecture is only manifest in the operational enterprise. For our purpose we will consider enterprise architecture as a driver for the creation of enterprise models, some of which constitute an enterprise architecture description. In any event, the enterprise models drive the creation of system architectures related to the enterprise purpose. The system architecture then drives system models, some of which may be the system architecture description. The system models then drive system creation, operation, and disposal. In the realm of enterprise architecture, it seems almost impossible to talk about architecture without referring to models. We would like to consider models as a utility of architecture rather than architecture being a utility of the models.

Slide 10: Different perspectives
Recall that we indicated there are opportunities for architecture related to each component of the enterprise. So an enterprise of discrete systems will have an enterprise, and therefore architecture, associated with each of those systems as well. Therefore, as an observer of enterprise architecture, your perception of architecture is relative to where you are positioned in the enterprise/system hierarchy.

Slide 11: Elaboration Hierarchy
This hierarchy can be extensive and the many iterations of architecting raise concerns for the consistency and coherence between levels in the hierarchy. For a hierarchy of nine levels like Enterprise, family…, the architecture of a part likely has very little in common with the enterprise architecture, unless of course that part is critical to the structure of the enterprise architecture.

Many kinds of transformation occur along the hierarchy to elaborate the structure, function, and fit of each level into the whole enterprise. Each transformation must be considered in the appropriate level context and with respect to the meta-architectures for that context.

Slide 12: Life cycle evolution
Because architecting spans the phases of the enterprise life cycle, it is evolving through much of its existence. In any particular phase, it is the artifact produced in a previous phase and serves as guidance for subsequent phase activities.

While systems and enterprises that create or use them exhibit common life cycle patterns, those life cycles are distinct and often referred to with different labels for the phases they express.

One consequence of the relationship between architecture and its meta-architecture is the potential for instability in the architecture as the meta-architecture changes. Instability can occur when there is an overlap in life cycle phases across meta-levels. But stability is enhanced when there is the expected overlap in artifacts across meta-levels.

Slide 13: Stability and instability
For example, consider the life cycles of an architecting enterprise and an architected enterprise. The output of the implementation of the architecting enterprise is often a procedures manual that is then used during architecting operations. As long as the manual is static, its use is somewhat predictable. However, if the manual is changed that modification may or may not impact the activities related to the architected enterprise. A new architecture implementation procedure may expect architectural artifacts that were not previously specified and therefore fail.

Slide 14: Meta- vs. time
A meta-architecture is likely to identify processes for architecting the enterprise and these processes will occur over some interval of time, e.g. document a concept of operations from which the elements pertaining to enterprise structure can be identified. And while the Architecting Procedures Manual of the meta-architecture is thought of as static, its application changes over the course of an architecting project.

Different meta-levels have different time spectra with lower-level activity being more or less continuous and higher-level activity perceived as more or less discrete. The clock ticks at the meta-level is more granular than the clock ticks at the level below – at least that is how we perceive it when looking up. As discussed for the previous slide, this is not always the case. In particular, the uncertainty with which management must contend often conflicts with this desirable granular ordering and results in instability that reinforces the uncertainly.

Slide 15: Expressive limits of architecture
Another aspect of enterprise architecture that we need to understand better is the expressive limits of architecture. This is a particularly vexing aspect precisely because architecture is exhibited at so many extents of decomposition within the enterprise. But it is important not to confuse these various levels of architecting or push architecture to far into the realm of detail design and stifle innovation that is often the only way to achieve enterprise success.

Eden and Kazman have proposed the Intentional/Locality Thesis to distinguish architecture from design and implementation. While they have applied this distinction to the topic of software design patterns, the thesis seems appropriate at any level within an enterprise. The next slide will add formality to this thesis distinction but for now note that

both architecture and design are intentional, lacking an extensional characteristic. They are distinguished by the concept of locality, i.e. is applicability to all or only a part of. With this distinction much of the conversation about architecture becomes a conversation about design.

Notice that there is a missing piece in this distinction of enterprise aspects. Much experience informs us that this missing piece is really trouble and occurs with far too much frequency.

We want the description and use of architecture to be intentional and non-local in the particular context in which it is presented.

Slide 16: Architecture more formally
More formally then, intentional specifications provide infinitely-many possible instances. Consider again the OMG model stack where there are many possible platform specific instances for any platform independent instance and thus PIM is intentional with respect to PSM.

The locality criterion is a bit more difficult to understand. The example cited is that of a "universal base class" often found in Object Oriented programming as a means of providing uniform object identify and access. The use of this construct is architectural because it is intentional in that it sits at the top of the class hierarchy and it is non-local because every class must belong to this class.

What is not clear is that this distinction of locality translates well beyond the domain of software engineering. What is clear is that to disambiguate levels of architecture we need to be able to distinguish architecture from detail design. At present, much of the work effort occurring under the guise of architecture is in fact just detail design and what is then called detail design is actually implementation.

Slide 17: Stakeholder utility
Earlier we mentioned the utility of architecture resulted form the models in its description. For an enterprise the succession of stakeholder concerns that are addressed by architecture models evolve the architecture description toward completeness. As it evolves, the intention of one set of stakeholders is captured as a description that is passed along to elicit more concerns from additional stakeholders. When the expectation for additional concerns ceases or no more concerns are forthcoming, the architecture description is essentially complete.

While the utility of the architecture description is found in its expression and guidance relative to stakeholder concerns, the utility of the architecture is realized by the service it provides those stakeholders. If the enterprise architecture is not of service to managers of the enterprise then it probably has little or no utility.

Slide 18: AD Meta-model

For the past several years an ISO working group has been preparing a revision to the IEEE 1471, Recommended Practice for Architecture Description of Software-Intensive Systems, that is intended to extend the standard for the full range of architecture description situations. Without going into detail, notice that the architecture description meta-model is depicted using UML notation, which is consistent with its software system origins. Beyond the realm of software engineering, this figure can be hard for people to comprehend. Fortunately there is supporting text in the form of normative statements about the entities depicted and their relationships. But it does not tell us anything about how the AD evolved or how it is to be used.

Slide 19: AD as boundary object
In a less rigorous manner consider the architecture description as a boundary object, i.e. an object that serves as an interface between different communities of practice. The various stakeholder concerns are addressed by an architecture description that must be understandable both to those stakeholders so that verification is possible and to users of the architecture. Smolander, with refinements by Purao, identified four different ways in which the architecture description is used in the domain of software engineering, which can be extended to enterprise situations.

Some practitioners use the description as a blueprint specification for implementation. Some use it as literature for current and future users. Some use it to communicate with others for achieving a common understanding. And, still others use is to make decisions about implementation. While all of these uses are inter-related, each has a different demand on the architecture description. How we create architecture descriptions to serve these diverse needs throughout the enterprise life cycle is for more art than science.

Consider that each of these uses has an interface with the object that is the architecture description, four different object type interfaces. Then to provide common understanding among the communities of practice that each interface serves, there will need to be ways of transforming the architecture description content into meaningful information at the interface. As the description and the architecture it describes evolve, more transformations of intention and description will also occur.

Slide 20: Evolutionary transformations
In evolving from an abstract concept to a concrete realization, architecting of the enterprise involves many transformations to achieve a description that is meaningful to the broad range of stakeholders identified and useful to those who must effectuate the architecture in enterprise operations. We will present five kinds of transformations that occur: projection….

Slide 21: Projection
Projection is the most common way of extracting some portion of a model or set of models for use in a specific context. An SQL query on a relational database to produce a new relation is widely used data projection method. In the realm of architecting, the operational phase of an architecting enterprise may project through a catalog of models to

select those most suitable for a new enterprise. Earlier today we heard about a work of 60 major management models that could serve as such a catalog.

Slide 22: Instantiation
Recalling the OMG stack presented earlier, each level 0, 1, 2, 3 was an instances of the meta-level above. Instantiation is the mechanism by which a meta-architecture is used to produce architecture in much the same way a meta-model is used to produce a model. However, in the meta-model to model instantiation there is usually present an implicit process for doing so based upon some underlying formalism. Such a process must be explicit when instantiating enterprise architecture from a meta-architecture. One criticism of the Zachman Enterprise Architecture Framework is its lack of mechanism for creating the identified enterprise models. TOGAF on the other hand is much more explicit about the processes to follow in creating the IT enterprise architecture.

Slide 23: 'Meta-' as abstraction
In the domain of database information systems we can identify four distinct information spaces, each composed of two levels and successive overlap of levels between spaces. As a practitioner we typically work in a space that spans two meta-levels, with instances and their abstractions. Modelers and architects commonly work across three levels, sine their modeling tool kits are defined one level up and they must model instances one level down.

In this figure, abstraction occupies the vertical plane with the most abstract concept space at the top and the concrete real world at the bottom. The instance level of the Concept space could also be depicted as the E-R meta-model for E-R models, an instance of which appears in Model space.

In that particular model we see another kind of transformation, specialization. In this case the Employee entity is a specialization of the Party entity. Customers and vendors are also common specializations of the Party entity.

Slide 24: Specialization
Specialization occurs by adding information attributes to an existing information construct. The elaborated construct becomes a “kind of” or “sub-type of” the original rather than a “part of” the original.

Recall the earlier discussion about the different interfaces an architectural description needs to support its various stakeholder communities. Each of those interfaces reveals a description sub-type of the architecture specialized for that stakeholder community.

Slide 25: Refinement
The “part of” transformation is performed by refinement. Refinement can occur in two different ways. In the first, an entity or relationship is decomposed into constituent parts and each of those parts is elaborated in more detail. In this figure, the entities within a frame of the Zachman Enterprise Architecture Framework is elaborated as a new Zachman Enterprise Architecture Framework by adding all of the framework details

associated with that entity. Such an approach can be useful for complicated system-of-systems situations.

The second way to achieve refinement is to add details to an existing construct without creating something different. For example, you have all experienced a check list of attributes that you can see in a report. The more attributes you check the more information you receive in the report. Refinement adds more attributes to the check box list.

Slide 26: Mapping (other transformations)
And finally, mapping includes a wide variety of direct assignment of elements between models and other kinds of transformations not previously identified. For example, an element of one model could be derived from a function applied to elements of another model.

Here we depict elements from a Zachman Enterprise Architecture Framework description being mapped into a ISO 19439 life cycle integrated model description.

Slide 27: Meta-mixing
One final thought! Meta-architectures can be used together. Here is shown the ISO 19439 life cycle phase architecture structure on the bottom right for use during a To-Be modeling activity. At various phases, processes defined in ISO 15288 are used to create artifacts appropriate for each phase.
On the left is a Zachman Enterprise Architecture Framework used to capture the As-Is architecture from the bottom up – observe the reality and abstract the design and concepts in use. This information can be used going forward into the To-Be activity.

Three meta-architectures applied to one architecting activity. But be careful not to mix the meta-architectures inappropriately. For example, the Architecture Design Process must take into account both the Stakeholder Requirements Definition Process results that identify stakeholder concerns that the architecture description must address and knowledge of architecture related behavioral capabilities identified in the As-Is analysis.

Sldie 28: Discussion
There are many aspects to the evolution of architecture and architecture descriptions with which an enterprise must be aware. Here we have presented a few to evoke discussion about the evolution of enterprise architecture and the relevance of the aspects presented.
Thank you for your attention.