IBM Research

Architects' Workbench ("AWB") – A Tool for Architectural Thinking and Modeling

Contributors:
Steve Abrams, Bard Bloom, Paul Keyser, Doug Kimelman, Eric Nelson, Wendy Neuberger, Tova Roth, Ian Simmonds, Carl Spencer, Steven Tang, John Vlissides

Architects' Workbench

What is Architects' Workbench ("AWB")?

(Pragmatic view)

- A tool used by IT architects (application, infrastructure, ...) to produce models and work products

(Philosophical view)

- A tool that supports the techniques and thought processes used by practitioners in analyzing architectural information and synthesizing solution designs

("Effect")

- Works *with* practitioner instead of getting in the way: "AWB seems to understand something about where I need to go, and helps me get there"

(Technology)

- Customizable to method/process and metamodel

Architects' Workbench (cont'd)

What is AWB not?

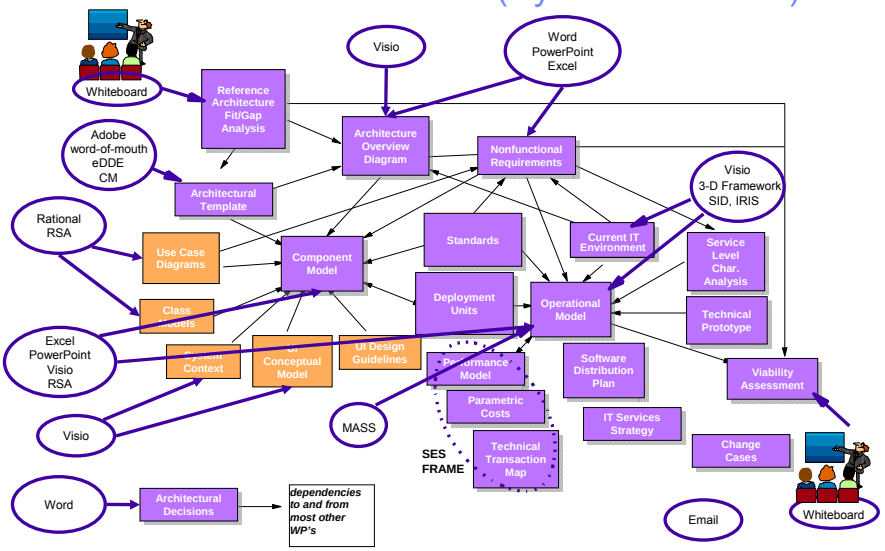
- AWB is NOT for (but feeds from/into):
  - business architecture modeling
  - business process modeling
  - detailed class design and code generation
  - detailed server and network configuration / provisioning / deployment
  - etc.

- **AWB is NOT a product**; it is a research prototype (although it continues to be used as a proprietary tool by IBM consultants on significant architecture engagements for major customers).

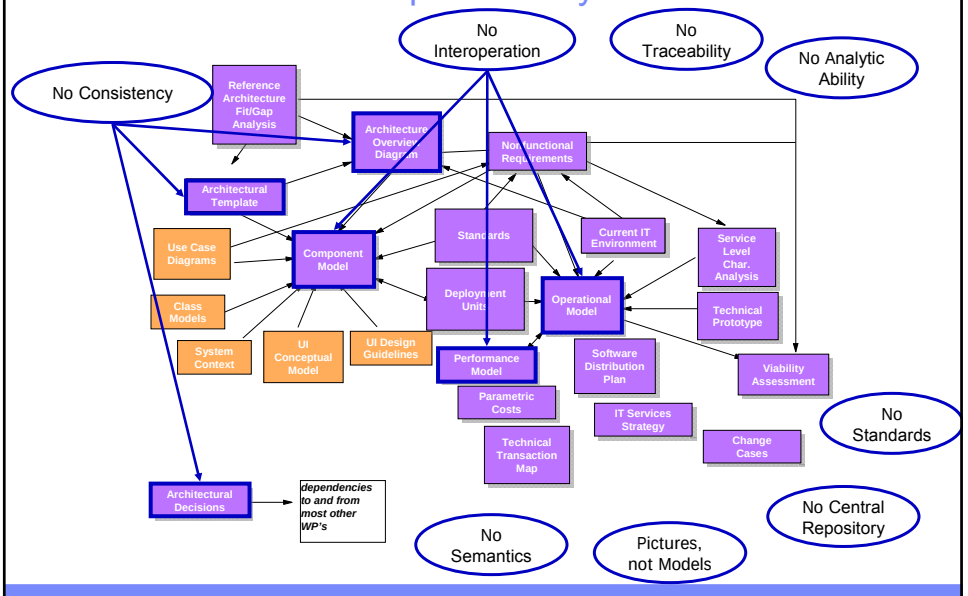
Architects' Workbench – Part I

Part I – Principles

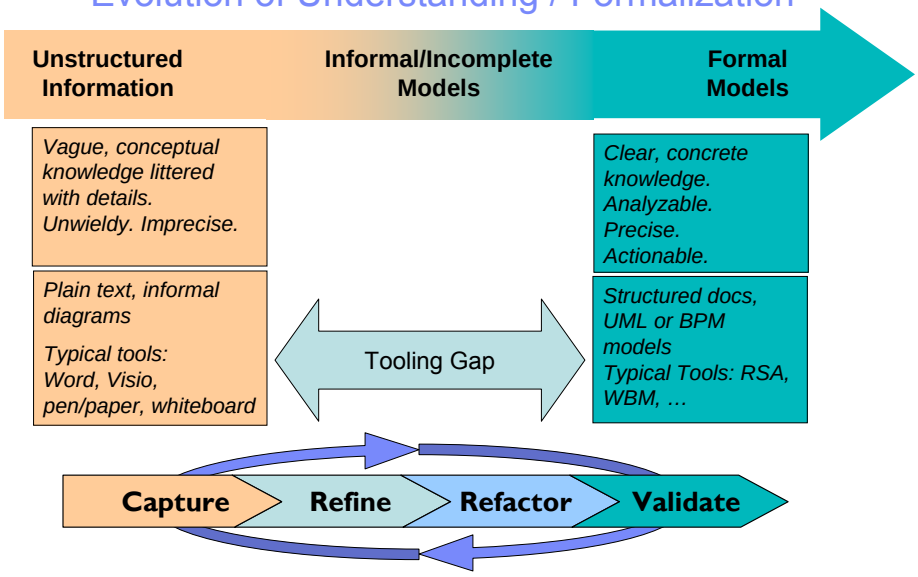
Tools Architects Use (By Work Product)



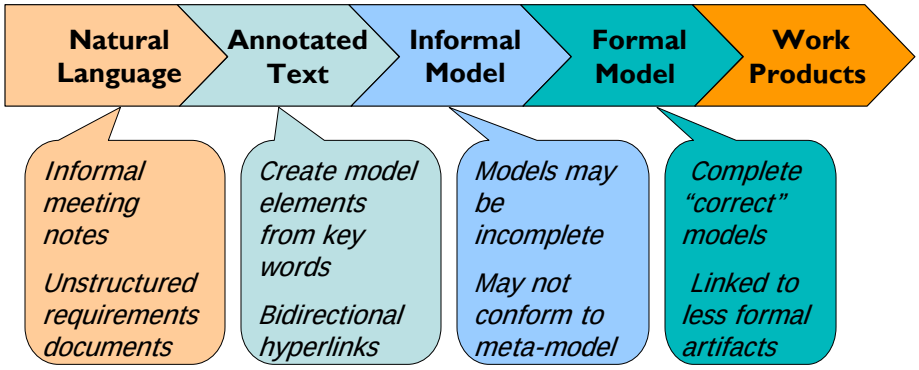
Concerns Expressed By Architects



Evolution of Understanding / Formalization



(Iterative) Evolution with AWB



AWB provides guidance, helps remember details

Refactoring wizards assist in model refinement

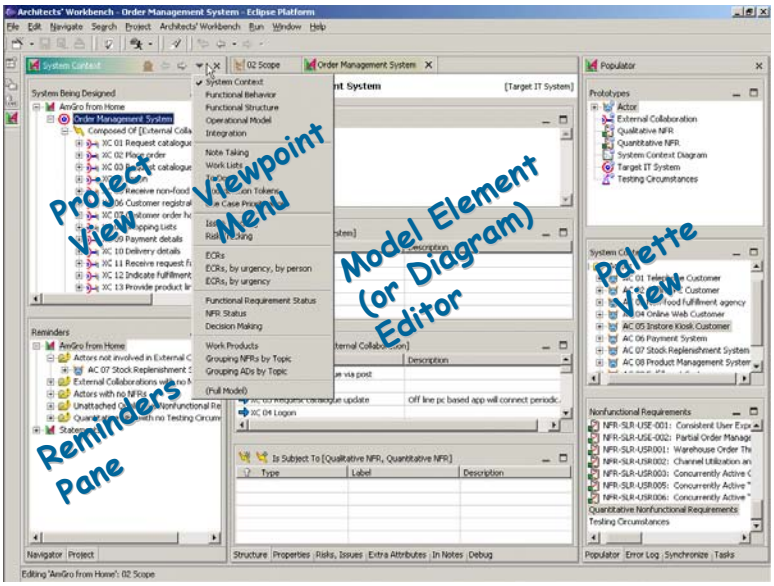
Architects' Workbench (cont'd)

AWB Approach

- ***Gather, structure, and maintain***, the information that constitutes the ***architecture*** of an IT System (or *collection* of IT Systems)
- Emphasis on ***evolution*** from ***partial informal*** overlapping and inconsistent information into ***precise formal*** models / "actionable" specifications
- Maintain information in ***one set of models / one composite model*** in a central repository, then generate ***many formatted work products*** as accurate consistent up-to-date reports, at any time
- Nature of information includes:
  - system scope, use cases and volumetrics
  - non-functional requirements (performance, availability, security, ...) and constraints
  - application and infrastructure software components
  - hardware topology and configuration / parameters
  - deployment specifications
  - architectural decisions / rationale
  - etc.

Architects' Workbench – Part II

Part II – Prototype



Preface to AWB Demo

- For an in-depth description of AWB, please see:
  "Architectural Thinking and Modeling with the Architects' Workbench",
  IBM Systems Journal 45(3) 2006.
- Demo time constraints →
  - Just touch on (distinctive) features of AWB
  - "Smaller than life" (greenfields!) examples (but nonetheless representative)

Part III – Advances

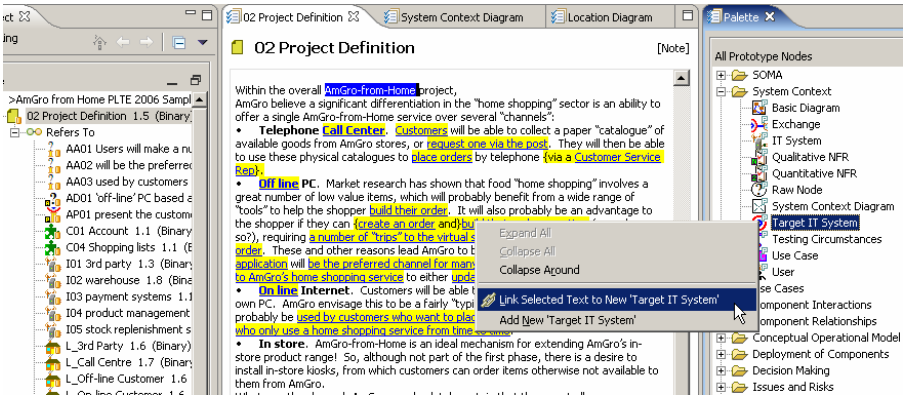
- AWB supports the creative process of architectural thinking
- Practitioners remark that it helps them focus and get where they're going, instead of getting in their way

Key to AWB Design – “Outside In” / Requirements-Driven

- AWB from the beginning has been driven by stated user (architect) needs... is continually validated in field trials... and is refined / evolved based on feedback.
- AWB continues to be used in significant production engagements – users are delighted with the AWB approach, even as they identify areas for evolution and are clamoring for improved implementation

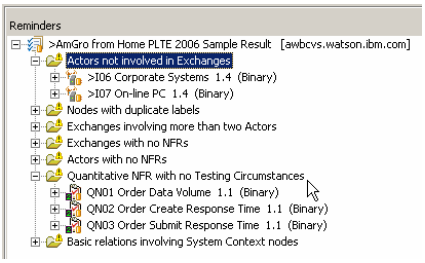
Key Advances of AWB

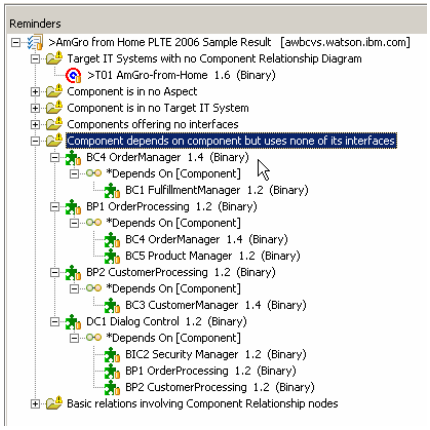
- Markup ‘N Model: Mark up free-form notes to identify and link to model elements



Key Advances of AWB (cont'd)

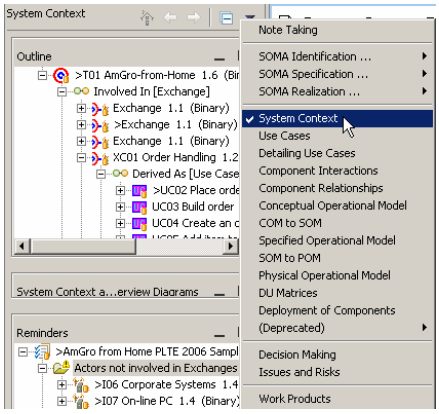
- Reminders: Validation concerning model completeness / consistency

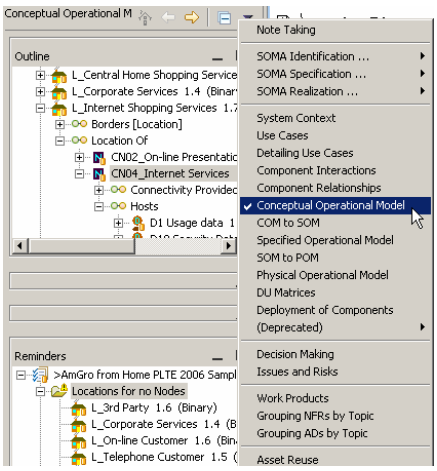




Key Advances of AWB (cont'd)

- Viewpoints
  - Task-specific slice of the semantic net / views of model, reminders, and palette





Key Advances of AWB (cont'd)

- Customizable work product generation and template definition

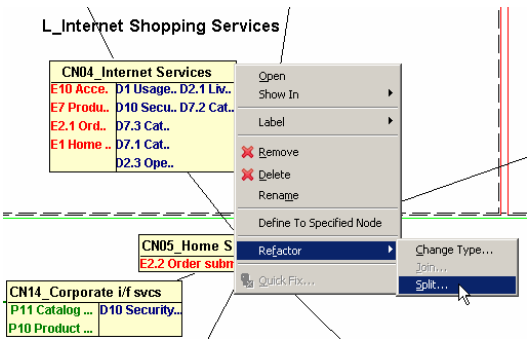
Physical Build by Location

Application Zone

Physical Node	Machine Type	Conceptual Node(s)	Deployment Unit(s)	Technical Deployment Unit(s)
SBYMIIS003	HS20 w/ 2.5 RAM	External IIS Node	Financial Calculator Xxxxxxx and PDF Library Xxxxx Web Xxx xxxx COM+ Web Component	MS IIS Xxxx Server
SBYMVIN004	HS20 w/ 2.5 RAM	Web Analysis Collecting		Xxxx xxxx xxx Server
SBYMWAS003	HS20 w/ 2.5 RAM	External Application Server Node	Xxx Application Xxx Application Xxx Hub Application Xxx Web Calendar Application Web Services Broker Application	MQSeries Server WebSphere Application Server V6
SBYMWAS004	HS20 w/ 2.5 RAM	External Application Server Node	Xxx Application Xxx xxxx Application Xxx Application Xxx Hub Application Xxx Web Calendar Application Xxx Services Broker Application	MQSeries Server WebSphere Application Server V6
SBYMWAS005	HS20 w/ 2.5 RAM	Xxx xxxx Application Server Node Web Analysis Collecting	Adobe Form Server Service Xxx xxxxx Application	Xxx xxxxx Xxx xxxxx xxxxx Server WebSphere Application Server V5
SBYMWAS503	HS20 w/ 2.5 RAM	External Application Server Node	Xxx Application Xxx Application Xxx Hub Application Xxx Web Calendar Application Web Services Broker Application	MQSeries Server WebSphere Application Server V6

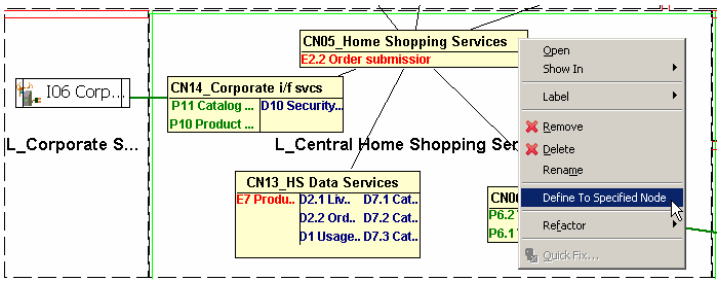
Key Advances of AWB (cont'd)

- Model refactoring
  - Split, merge, change type of model elements
  - Derive / refine model levels
  - Redirect or remove relations en masse



Key Advances of AWB (cont'd)

- GSMethod Operational Modeling support
  - rich semantics → domain-specific diagrams and operations



AWB Innovations & Features To Date (Summary)

- Opportunistic Modeling
  - Mark up free-form notes to identify and link to model elements
  - Validation reminders concerning model completeness / consistency
- Viewpoints
  - Task-specific slice of the semantic net / views of model, reminders, and palette
  - Best-practices, process, reminders, and anti-patterns
- Versatile model manipulation
  - Complementary power of textual / hierarchical and diagrammatic editing
- Model refactoring
  - Split, merge, change type of model elements
  - Derive / refine model levels
  - Redirect or remove relations en masse
- Work product and report generation
  - Customizable template definition
- Operational Modeling support
  - Rich semantics → domain-specific diagrams and operations
- End-to-end representation, linking / relations throughout
  - Requirements through deployment
  - Decisions, issues, ...

Part IV – Reality Check

"Customer R"

- Infrastructure Re-architecture project
  - Moving from thick → thin client
  - 100's of nodes for ~ 11,000 end users
    - 95% replacement of infrastructure
  - Architecture developed "***in far less time and with far less effort than it would have taken using traditional tools***."
  - "***AWB is now truly a viable tool for generating and manipulating large complex architectures for real engagements***"
- Status:
  - Architects to hand over the architecture to the development design authority – "***AWB will become the place that the architecture is held and manipulated"***.

All quotes from Infrastructure architect

"Customer L"

- Moving customer data centers to hosting center
  - Complex 3-tiered system, with complex firewall rules, high-security, and high-availability
- As part of re-hosting, application and infrastructure architects used AWB to consolidate 38 web applications → server reduction: 86 to 36
  - Involved: documenting functional application components and their dependencies, mapping them to conceptual nodes, and realizing those with the physical nodes that became the servers
  - Also: mapping middleware components based on application requirements
  - Ultimately: Custom server build work product, giving machine config, middleware requirements and deployed application components, fed into build sheets handed over to the server build team.

"Customer L" (cont'd)

- Use of **AWB saved 200-300 hours** (25%-50%)
- The architects estimate that **on the next round of consolidation**, having used AWB for the first round, and having the AWB-based repository as a starting point, **AWB will save them between 60%** (vs. projects where there are no docs from previous rounds) **and 10%** (vs. projects where some docs get done in previous rounds using tools like Visio and MS Word)

*[…] **Customer L is reaping the benefits** from using the tool: The time it took to produce the new mapping of application components to physical servers was **significantly reduced** from what it would have been without the tool, and a **repository documenting the operational models of Customer L's 38 web applications has been established**.[…]*

*The creation of this repository is significant in that it **enables a consolidated functional and operational view** of all of the web applications that was not possible to achieve before with MS Word documents and Visio diagrams. Components are re-used across multiple applications; before AWB that would mean documenting the same component in multiple Visio diagrams and MS Word documents.*

***Keeping that information up to date and in-sync was difficult, if not impossible**. With AWB the architecture team is confident that information will be kept up date and accurate.*
– IT Infrastructure Architect

Other Pilot Engagements

- "Customer B":
  - Large-scale change program (>1000 project members) mostly SAP
  - Using AWB for architectural thinking and component modelling
- "Customer S":
  - Infrastructure to support manufacturing
- … many more …

*AWB enables us to **structure our architectural** work, **increase effectiveness** producing architectural work products and makes it **easier to reuse and share** architectural designs across different practice areas. Certainly **we do not want to go back to those Visio and Powerpoint architectures** […]*

-- Architect

Application and infrastructure architects' response?

"The excitement that the tool generated actually was a catalyst to getting architects from the two sides to work together."

"I used to shudder when a customer asked me 'by this afternoon, can you get me a picture of what my application looks like in our [current] environment?'"

"Components are re-used across multiple applications; before AWB that would mean documenting the same component in multiple Visio diagrams and MS Word documents. Keeping that information up to date and in-sync was difficult, if not impossible. With AWB the architecture team is confident that information will be kept up date and accurate."

"The creation of this repository is significant in that it enables a consolidated functional and operational view of all web applications; that was not possible to achieve before with MS Word documents and Visio diagrams."

"We don't always have all of the viewpoints for a system all diagrammed out. The infrastructure is huge. Outsiders [who don't understand this] ask us for 'a picture of the IT environment'(!) [The filtering in AWB's diagram and work product generation was a big help.]"

"Without AWB we'd still be documenting [this consolidation], or maybe we wouldn't be documenting it at all."

"Without question, hand-offs are a big problem [(e.g. application architects to infrastructure architects) (Transition to Operation)]; with AWB it was a seamless transition"

QA architect's response?

"Use of AWB made QA much easier – a review that could take two weeks took only one."

"AWB helped the architects follow the method closely. The models and work products were all very much the way I needed to see them. I was impressed with how well and how clearly all the components, dependencies, and DUs were laid out. And AWB allowed me to navigate it all very quickly. Overall, AWB made the architecture really easy to verify."

"Often, there are a lot of calls back and forth, with me asking architects about things I don't see in whatever documents they provide. In this case, there were just three calls to ask why something had been done a certain way, or why something didn't appear to have been considered, and AWB allowed very effective communication."

"I was able to get up to speed and productive with AWB remarkably quickly"

Architects' Workbench – Part V

Part V – Lessons Learned

Dispatches From The Front

- It's brutal in the trenches – time is precious
  - → tools better be Zero Impedance ™ :-) to learn and use
- Practitioners in the heartland are overwhelmed by mountains of metamodels, process / method, best practices, ...
  - → tools providing roadmap / guidance will be greatly appreciated
- Practitioners are not CS majors; they're more likely to be business or IT majors
  - ("Graph?? My stock portfolio value over the last year??")
  - → modeling tools will be more effective with a **document editing paradigm** than with a graph manipulation paradigm (whether textual or graphical) (for some users)

    and
  - → tools will be more effective if they allow elision / elaboration – progressively more detailed view (and manipulation) of the (meta)model

Bottom Lines / What Really Resonates With Practitioners?

- It's not just about entering finished models;
  it's about supporting architectural thinking / the creative process
  - What to think about when, how to make decisions, what info to combine
- It's not just about the application / functional side;
  it's also about the operational side – infrastructure and deployment satisfying NFRs
- It's about integration with legacy, and about fitting into an enterprise architecture;
  "greenfields" is a dream
- It's about phases / staging – as-is, to-be, stage i+1, …
- It's about a team, not an individual, and the team includes the "uninitiated" (w.r.t. architecture)
  - SME, business analysts ← architects → developers, operations staff
  - Consequently, ...
- It's about integration with other tools (including Word!), and roundtripping
- It's about flexibility / tailoring, and polished deliverables – customers demand it

The Ideal

- Utopia would be leverage off the model by (semi-)automation of:
  - Consistency across a wide assortment of up-to-date / consistent views
  - Traceability / impact analysis
  - NFR formalization, propagation, analysis – load, performance, capacity,
  - Scenario walkthroughs (for validation)
  - Derivation of lower levels from higher levels
- Holy Grail:
  - **Asset Reuse**:
    - "Show me any previous architectures based on split server reverse proxy with orange book B1 security and eight nines of availability, for banking, based on pSeries Linux"
    - "How does it match with the architecture I've built up so far?"
    - "Extricate this 'cloud' from that reference architecture and 'graft' it into my architecture-in-progress at this place here"