Knowledge Management and XML

Kurt W. Conrad

Director of Knowledge

Tomorrow Farm

Documation 2000

Montreal, Quebec

June 6

Background

- Director of Knowledge for Tomorrow Farm
- Experience
  - Enterprise document management
  - Strategic planning and mediation
  - Process and methodology development
  - SGML, XML, and related standards
- kurt@tomorrowfarm.com, conrad@sagebrushgroup.com

Tomorrow Farm

Strategically integrates creativity and technology to produce effective digital media

- Highly-dynamic, database-driven web sites for cutting-edge dot-coms
- Film and video production for advertising, marketing, and corporate communications
- CD-ROM and DVD production for marketing, instruction, and entertainment

www.tomorrowfarm.com

What this is about

- Report from Silicon Valley
  - Emerging “New Economy” business models
  - Emerging “Knowledge-enabled” technology architectures
- What I’ve learned applying KM and KE
  - Fundamental understandings
  - Specific techniques
  - Work in progress

Agenda / Outline

- Market forces and other trends driving complexity and interest in KM
- What this means to organizations and systems
- What this means for XML
- Applying KM and KE principles to the development of markup languages

Market Forces and Trends

- More complex decision making
- Demand for simplicity
- Paradoxical demand
- Impacts on solutions providers

Market Forces and Trends

More Complex Decisions

- More decision points
  - Increasing autonomy
  - Decentralized decision making
  - Demand for options
- Decisions — themselves — becoming more difficult, complex, and costly
- Decisions must be made more quickly

Market Forces and Trends

Quest for Simplicity

- Complexity drives desire for simplicity
  - – Fewer, more targeted choices
  - – Context-sensitive decision support
- Less effort: All gain, no pain

Market Forces and Trends

Paradoxical Demand

- One on hand, customers want
  - Full decision making authority
  - The widest possible array of options
- On the other hand, customers want
  - To be relieved from decision making requirements
  - Turn-key solutions

Market Forces and Trends

Impact on Solutions Providers

- Shifts decision making costs upstream
- Community-focused “portals”
- High-performance, dynamic documents (“decisions, not documents”)
- Intelligent, context-sensitive, “knowledgeable” behavior

Organizational and Technology Trends

- Conceptualization as the critical differentiator and determinant of value
- Translation of conceptual frameworks into semantically-rich
  - – Knowledge architectures
  - – Knowledge representation standards
  - – Associated artifacts

Organizational and Technology Trends

Commoditization

- Capital
- Automation
- Roles (Sales, Marketing, Engineering)
- Startup infrastructures and incubators
  - “Your logo here” dot-coms
  - Business planning
  - Idea development

Organizational and Technology Trends

Competitive Advantage

- The Compelling Idea
  - Unmet need (market value)
  - What to build (right thing)
  - How to build it (unique know how)
- Speed
- Intellectual Property
- How to think

Organizational and Technology Trends

Knowledge Architectures

- Golden arches: Billions defined
- Segmentation of problem and solution spaces
  - Components, boundaries, interfaces, interactions, dependencies
  - Knowledge requirements
  - Value and meaning of Knowledge artifacts
- Dynamic problem and solution spaces
- How to learn

Organizational and Technology Trends

Knowledge Representation

- Meaning is the link between artifacts and behavior
- Complex, context-sensitive behaviors require rich semantics
- Semantic models need to be formalized and standardized for reliable automation

Implications for XML

- Exploding interest
  - – Industrial-grade solutions
  - – Open standard
  - – Implicit values (embracing, encompassing, local control)
  - – “Semantic richness”
- Better tools, lower cost

Implications for XML – Limitations

Semantics Gap

- Semantically thin
  - Largely semantically neutral
  - Containership, naming, validation
- Pulled in competing directions
  - Documentation (structural validation)
  - Data transfer (formal ontologies)
  - Internal coding and messaging standard
- Meaning can never be fully codified (syntax standards)

Implications for XML – Limitations

Process Gap

- How to design a markup language
- How to make policy and design decisions
  - Anticipation of uncertain future requirements
  - Balance with operational requirements
  - Social decisions
- Resourcing issues

Developing Markup Languages

- What does it really mean to manage Knowledge?
- Knowledge Engineering principles

Developing Markup Languages

Managing Knowledge

- Make investment decisions
- Assess basic KM philosophy
- Settle on appropriate Knowledge Engineering (KE) strategy

Developing MLs – Managing Knowledge

Making Investment Decisions

- Get everyone engaged
- Figure out what problem you're trying to solve and what problems you're not solving
- Determine how you will know when the problem is solved
- Identify major constraints, issues, and barriers to change
- Develop strategies for dealing with issues

Developing MLs – Managing Knowledge

Assessing KM Philosophy

- Engineered
  - Specific objectives, linear train track, no change
- Dynamic
  - Don’t fully understand problem or solution
  - Some learning involved
  - Unstable balance of competing objectives
- Organic
  - Not sure of objectives or don’t want to pin down
  - Creative, elaborative change and/or adaption

Developing MLs – Managing Knowledge

Clarifying KM Philosophy

- Reality is that likely to see a mix of engineered, dynamic, and organic themes
- Segment and differentiate
- Give each goal and subgoals clear objectives in only philosophical area
- Segment change along philosophical lines

Developing MLs – Managing Knowledge

Determining KE Strategy

- If Engineering
  - Behavior-based engineering of K flows
  - Define performance targets and work back
- If Organic
  - Artifact-based engineering of K assets
  - What K can be stored in databases and embedded in documents?
- What is the meaning and value of information?

KE Principles

- Selection of engineering targets
- Capture of transforms
- Capture supporting rationales
- Tacit Knowledge capture
- Segment ontologies
- Miscellaneous Principles

Developing Markup Languages – KE Principles

Engineering Target

Performance Target	Engineering Target
Individual Behavior	P: Artifacts (content) S: Behavior (method)
Organizational Behavior	P: Behavior (process) S: Agent (organization)
Automated Behavior	P: Agent (tool) S: Artifacts (data stds)

Capture Transforms

- Provides context
- Allows artifacts to be more efficiently “backed-up” and recontextualized
- Critical transforms can be encoded using markup
  - Revision control
  - Origins

Capture Rationales

- Codification of decision and design rationales creates “resilient memory layer”
- Capture all ideas, opportunities, desires, needs, and issues
- Record decision points, constraints, logic, confidence levels, alternatives, and triggers to revisit decisions

Tacit Knowledge Capture

- Tacit knowledge cannot be captured directly
- Can be inferred from stories
- U2 Example

Developing Markup Languages – KE Principles

Segment Ontologies

- Separation of content from formatting
- Next generation segmentations
  - Terminology and syntax
  - Semantics (common and context-sensitive meanings)
  - Rules (constraints and other behaviors)
- Formalization ranges from informal (human use) to highly formal (intelligent automation)
- Formalization is never complete

Developing Markup Languages – KE Principles

Miscellaneous Principles

- Individuals and automated agents require different semantic/markup strategies
- Consider use of semantic indirection
- Use primary axis of conditionality for element names (overlapping hierarchies)
- Define optimal granularity
  - Varies throughout lifecycle
  - Multi-contextual interfaces (encyclopedias)

Wrap - Up

- Complexity is forcing the development of more responsive, context-sensitive organizations and technologies
- This is driving interest in semantically-rich knowledge representation standards

Wrap - Up

- XML is often considered to be semantically rich
- But differences between the capabilities of individual and automated agents prevent any XML application from being semantically complete
  - Tacit knowledge of meaning
  - Implicit knowledge of meaning

Wrap - Up

- Knowledge Management can help ensure that the right problem is being addressed
- Knowledge Engineering can help ensure that problems are addressed completely
  - Engineering of processes
  - Engineering of artifacts