Management of Ontology Projects that Rely on Virtual, Volunteer Teams

Kurt Conrad
2004.07.06

Introductions

- A healthy community is self-organizing
- This is a report of an informal Ontology Management Team
  - Kurt Conrad
  - Bo Newman
  - Bob Smith
  - Joe Beck

Purpose

- To report lessons learned from an open community's work to develop a formal ontology (the Ontolog community's UBL ontology project)
- Use the lessons learned to point to essential characteristics of a systematic Ontology Management (OM) program

Lessons Learned

- Weak team structure / hierarchy / defined roles
- Resourcing (personnel and by implication, funding)
- Engineering-centric strategy of "making everyone an ontologist" didn't work
- Staring small would be easier than starting with a large conceptual scope
- Too many "observers"
- Diverse set of backgrounds gave the team a strong foundation

Lessons Learned

- The core team demonstrated ongoing commitment to the effort through their ongoing participation (on phone and in person)
- Loose objectives and performance targets helped drive positive external perceptions
- Web-based communications infrastructure worked well
- Project launch and definition was limited. Resulted in ongoing alignment issues.
- Lacked a traditional project management

Lessons Learned

- A number of key decisions were driven without a majority of the participants fully understanding or discussing the downstream implications
- Relied on an incomplete methodology. Just as there was no single project manger, no one took responsibility for maintaining / refining our methodology as such changes were required
- Early polling of the Ontology community indicated that many joined to learn more about developing ontologies, they represented much of the core participants and lacked the technical skills necessary for OE

Lessons Learned

- Resulting work processes were not well-suited to exploiting the skills of the volunteer base
- Participant's skills were not adequately augmented with effective training programs

Introducing Ontology Management

The Problem

- The conceptual formalization central to Ontology Engineering (OE) is emerging in a whole variety of technological cloaks.
- They share a common set of management issues that are largely independent of technology, approach, or targeted solution space

Goals

- Develop a reliable method for driving ontological alignment within virtual groups
- Clearly differentiate programmatic and Ontological Engineering activities

Issues / Drivers

Implicit policy-making

- Engineering-driven projects are problematic for a number of reasons
- Allowing engineering team to drive the project means that very likely to get the type of ontology that the technologists are experienced in or interested in
- May result in getting the “wrong” formalization or one with poor costs and benefits

Issues / Drivers

"Big O" Ontologies vs "little o" ontologies

- Continual cognitive dissonance is not only to be expected, but is likely required for continued knowledge development and organizational growth
- Creating and maintaining alignment of project participants
- Implicit values and value system optimizations
- How to integrate multiple "operational ontologies"
- Handling ontological differences among project participants and stakeholder communities
- How to balance individual conceptualizations, social consensus, and expressiveness
- Danger of relying on informal methods for "driving" alignments
- How to drive policy making that leverages ontology-oriented solution architectures

Issues / Drivers

Resourcing and Lifecycle Issues

- Resources are scare / Perfection is expensive
- Making sure that OE is grounded in clear business objectives
- How to create solid specifications
- Choosing between waterfall and iterative development methods
- Heterogeneous vs Homogeneous roles and responsibility models
- Knowledge requirements of the development team
- Sustaining needed capabilities after a specific OE project is completed
  - The resulting ontology should be expected to continue to exist and be useful well after the initial engineer project has ended
  - Staffing and technical expertise
  - Understandings and alignment
  - Maintenance of developed ontologies

Key OM Decision Points

- What is an ontology?
- Goals, objectives, ROI?
- Project scope and requirements?
- What level of effort / investment? How much formalization?
- What tools and representation language(s)?
- Who's conceptualization is to be formalized?
- To what degree is OE effort to be driven by explicit needs vs representing an untargeted enabling capability?
- Governance mechanisms (who makes what decisions at the project and programmatic levels?)
- Quality controls
  - How is the initial quality of the ontology to be judged?
  - How is quality to be judged through time?
  - How will the significance of conceptual drift and misalignments be determined?

Defining Requirements

- Traditional requirements process likely inadequate
  - Lack needed policy development mechanisms
- Requirements process should reflect issues unique to OE
- Likelihood of dynamic solution space: Iterative learning and knowledge creation impacting understanding of problem and optimal solutions ("wicked problem")
- Need to drive consensus and social agreement in poorly understood areas

Process

- Use business process analysis to identify and articulate targeted business problems an/or opportunities
- Translate the business target into a conceptual scope (in-scope and out-of-scope concepts)
- Identify the stakeholder groups that interact with / rely upon the targeted concepts
- Use knowledge flow modeling to identify specific communication / conceptualization breakdowns of interest, especially where agent-to-agent "hand-offs" cross behavioral contexts
- Use process modeling to identify standing business rules that represent stable axiomatic components and areas of critical instability
- Use artifact modeling to define critical knowledge artifacts and their associated non-axiomatic properties that also imply meaning in the organizational context
- Judiciously apply a "relative importance" measure to prioritize analysis and modeling activities

Conclusion / Summary

- Implicit in the shift to OE-based methods of conceptual formalization is the issue that underlying conceptualizations are inherently unstable and prone to be the focus of stakeholder conflict
- Consolidating these conceptualizations in way that they can be explicitly negotiated and maintained through time should improve the alignment of IT systems with business needs (and the Business-IT relationship, generally)
- OE models need to be developed and implemented in such a way as these change dynamics are well-accounted for
- The vectors driving conceptual change need to be accounted for and understood prior to OE, both in terms of their expected timings and significance
- OM seeks to shift attention from isolated formalization models to the whole system, by addressing the conceptualization and consensus processes that must necessarily precede formalization
- The application of KFlow methods provides OM with an underlying theoretical framework that unifies and integrates the other specific methods and practices