Going Live with Enterprise Solutions

TopQuadrant™

Enterprise Semantic Web Technology Adoption

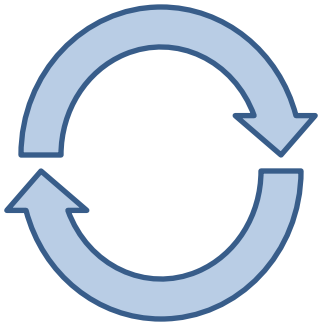
Readiness Assessment, Guidelines, and Recommendations for Getting Started

Framework for Successful Implementation, Adoption and Deployment

- Semantic web technology (SWT) adoption involves an interwoven set of *social/organizational* and *technical* considerations
- How are technical and social/organizational issues connected?
  - *Semantic solutions entail the engineering of shared knowledge structures* (ontology models) that embrace the awareness and concerns of multiple stakeholders, and the commonality and variability among their needs.
  - *Semantic technology must be adopted incrementally,* starting with a well planned initial application with a limited number of stakeholders. This is informed by experience on multiple enterprise-scale semantic initiatives as well as conventional wisdom (e.g. "Crossing the Chasm", Geoffrey Moore).
  - *A clear statement of the value proposition for each implementation increment* is a key factor for driving stages of ontology and solution development
  - *Organizational change, training and competency development issues are critical* to semantic web solution development and maintenance; it brings new terms and concepts, new roles, new ways to develop applications, new modeling and other technical skills.
- This presentation provides an overview framework of social/organizational and technical considerations for readiness assessment and getting started on semantic solution development and strategic adoption.

Areas of Consideration

- **Social/Organizational Readiness and Adoption Guidelines** – there are several factors to understand and plan for including:
  - Multiple Stakeholders will be Involved / Impacted
  - Shared Strategic Vision and Roadmap for Adoption
  - Clear Value Proposition(s) to Drive Semantic Solution Development
  - Integrated Lifecycle Approach and Method for Development
  - Training and Competency Development
  - ...



- **Technical Guidelines and Best Practices –** a number of interdependent practices must be addressed such as:
  - Naming Conventions
  - Ontology Modeling Guidelines
  - Ontology Architecture
  - Inferencing & Querying
  - Model-driven Application Architecture
  - Version Control & Governance
  - ...

Social/Organizational Readiness and Adoption Guidelines

Multiple Stakeholders will be Involved / Impacted

Why is this important?

- At the heart of semantic solutions is the engineering of shared knowledge structures (Ontologies) that reflect and address the concerns of multiple stakeholders.
  - An ontology is a run time model of information defined using constructs for Concepts – classes, Relationships – properties (object and data), Rules – axioms and constraints, and instances of concepts – individuals (data)
- Ontology Engineering must encompass integration *across contexts and stakeholders*:
  - Ontology applications involve several stakeholders.
  - Ontologies must be specific to their context to be useful, but connectable to other domains

Guidance:

- Engage with a key set of stakeholders; stakeholders may be both people and applications / systems
- Determine a value proposition that will convincingly demonstrate the approach

Shared Strategic Vision and Roadmap for Adoption - 1

Why is this important?

- Semantic solutions face not only the usual challenges of any enterprise technology project but also encompass additional requirements such as people understanding new terms, capabilities, and possibilities
- Shared understanding and vision are essential for engaging stakeholders to establish organization alignment, consistent communication and direction for adoption.
- To forge shared understand and vision, a process is needed that can:
  - Encompass early stage project planning and requirements work
  - Utilize both exploratory and convergent thinking
  - Bridge the gap between a diverse set of stakeholders and a large number of possible views and value propositions.
  - Provide a common ground and common language for business people and developers
  - Provide a rapid means to explore fully the solution space,
  - Create a framework for assessing and deciding among complex set of technology options,
  - Foster and communicate shared understanding and best practices in the way that gets people on board to using them as “catalysts” for innovation
  - Provide a traceable decision flow from business objectives to solution capabilities.

Shared Strategic Vision and Roadmap for Adoption - 2

Guidance:

- Engage stakeholders within a well-designed process to forge a shared vision for semantic technology initiatives and each specific solution project
- Inform and complement technical requirements and guidance by conducting activities that deliver key shared work products:
  - Co-develop a rich picture of the future state
  - Develop a capability architecture aligned with a roadmap
  - Agree on the acceptance strategy and plan against each capability milestone
  - Formalize this as a concept of operation and concept of execution

Clear Value Proposition(s) to Drive Semantic Solution Development

Why is this important?

- Value propositions for semantic technologies usually depend on some interchange of information among different stakeholders
- Providing value for all these stakeholders requires initiative leaders to think enterprise-wide; but adopting a new technology requires them to act locally.
- Ontology engineering methods suggest that you must understand your application before beginning a solution.
- Since semantic solutions mediate between stakeholders and applications, a broader approach is needed to understand the business situation, and what business value the solution and ontologies are to provide.

Guidance:

- Use the agreed value proposition to drive the semantic solution concept, design and scoping of ontologies
- Express a benefits model in terms of benefits, objectives and measures
- Agree on the quantitative indicators of success; embody these in acceptance test cases

Integrated lifecycle Approach and Method for Development

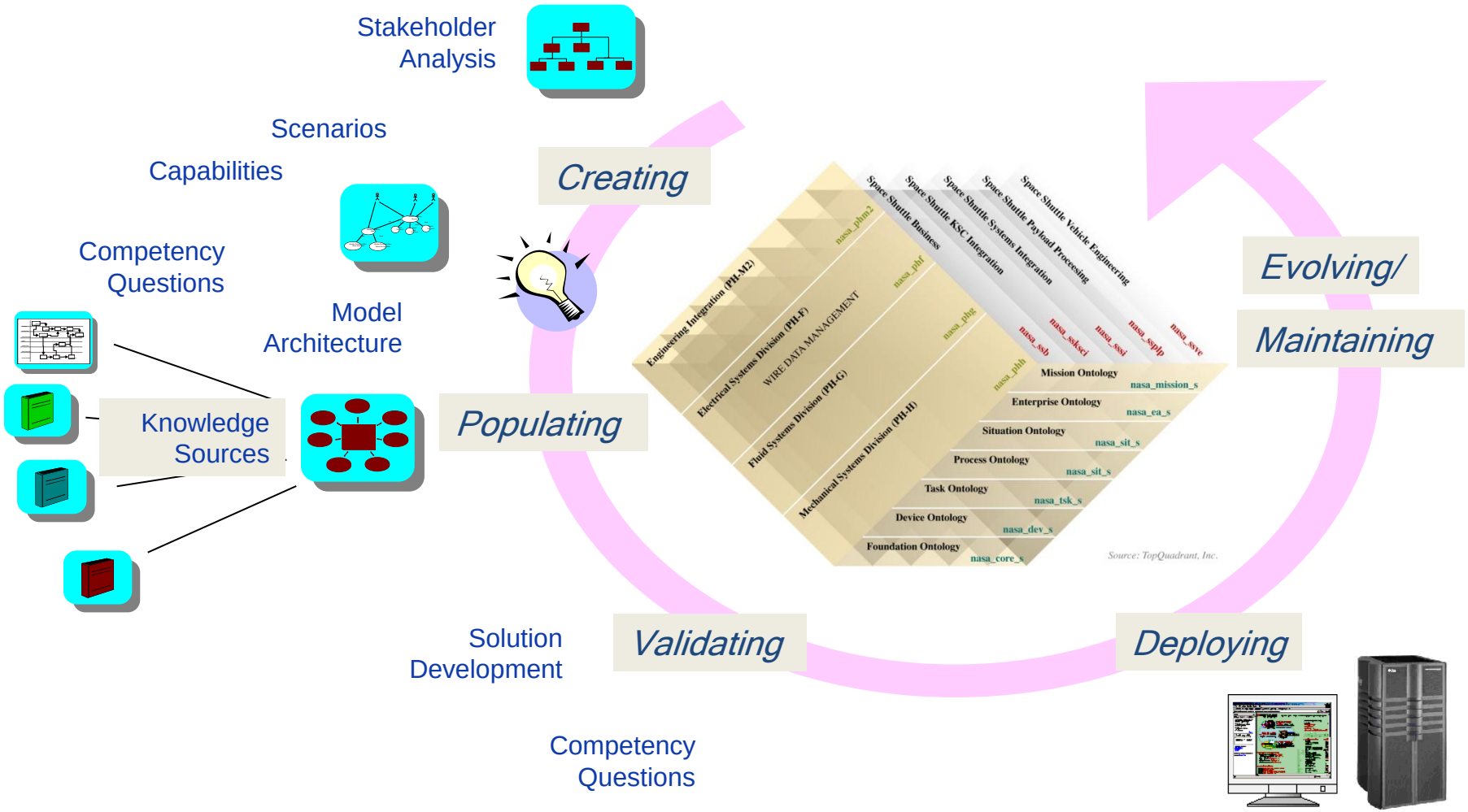
Why is this important?

- Some major challenges in developing ontology-based system solutions are:
  - identifying, positioning and sourcing all the ontologies that will play a role.
  - deciding on the boundaries of each model and dependencies between the different models
- A semantic model-based solution engineering approach must be effective for:
  - initial identification and creation of the necessary ontologies
  - taking into account the multiple stakeholders who will work with the ontology models: *understanding, using, populating, maintaining, extending, …*
  - deploying solutions that will be dynamically evolved by users continuously

Guidance:

- Adopt a strongly incremental and iterative, lifecycle approach where models are put to use early and often
- Validate modeling decisions against scenarios of model use early in the lifecycle

Semantic Solution Engineering: Elements of an Integrated Lifecycle Approach



Training and Competency Development

Why is this important?

- Semantic technology introduces new vocabulary – new concepts and terms that must be understood by individuals and shared broadly across the organization
- Development, deployment, use, maintenance and evolution of semantic solutions require new roles, new ways to develop applications, new processes and tools, new modeling methods, and other technical skills.
- Just a few of the critical issues in ontology engineering that need to be covered in trainings are:
  - Relationship of the creation/evolution of ontologies to stakeholders, domains, applications that will use the ontologies, and so on
  - Importance of re-use and approaches to ensure re-use
  - Understanding and accommodating commonality and variability across domains
  - Scoping domain models
  - Having a repeatable, explicitly defined modeling process, shared by the whole solution team

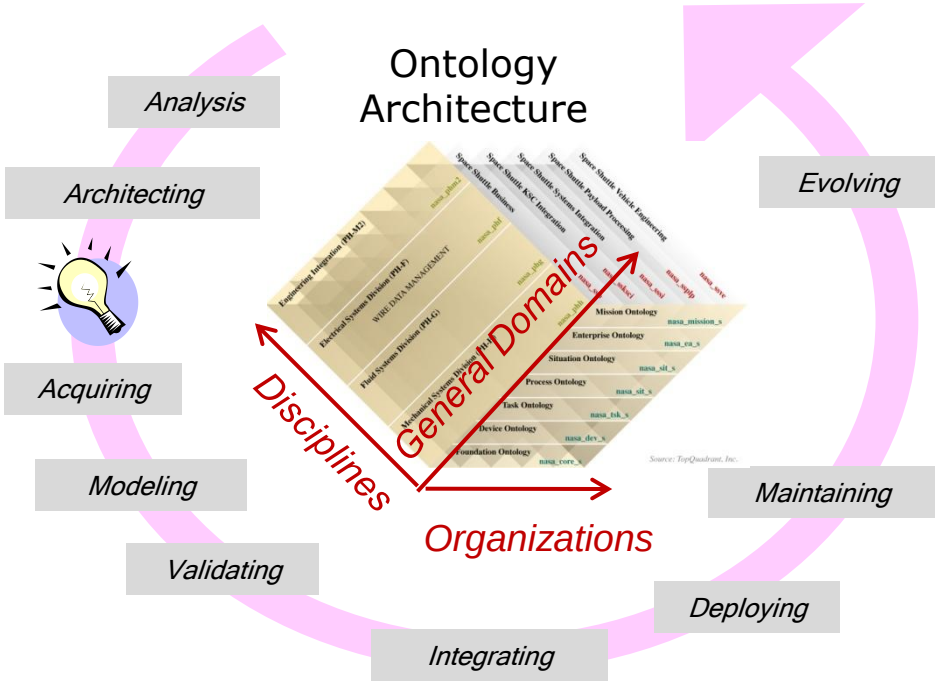
Guidance:

- Adopt a well-sequenced program of training, workshops and other types of competency development for the whole spectrum of stakeholders ranging from developers, to modelers, to users, to management, and executive sponsors.
- Match competency development approaches and content to learning styles of team members
- Establish a center of competency with an outreach strategy

Technical Guidelines and Best Practices

What have we Learned to be Key Considerations for Successful Ontology Development?

- Naming Conventions
- Modeling Guidance/Conventions
- Model Architecture
- Inferencing and Querying – Process, structure, performance
- Model-driven application architecture
- Provenance & Collaboration
- Version Control & Governance



Naming Conventions

- Why they are important:
  - Make it easier for people to understand and work with each other models
  - Avoid mistakes that are bound to happen due to misunderstandings
  - Speed up the development process by eliminating the need for each person/team make their own naming decisions

- **Guidance** examples:
  - Camel case convention for naming classes (upper) and properties (lower)
  - Class names should be singular
  - Instance names may be ‘meaningless’

Modeling Guidance/Conventions

- Why they are important:
  - Make it easier for people to understand and work with each other's models
  - Increase quality of models
  - Speed up the model development process by providing reusable patterns
  - Speed up the software development process by making it possible to assume and rely on certain patterns

- **Guidance** examples:
  - Patterns for representing information bearing relationships
  - Required practices for the use of labels and comments
  - Patterns for capturing provenance

Modeling Guidance/Conventions: Key Example Guidelines

1. Standardize: modeling patterns, concept and property names and namespaces - provide human-readable names with rdfs:label
2. Keep ontologies small and modular - evolve an ontology architecture
3. Model for reuse – separate instances from classes
4. Assimilate enterprise knowledge, for example, internal lists, vocabularies, taxonomies.
5. Be clear on the role of each ontology: specification versus knowledge discovery
6. Analyze ➔ Synthesize ➔ Evaluate: Iterate with stakeholders using blueprints. Validate models using competency questions
7. Test often using sample data
8. Be aware of the differences between open and closed world reasoning when using OWL restrictions

Model Architecture

- Why it is important:
  - Modular models are significantly more manageable and re-usable than a single monolithic data model
  - It is much easier to merge than try to later split something that should have been developed in a modular way
  - Providing basic, core concepts improves quality and consistency and decreases complexity

- **Guidance:**
  - Define a scope for each named graph (separation of schema from instances, separation of domains, etc.)
  - Adopt standards for forming URLs for each graph
  - Identify core modules reusable across the domain(s) (i.e. an upper ontology)

Inferencing and Querying Considerations - 1

- Why they are important:
  - OWL is a logical framework defined by a set of axioms
  - Depending on the intended utility of the models, some parts of the framework are more useful than others
  - Inference carries performance costs:
    - Recognizing this, W3C OWL 2.0 committee created profiles for certain usage scenarios - OWL EL, OWL RL, OWL QL
    - These profiles represent particular trade-off points in the continuum between expressive power and performance
    - While all semantic web software supports RDF storage and query of RDFS/OWL axioms, support for inferencing ranges from none to some subset of a profile to the entire OWL system and beyond
  - Modeling decisions will be impacted by the minimum expected capability of the platform and the performance requirements
- **Guidance:**
  - Balance the value of using certain forms or combinations of axioms against the expected performance trade-offs
  - Make the technical constraints behind the modeling practices explicit

Inferencing and Querying Considerations – 2

- Practical engineering considerations typically require some use of domain specific rules in addition to OWL axioms
- Options for rules – and respective commentary – include:
  - Implementing rules directly in the software code (Java, JavaScript, Ruby, etc.) – brittle, error prone, some definitions are in RDF models, others are in code
  - Rule Interchange Format (RIF) – a new standard from W3C intended primarily for the interchange of rules, currently has very limited adoption and platform support, integration with RDF is immature
  - SPARQL (RDF query language) – has a very wide adoption, can be used to express rules and to store them with the models
- Example: min-cardinality rule (will not generate constraint violation in OWL – open world)
  - 'An Information Exchange must have at least one Data Object'
- **Guidance:** Use an approach to rules that is sufficiently expressive but works with RDF and preserves the W3C semantic-web standards-based approach for flexibility and 'future proofing'

Model-driven application architecture

- Why it is important:
  - In most areas of functionality, responsibility can be divided between:
    - Models
    - Rules
    - Code/scripts
  - To be effective, model development must be informed by the role of the models in the application architecture
  - Changes in the application functionality can be achieved through changes to any of the three components above

- **Guidance:** Consider performance, scalability, maintainability, reuse, future-proof-ness and portability requirements to determine the optimal division of responsibility

Provenance and Collaboration

- Why they are important:
  - It is typical in the projects using Semantic Web standards to use information from multiple sources
  - Having provenance for each fact is key to the ability to trust the information, to understanding the information's context and who to work with should more clarification and/or changes be required

- **Guidance:**
  - Develop modeling patterns for capturing provenance
  - Establish a collaboration framework for parties to work together
  - Automate capture of provenance information

Version Control and Governance

- Why they are important:
  - Information, including models, evolves over time
  - Organizations want to make evolution inexpensive and seamless. At the same time, they want to be able to control and govern the evolution
  - The usefulness of information is sensitive to many factors, such as timeliness and scope, for instance consider questions such as:
    - Is a fact (collection of facts) always true or is true only during a certain period of time?
    - When did a fact (collection of facts) first became known?
- **Guidance:**
  - Provide the capability to trace the nature and cause of the change
  - Automate processes for publishing new versions including, as appropriate, approval and review cycles
  - Make it possible to distinguish between and query for current and historic information

Conclusions:

Semantic Web Technology Adoption and Getting Started

- This presentation highlights representative categories of important topics that need to be considered/addressed
- An exhaustive list can't be presented without considering:
  - Organization specific topics / constraints
  - Specific issues dependent on what is envisioned to be done
- To successfully address the topics covered and those not yet uncovered/articulated, we recommend:
  - Doing an initial, brief but intensive, planning and envisioning project
  - Using specific processes, methods, and best practices
    - based on the experiences of organizations adopting large scale semantic solutions
    - proven to iteratively address a complex set of social/organizational and technical considerations

Jumpstart Packages

- TopQuadrant offers special jumpstart solution packages to help you and your organization get started with successful semantic solutions and adoption. These packages are carefully designed to:
  - help customers quickly achieve their goals for a reasonable price
  - respond to the experiences, needs and requests of customers
  - address the interwoven set of *social/organizational* and *technical* challenges outlined
  - employ specific processes, methods, and best practices based on the guidelines presented
- Jumpstart packages combine just the right level of targeted training, solution envisioning, expert services for design/implementation, and software tools, custom tailored as needed from our complete set of offerings:
  - Comprehensive Training Program
  - Solution Envisioning and Planning
  - Consulting and Solution Development Services
  - Lifecycle Method for Ontology Engineering and for Building Semantic Model-driven Solutions
- We will work with you to tailor the right jumpstart package to your needs and budget to assure your path to a successful solution, value and ROI.
- To inquire about Jumpstart Packages , contact us at sales@topquadrant.com