Annotating Decision Analyses using Semantic Web Technologies
Thesis Submitted to the Faculty
of
College of Computer and Information Science
Northeastern University
in partial fulfillment of the requirements for the
Masters of Computer Science
# Table of Contents

1. Acknowledgements................................................................................................................. 3
2. Abstract........................................................................................................................................ 4
3. Introduction................................................................................................................................. 5
   3.1. Motivation.............................................................................................................................. 5
   3.2. Statement............................................................................................................................... 5
   3.3. Problem.................................................................................................................................. 6
   3.4. Use Cases.............................................................................................................................. 6
      a) Development Process........................................................................................................... 6
      b) Business Process ................................................................................................................... 6
      c) Documentation....................................................................................................................... 7
      d) Interoperability & Reuse......................................................................................................... 7
4. Background................................................................................................................................. 8
   4.1. Terminology.......................................................................................................................... 8
   4.2. Semantics............................................................................................................................... 8
   4.3. Ontology................................................................................................................................ 8
   4.4. OWL: Web Ontology Language............................................................................................. 9
   4.5. Eclipse Process Framework.................................................................................................... 10
   4.6. Influence Diagram.................................................................................................................. 10
   4.7. Decision Analysis................................................................................................................... 12
5. Related work............................................................................................................................... 12
   5.1. Literature............................................................................................................................... 12
6. Outline.......................................................................................................................................... 12
7. Building Ontology....................................................................................................................... 13
   7.1. Protege................................................................................................................................... 13
   7.2. Cmap for diagrams............................................................................................................... 13
   7.3. Part of our Ontology as a Cmap diagram.............................................................................. 14
   7.4. Methodology........................................................................................................................ 16
   7.5. Reasoners in Protege 4.......................................................................................................... 19
   7.6. Case Study............................................................................................................................ 19
8. Conclusion................................................................................................................................. 25
9. Future Work............................................................................................................................... 25
10. Bibliography/Reference............................................................................................................ 26
11. Appendices............................................................................................................................... 26
   11.1. Entire Ontology.................................................................................................................... 26
1. Acknowledgements

I would like to deeply thank my advisor Prof. Kenneth Baclawski for all his support and guidance during the last two years. He always had an ear for the queries I had even outside academics. He introduced me to the Semantic Web community which I didn't know existed.

I would also like to thank Prof. Mitch Kokar for his suggestions and input during my thesis.

I want to thank my dad Vijay Duggar, my mom Sadhana Duggar and my sister Vibhuti Mutha for their undying support and encouragement all my life.

Special thanks to Shreya for making life bearable away from home.
2. Abstract

In this thesis we use Semantic Web techniques for the formal representation of the decision analysis and the Eclipse Process Framework (EPF) to coalesce decision making process with a software engineering life cycle. In this way, a formal artifact can be produced that expresses how and why a decision was made, which can be linked into the many other artifacts of the development process. Semantic Web search tools may then be employed to find similar decisions so that they may be reused. Semantic technology can have a significant impact on retrieval, reuse, interoperability, consistency, enforcement of policy and validation of decision analyses. We have developed an OWL ontology for the basic decision process that we have represented using EPF. We integrated the process ontology with our OWL decision analysis annotation ontology. Finally, we have used the resulting ontology to annotate a sample of decisions and the processes by which they were made.
3. Introduction

3.1. Motivation

It took me a while to decide what to write in this section of my thesis! But here we are concerned about decisions where a lot is at stake. An example would be the decision to choose which Software development life-cycle to follow for a project.

How are decisions taken in a process model? Certainly not by rolling dice! To make such decisions first the issue is thoroughly analyzed. There are a lot of meetings, discussing the alternatives and the factors that should be considered. And then senior management comes to some conclusion based on that.

The decision is made based on how, when, why and what were the other alternatives, what were the arguments made and what were the criteria. The analysis is documented, but what happens to it?

Modern development processes require a team to make a series of decisions. This is especially true for software development. While sophisticated decision support tools are available, the connection between the decision and the development process is informal if there is a connection at all.

Consequently, it can be difficult to revisit or reuse a decision analysis at a later time or for another purpose. Once a decision has been made, the analysis that was performed is often discarded, and even if it is retained, the context within which the analysis was performed is not documented sufficiently for the analysis to be easily revisited when the situation changes or to be reused for similar decisions.

On a more practical level, the lack of a formal connection to the process in these systems limits their scalability, tailorable and adaptability, and requires that every new decision be developed from scratch, even if it is only a slight variant of a previous decision.

3.2. Statement

Semantic technology can have a significant impact on retrieval, reuse and validation of decision analyses. We developed an OWL ontology for the basic decision process that we have represented using EPF. We integrated the process ontology with our OWL decision analysis annotation ontology. Finally, we used the resulting ontology to annotate a sample of decisions and the processes by which
they were made.

3.3. **Problem**

To determine the extent to which semantic technologies add value for search, reuse and validation on the sample developed.

3.4. **Use Cases**

a) **Development Process**

Numerous decisions have to be taken during any development process. Regardless of the structure of the management or people who are involved in that decision, it always has a rationale. With a large group of people the process tends to get complicated because of the number of options suggested by them. This by all means is good for that decision. But if no record is kept of how or why a person or a team arrived at that decision then it would be a tedious job to rediscover those reasons or other possible alternatives later during the development process. Thus having a rationale would help in development process by eliminating the rediscovery of a decision completely.

A simple example: During the design cycle of a development process of a hardware product the length of a cylindrical shaft was decided to be 123 centimeters because anything below that would damage the hardware and would be unsafe. Later, during the end of the design cycle a team member suggested that if we keep the length to 110 centimeters we could reduce the cost by 10%. If they kept no record of why they earlier had made the decision for 123 centimeters they would have to rediscover that either by again having to think over it or by taking the wrong decision and then seeing the result at the end after manufacturing it.

b) **Business Process**

Analogous to the development process, every business process is designed based on various factors like cost, profit, quality, customer, product or service market, etc. Even during the business process there are infinite occasions were critical decisions are to be made and if no record of the decisions or the reasons for those decisions is kept it would be a huge overhead to revisit them.

In any process, but more importantly in business, there are many alternatives and if the chosen alternative does not yield the intended results that decision has to be revisited. If a record of the
decision and the alternatives visited during that decision is present, the problem at hand could be fixed or eliminated in fraction of the amount of time that would have been spent if one had to rediscover them.

For example: 4 months back a supplier X was chosen among 9 other suppliers because the cost from him was the lowest, even when one of the suppliers Y offered free shipping and handling. Today due to some reason the supplier increases his price but it is still lower than others. But it would be cheaper for you to buy it from the supplier who was offering free shipping. It would be a straight forward decision if you did not have to rediscover the alternatives and had a rationale for that decision.

Another important business process issue is policy enforcement. Companies have policies that must be followed (including some that are required by law). Formalization is a much more reliable (and much less expensive) way to ensure that policies are enforced than informal checking by lawyers.

c) Documentation
All design and development processes in the engineering industry as well as academia have documentation at almost every stage of the whole process. In particular software engineering is heavily dependent on documentation. The documentation of all decisions made during the process would ease the ability to take other alternatives when the chosen alternative is found to be wrong. Or if the management/team wants to know why a particular decision was taken. With every decision documented in the form of a rationale it would be a lot easier to do that. The amount of time saved, is a critical factor in a successful project.

For example: There are scenarios where depending on the decision made the management or team would opt for a branch, out of one or more possible directions. Later during the development life cycle, the team could well be at a situation where the opted branch is found not to be fruitful or in fact putting the project itself at risk. The option then would be to go back to where you branched out. And if a rationale was maintained it would be easier and a lot quicker to do that.

d) Interoperability & Reuse
In almost all processes if a part of the work product or the whole product itself is reused it saves time and effort. And the most time consuming part when a group of people are involved is making decisions. If a collection of all the decisions made in the past is kept, it could be reused. A rationale could be
reused in a similar context in future or in a different context as well as by extracting what is useful to that context.

For example: Every law practicing firm does this. They have a collection of decisions i.e. decisions made by different or same people in the past and use it according to their need in a similar or different context. Even with the current system in place it would be helpful to have a rationale of why a particular suit or example was cited and what the other options were that could have been cited. That would be valuable data the firm could use in future.

4. Background

4.1. Terminology
Decision analysis, Decision Support, Semantic Web, Rules, Inference, EPF

4.2. Semantics
A concept could be described in more than one way depending on the context for which it is described. When two concepts have the same meaning they are said to be semantically equal. The usual method for defining the sameness relationship is to use mathematics. Terminology and statements are then mapped to an abstract mathematical structure, usually called the model. Two terms or statements are the same when they map to the same model[2].

Formal semantics is used to provide precise meaning of knowledge in a particular area. “Precisely” here means that the semantics does not refer to subjective intuitions, nor is it open to different interpretations by different persons (or machines). The importance of formal semantics is well-established in the domain of mathematical logic, among others.[3]

4.3. Ontology
One of our goals is to document decision analysis so that it can be easily reused both by humans and by software agents. This is done with a formal representation of terminology called an ontology. An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usuable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge
that spans domains. In this way, they make that knowledge reusable. Semantic Web ontology tools can perform automated reasoning using the ontologies, and thus provide advanced services to intelligent applications such as: conceptual/semantic search and retrieval, software agents, decision support, speech and natural language understanding, knowledge management, intelligent databases, and electronic commerce. [4][2]

4.4. OWL: Web Ontology Language

Web Ontology Language or OWL was defined by W3C. It is now a standardized and widely accepted ontology language.

OWL has three different sublanguages each of which is intended to address specific communities of implementers and users.[5]

• **OWL Full**: Its constituent is a set of all the OWL language primitives. It also allows one to combine these primitives in arbitrary ways with RDF and RDF Schema. The advantage of OWL Full is that it is fully upward compatible with RDF, both syntactically and semantically: any legal RDF document is also a legal OWL Full document, and any valid RDF/RDF Schema conclusion is also a valid OWL Full conclusion. Hence users have maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right[3]

• **OWL DL**: To attain computational efficiency, OWL DL (short for: Description Logic) was created as a subset of OWL Full which restricts the way in which the constructors from OWL and RDF can be used, ensuring that the language corresponds to a well studied description logic. The advantage of this is that it permits efficient reasoning support. The disadvantage is that we loose full compatibility with RDF: An RDF document will in general have to be extended in some ways and restricted in others before it is a legal OWL DL document. Conversely, every legal OWL DL document is still a legal RDF document. [3]

• **OWL Lite** is a further subset of OWL DL and supports a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than the other two.[5]
4.5. Eclipse Process Framework

The process model we use is expressed using the Eclipse Process Framework Composer. EPF is a tool for producing a customizable software process engineering framework, with exemplary process content and tools, supporting a broad variety of project types and development styles[6]. The extensible process frameworks bundled with EPF Composer are OpenUP/Basic, OpenUP/MDD, and recently added “other agile components.” We are not going to extend any of those but create a new process model using EPF Composer. EPF was derived from RUP or Rational Unified Process. A subset of RUP that was released by IBM to the open source community is known as OpenUP. OpenUP is a minimally sufficient software development process - meaning that only fundamental content is included. Thus, it does not provide guidance on many topics that projects may deal with, such as large team sizes, compliance, contractual situations, safety or mission critical applications, technology-specific guidance, etc. RUP/OpenUP is an iterative software development process framework. It is not a step by step process cycle but an adaptable process framework which allows an organization or a development team to put together a process model according to their needs. Such a framework was and is needed desperately as no single software development life cycle is suitable for all the software development. What EPF or RUP essentially do is, provide us with a tool through which we can define the method content separately. Method Content as the name suggests is the description of goals, required skills, various artifacts and deliverables etc. Without worrying about where and when they are used in the process life cycle. After defining the method content the processes are defined using the method content. The processes describe the sequence in which the work is performed by different roles and the deliverables that would be the delivered during the course of the process life cycle. The important thing to note here is that the work, roles and deliverables are all defined in the method content. The method content and processes can be re-used as they are. They can also be configured according to the need of a project and then re-used. They can be extended if there seems to be a need for it. Or they can be modified a little to meet the requirements. Separation of the definition of the method content and processes enables one to re-use, extend or tailor them independently.

4.6. Influence Diagram

Uncertainty is an inevitable part of any decision making process. There is uncertainty involved in the decisions where the circumstances are evolving. For example, a client could change the requirement a little bit; which happens all the time. Then every related decision needs to be revisited. One cannot eliminate uncertainties involved in decision making completely as there is always some randomness involved. But we might change our mind as the uncertainties reduce and hence would need to change the analysis.
Influence diagrams are a systematic, empirical technique for representing uncertainty in a decision. The formal basis for influence diagrams is probability theory, and they are represented using Bayesian networks. A Bayesian network (BN) is a graphical mechanism for specifying the joint probability distribution of a set of random variables[7]. As such, BNs are a fundamental probabilistic representation mechanism for stochastic models. The use of graphs provides an intuitive and visually appealing interface whereby humans can express complex stochastic models. This graphical structure has other consequences. It is the basis for an interchange format for stochastic models, and it can be used in the design of efficient algorithms for data mining, learning, and inference. The range of potential applicability of BNs is large, and their popularity has been growing rapidly. BNs have been especially popular in biomedical applications where they have been used for diagnosing diseases[8], among many other applications. The random variables of the Bayesian network are represented as nodes of a graph. The edges denote dependencies between the random variables. This is done by specifying a conditional probability distribution for each node. It is also required that the edges of a BN never form a directed cycle: a BN is acyclic. If two nodes are not linked by an edge, then they are conditionally independent. One can view this independence property as defined by (or a consequence of) the following property of a BN: The JPD of the nodes of a BN is the product of the CPDs of the nodes of the BN. This property is also known as the chain rule of probability. This is the reason why the BN was assumed to be acyclic: the chain rule of probability cannot be applied when there is a cycle. When the BN is acyclic one can order the CPDs in such a way that the definitions of conditional probability and statistical independence can be applied to get a series of cancellations, such that only the JPD remains.

“An influence diagram is a way of describing the dependencies among aleatory variables and decisions. It can be used to visualize the probabilistic dependencies in a decision analysis and to specify the states of information for which independencies can be assumed to exist.”[9] It is derived from Bayesian Networks, but has a very high degree of annotation. This is a key requirement for understanding and re-using a decision analysis. An influence diagram is a network comprising of directed graph which has three types of nodes. It has at most one value utility or value node that represents the quantity that is to be maximized. It also has zero or more chance nodes representing random variables. And lastly, it may have zero or more decision nodes, which represent the alternatives available to the decision maker.[10] The diagram as a whole encodes the criteria. Evaluating the diagram for each alternative to get the utilities for comparison to reach a decision would be the argument. There are various algorithms that have been proposed for evaluating influence diagrams. Schachter developed an algorithm that can evaluate any well formed influence diagram and determine the optimal policy for its decision.
4.7. Decision Analysis

Decision Analysis comprises strategies, methods and tools needed to identify, clearly represent, and formally assess the important aspects of a decision situation. For prescribing the recommended course of action by applying the maximum expected utility action axiom to a well-formed representation of the decision, and for translating the formal representation of a decision and its corresponding recommendation into insight for the decision maker and other stakeholders.[reference needed]

Design rationales are frequently documented by Software Architects but the reasons about why a design is chosen and why it was considered better than other alternative designs are usually not captured.[11]

If we have taken a decision in the past, i.e., spent time going over the alternatives, their advantages and their disadvantages, then we need to make sure that we use it in future if a similar decision comes our way and not waste time and money redoing the work from scratch. Perfectly good ways of making decisions are discarded, and others with a poor track record continue to be used[12].

5. Related work

5.1. Literature

A lot of work has been done on design rationale. See Moran [13] or Lee [14] for an overview. However our focus is on generic decision support using Semantic Web technologies. Our approach is close to Kruchen[15] or Burge[16]. Krutchen tries to describe an ontology for decision rationale as a whole and not for the constituents of a decision. His description is an ontology of architectural design decisions, their attributes and relationships, for complex, software-intensive systems. [15] Burge on the other hand has developed a system called SEURAT (Software Engineering Using RATionale) for which he defines an Argument Ontology. Our approach is to annotate not only the decision analysis as a whole but also annotate its constituents. We provide an implementation of the ontology in OWL.

6. Outline

We started with finding out the pains a user had when dealing with decision analysis. Documenting decision analyses is often not defined in the process model of a development life cycle. We then created a EPF representation of the decision analysis process model such that it could be easily integrated with current development life cycle.
Once having a clear understanding of the process model we went on to design the decision ontology with which one could annotate a decision analysis.

7. Building Ontology

7.1. Protege

Protege was developed at Stanford Medical Informatics. Protege has been heavily used for creating and working with ontologies in the biomedical domain. It allows one to store the ontologies in CLIPS, RDF, XML, UML formats and now in OWL too.

OWL as described earlier is what we use to implement our ontology. The syntax for OWL is complex and wordy. The implementation could be simplified a lot if a tool like Protege is used. It not only provides the graphical interface to design the ontology but also generates forms to create individuals (instances). Protege has 8 tabs for navigating the ontology. The tab Active Ontologies is used to work with Ontology Annotation, Ontology Metrics, Imported Ontologies, General Axioms, RDF/XML Rendering, OWL/XML Rendering and OWL Functional Syntax Rendering. The entities tab is used to work with all the entities like classes, object properties, data properties and individuals. The second half of this tab changes depending on which type of entity the user has selected. And they also have separate tabs for these 4 types.

7.2. Cmap for diagrams

CmapTools is a software environment developed at the Institute for Human and Machine Cognition (IHMC) that allows one to represent knowledge models (in our case our ontology) using concept maps [17].
7.3. Part of our Ontology as a Cmap diagram
ProperRationale

is defined as all of

hasDecision can be
hasAlternatives can be
hasArguments\text{ must be }
hasIssue can be
hasCriteria can be

Decision Alternatives ProperArguments Issue Criteria

are same class as

Artifact Things which

hasArgument can be

ProperArgument

same class as are

Things which Argument

basedOnCriterion can be

Criterion
7.4. **Methodology**

We now discuss the methods we will be using. Starting with ontology development as the basis, we then develop decision analysis process models.

The description of a decision analysis, including the reasons underlying a decision, will be called a rationale. The main constituents of a rationale are:

1. Issue
2. Analysis
   a) Background
   b) Assumptions
   c) Argument
3. Criteria
4. Alternative
5. Decision/Suggestion

The issue is the matter at hand that has to be decided. The analysis includes all relevant background information and their relationships to one another. When stated informally, the analysis consists of the statements made based on one of or all criteria, to support or oppose one or more alternatives. When stated formally, the analysis consists of a diagram or table that expresses to what extent the criteria support the alternatives. Criteria are the requirements that are necessary for selecting a particular choice among all the alternatives. Alternatives are the choices that could be made to make a decision. A formal decision analysis may be expressed using influence diagrams or other mechanisms such as decision trees or tables. Decision/Suggestion is a description of the final decision or suggestion made. These constituents are used later to define the process model.
The process model in essence implements a basic decision making sequence of processes but it extends it by specifying a model to document it. The use case diagram showing the actors and activities during formal documentation of decision analysis is:

The two roles/actors are the developer of the decision analysis documentation and the user of the decision analysis. The user is the actual decision maker who works with the developer to ensure that the decision analysis is properly integrated with the development process. The developer can perform a number of actions on the repository of rationales, such as create, modify and reuse/repurpose. In collaboration with the rationale user, the developer helps to find relevant rationales. There are many other use cases that we have not developed that are concerned with activities such as reconsidering decisions and inference.
The process involves a number of steps and iterations as follows:
1. Gather background information available for the issue at hand
2. Enumerate all the assumptions that are relevant and can be inferred based on the context or situation
3. Exhaustive list of all the alternatives that can be chosen for a particular decision have to be documented.
4. Similarly, a list of criteria based on which any alternative would be chosen for an issue is documented.
5. Both step 1 and step 2 are done iteratively until a satisfactory list of both alternatives and criteria are
available.
6. Relevant arguments for each alternative based on the list of criteria are obtained.
7. Based on the arguments put forward a decision is recommended.
8. The whole process from steps 1 to 6 could then be iterated till a satisfactory decision is obtained.

7.5. **Reasoners in Protege 4**

Protege comes with two reasoners Pellet and Fact++. Pellet is the first complete OWL-DL reasoner with extensive support for reasoning with individuals (including nominal support and conjunctive query), user-defined datatypes, and debugging support for ontologies. It implements several extensions to OWL-DL including a combination formalism for OWL-DL ontologies, a non monotonic operator, and preliminary support for OWL/Rule hybrid reasoning.[18]

7.6. **Case Study**

The following case study was taken from Clemen 1991[19]. This is the decision analysis of Democratic Presidential nominee choosing a running mate for the 1988 U.S Presidential elections.
ArgumentForCandidates

hasArgument

Home_State

hasArgument

Arguments

is a

Bentsen_Ties

basedOnCriterion

AbleToWinState

is a

Criterion

Argument

is a

Criterion

IsFundRaiser

Based on the diagram, the structure can be described as:

- ArgumentForCandidates
  - hasArgument
  - Home_State
    - basedOnCriterion
    - AbleToWinState
      - is a
      - Criterion
  - is a
  - Arguments
    - is a
    - Argument
      - is a
      - Criterion
  - is a
  - Bentsen_Ties
    - basedOnCriterion
    - IsFundRaiser
We annotated this analysis using our ontology and performed reasoning using Pellet. The reasoner infers that an instance belongs to a ProperRationale class if it satisfies all the restrictions mentioned in the class ProperRationale. All inferred objects are displayed as yellow rectangles.
Then we remove the artifact Criteria from the rationale and run the reasoner again. Hence, the reasoner does not infer this as a ProperRationale. For an individual to be a ProperArgument it has to be based on a Criterion. If we have an argument that is not based on a Criteria the reasoner does not infer it to be a ProperArgument.

Bentsen’s ties with big business were unusual for a Democratic nominee. Bentsen was one of the best fund raisers around and might be able to eliminate or even reverse the Republicans’ traditional financial advantage. Even if some of the more liberal voters were disenchanted, Bentsen could appeal to a more business-oriented constituency.
But when you change it to be based on a criterion it is then inferred to be a ProperArgument.

Thus, using the decision ontology you would be able to validate your rationale. Inferences like these would add a lot of value to mission critical decisions that need to be valid. Annotating the rationale would also make it easier to search.

And when the constituents and the rationale as a whole are searchable it could be reused more often. For example you could search for rationales related to an individual and gather some background information.
8. Conclusion

The following conclusions are drawn from this thesis:

1. The decision ontology broadly captures a decision rationale and the process of making a decision.

2. By annotating a case study with this ontology it is possible to infer about the rationale and perform validation.

3. The annotation also helps in searching a rationale or a constituent of a rationale and in turn is useful in reusing a rationale.

9. Future Work

There is a lot of room for improvement to the decision ontology implemented in this thesis. Although it covers some ground at annotating a rationale, it could be enhanced to incorporate many more validations.

Also it would be nice to have a tool that generates a wizard for this ontology so that it is easier for a user to annotate his rationales.
10. Bibliography/Reference

Bibliography

[16] Janet E. Burge, Software Engineering Using design RATionale,

11. Appendices

11.1. Entire Ontology

<?xml version="1.0"?>
<owl:ObjectProperty rdf:about="input">
  <rdfs:isDefinedBy>
    An artifact that is fed as an input to a particular
<owl:ObjectProperty rdf:about="output">
  <rdfs:domain rdf:resource="#Step"/>
  <rdfs:range rdf:resource="#Artifact"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#steps -->
<owl:ObjectProperty rdf:about="steps">
  <rdfs:domain rdf:resource="#ProcessModel"/>
</owl:ObjectProperty>

<owl:FunctionalProperty rdf:about="#basedOnCriterion">
  <rdfs:domain rdf:resource="#Argument"/>
  <rdfs:range rdf:resource="#Criterion"/>
</owl:FunctionalProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAlternative -->
<owl:ObjectProperty rdf:about="#hasAlternative">
  <rdfs:range rdf:resource="#Alternative"/>
  <rdfs:domain rdf:resource="#Alternatives"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAlternatives -->
<owl:FunctionalProperty rdf:about="#hasAlternatives">
  <rdfs:range rdf:resource="#Alternatives"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasArgument -->
<owl:ObjectProperty rdf:about="#hasArgument">
  <rdfs:range rdf:resource="#Argument"/>
  <rdfs:domain rdf:resource="#Arguments"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasProperArguments -->
<owl:ObjectProperty rdf:about="#hasProperArguments">
  <rdfs:range rdf:resource="#ProperArguments"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:ObjectProperty> -->

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasArguments -->
<owl:FunctionalProperty rdf:about="#hasArguments">
  <rdfs:range rdf:resource="#Arguments"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAssumption -->
<owl:ObjectProperty rdf:about="#hasAssumption">
  <rdfs:range rdf:resource="#Assumption"/>
  <rdfs:domain rdf:resource="#Assumptions"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAssumptions -->
<owl:FunctionalProperty rdf:about="#hasAssumptions">
  <rdfs:range rdf:resource="#Assumptions"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasBackground -->
<owl:ObjectProperty rdf:about="#hasBackground">
  <rdfs:range rdf:resource="#Background"/>
  <rdfs:domain rdf:resource="#BackgroundInfo"/>
An alternative is one of the alternatives that could be recommended as decision.</owl:isDefinedBy>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Alternatives -->
<owl:Class rdf:about="#Alternatives">
  <rdfs:subClassOf rdf:resource="#Artifact"/>
  <rdfs:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAlternative"/>
      <owl:someValuesFrom rdf:resource="#Alternatives"/>
    </owl:Restriction>
  </rdfs:equivalentClass>
  <rdfs:isDefinedBy>
    >All constituent alternatives grouped together.<rdfs:isDefinedBy>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#AnnotatingDecisionAnalysis -->
<owl:Class rdf:about="#AnnotatingDecisionAnalysis">
  <rdfs:subClassOf rdf:resource="#ProcessModel"/>
  <rdfs:comment>
    >A context for processes for annotating a decision analysis.<rdfs:comment>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Argument -->
<owl:Class rdf:about="#Argument">
  <rdfs:subClassOf rdf:resource="#Constituent"/>
  <rdfs:isDefinedBy>
    >An argument is one of the arguments made based on one or more criteria for or against an alternative.<rdfs:isDefinedBy>
</owl:Class>

<owl:Class rdf:about="#ProperArgument">
  <rdfs:subClassOf rdf:resource="#Argument"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#basedOnCriterion"/>
      <owl:someValuesFrom rdf:resource="#Criterion"/>
    </owl:Restriction>
  </owl:equivalentClass>
  <rdfs:isDefinedBy>
An argument is one of the arguments made based on one or more criteria for or against an alternative.

All constituent arguments grouped together.

All constituent arguments grouped together.

Document the arguments that were made for an alternative.

All constituent arguments grouped together.

All constituent arguments grouped together.
An assumption is one of the assumptions made during this process. <rdfs:isDefinedBy>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Assumptions -->
<owl:Class rdf:about="#Assumptions">
  <rdfs:subClassOf rdf:resource="#Artifact"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAssumption"/>
      <owl:someValuesFrom rdf:resource="#Assumption"/>
    </owl:Restriction>
    <owl:equivalentClass>
      <rdfs:isDefinedBy>
        >All constituent assumptions grouped together.<rdfs:isDefinedBy>
      </owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Background -->
<owl:Class rdf:about="#Background">
  <rdfs:subClassOf rdf:resource="#Constituent"/>
  <rdfs:isDefinedBy>
    >Background information on a particular topic related to the issue.<rdfs:isDefinedBy>
  </owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#BackgroundInfo -->
<owl:Class rdf:about="#BackgroundInfo">
  <rdfs:label rdf:datatype="&xsd;string">
    Background Knowledge</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Artifact"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasBackground"/>
      <owl:someValuesFrom rdf:resource="#Background"/>
    </owl:Restriction>
    <owl:equivalentClass>
      <rdfs:isDefinedBy>
        >Background information on various topics grouped together.<rdfs:isDefinedBy>
      </owl:Class>
<owl:Class rdf:about="#Constituent" >
    <rdfs:comment>
        A Constituent is a constituent of an Artifact.</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="#Criteria" >
    <rdfs:label>Criteria</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Artifact"/>
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasCriterion"/>
            <owl:someValuesFrom rdf:resource="#Criterion"/>
        </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:isDefinedBy>
        >All the constituent criteria grouped together.</rdfs:isDefinedBy>
</owl:Class>

<owl:Class rdf:about="#Criterion" >
    <rdfs:subClassOf rdf:resource="#Constituent"/>
    <rdfs:isDefinedBy>
        >A criterion is one of the criteria based on which arguments are made
        for an alternative.</rdfs:isDefinedBy>
</owl:Class>

<owl:Class rdf:about="#Decision" >
    <rdfs:subClassOf rdf:resource="#Alternative"/>
    <rdfs:subClassOf rdf:resource="#Artifact"/>
    <rdfs:isDefinedBy>
        >This could either be a decision that was taken or a decision
        recommended</rdfs:isDefinedBy>
</owl:Class>

<owl:Class rdf:about="#DecisionAnalysis" >
<rdfs:subClassOf rdf:resource="#Artifact"/>
<rdfs:isDefinedBy>
  >Analysis of making a particular decision.</rdfs:isDefinedBy>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#DecisionRecommended -->

<owl:Class rdf:about="#DecisionRecommended">
  <rdfs:label rdf:datatype="&xsd;string">
    Recommended Decision</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Decision"/>
  <rdfs:isDefinedBy>
    >A decision that was recommended based on the decision analysis.</rdfs:isDefinedBy>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#DecisionTaken -->

<owl:Class rdf:about="#DecisionTaken">
  <rdfs:subClassOf rdf:resource="#Decision"/>
  <rdfs:isDefinedBy>
    >The decision that was taken after considering all the information in the decision analysis.</rdfs:isDefinedBy>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#EnumerateAssumptions -->

<owl:Class rdf:about="#EnumerateAssumptions">
  <rdfs:label rdf:datatype="&xsd;string">
    Enumerate Assumptions</rdfs:label>
  <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#GatherBackgroundInfo -->

<owl:Class rdf:about="#GatherBackgroundInfo">
  <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
  <rdfs:comment>
    >Collect all the background information related to the issue that could help in making the decision.</rdfs:comment>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#InformalDiscussion -->
<owl:Class rdf:about="#InformalDiscussion">
  <rdfs:subClassOf rdf:resource="#DecisionAnalysis"/>
  <rdfs:isDefinedBy>
    Analysis document in an ad-hoc format.</rdfs:isDefinedBy>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Issue -->
<owl:Class rdf:about="#Issue">
  <rdfs:label>Issues</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Artifact"/>
  <rdfs:isDefinedBy>
    A problem for which a decision needs to be taken.</rdfs:isDefinedBy>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ListAlternatives -->
<owl:Class rdf:about="#ListAlternatives">
  <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
  <rdfs:comment>
    Possible alternatives that could be recommended as a decision.</rdfs:comment>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ListCriteria -->
<owl:Class rdf:about="#ListCriteria">
  <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
  <rdfs:comment>
    Criteria that needs to be considered to choose an alternative.</rdfs:comment>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ProperRationale -->
<owl:Class rdf:about="#ProperRationale">
  <owl:equivalentClass>
    <owl:intersectionOf rdf:parseType="Collection">
      <!--
      http://www.semanticweb.org/ontologies/2008/11/Decision.owl#InformalDiscussion -->
      <owl:Class rdf:about="#InformalDiscussion">
        <rdfs:subClassOf rdf:resource="#DecisionAnalysis"/>
        <rdfs:isDefinedBy>
          Analysis document in an ad-hoc format.</rdfs:isDefinedBy>
      </owl:Class>
      <!--
      http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Issue -->
      <owl:Class rdf:about="#Issue">
        <rdfs:label>Issues</rdfs:label>
        <rdfs:subClassOf rdf:resource="#Artifact"/>
        <rdfs:isDefinedBy>
          A problem for which a decision needs to be taken.</rdfs:isDefinedBy>
      </owl:Class>
      <!--
      http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ListAlternatives -->
      <owl:Class rdf:about="#ListAlternatives">
        <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
        <rdfs:comment>
          Possible alternatives that could be recommended as a decision.</rdfs:comment>
      </owl:Class>
      <!--
      http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ListCriteria -->
      <owl:Class rdf:about="#ListCriteria">
        <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
        <rdfs:comment>
          Criteria that needs to be considered to choose an alternative.</rdfs:comment>
      </owl:Class>
    </owl:intersectionOf>
  </owl:equivalentClass>
</owl:Class>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasCriteria"/>
  <owl:someValuesFrom rdf:resource="#Criteria"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasAlternatives"/>
  <owl:someValuesFrom rdf:resource="#Alternatives"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasArguments"/>
  <owl:allValuesFrom rdf:resource="#ProperArguments"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasDecision"/>
  <owl:someValuesFrom rdf:resource="#Decision"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasIssue"/>
  <owl:someValuesFrom rdf:resource="#Issue"/>
</owl:Restriction>
</owl:intersectionOf>

<!-- </owl:equivalentClass> -->
<rdfs:isDefinedBy>
  >Analysis documented using this ontology that can be perceived as complete as it has all the constituents.</rdfs:isDefinedBy>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Rationale -->
<owl:Class rdf:about="#Rationale">
  <rdfs:subClassOf rdf:resource="#DecisionAnalysis"/>
  <rdfs:isDefinedBy>
    >Analysis documented using this ontology.</rdfs:isDefinedBy>
</owl:Class>

<owl:Class rdf:about="#RecommendAlternative">
  <rdfs:subClassOf rdf:resource="#RepeatableStep"/>
  <rdfs:comment>
    >Recommended alternative based on the arguments made.</rdfs:comment>
</owl:Class>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#AbleToWinState -->

<Criterion rdf:about="#AbleToWinState">
    <rdfs:comment>
        Will he or she be able to win a key state?
    </rdfs:comment>
</Criterion>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ArgumentForCandidates -->

<Arguments rdf:about="#ArgumentForCandidates">
    <hasArgument rdf:resource="#Bentsen_Ties"/>
    <hasArgument rdf:resource="#Home_State"/>
</Arguments>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#AssumptionsChoosingRunningMate -->

<Assumptions rdf:about="#AssumptionsChoosingRunningMate">
    <hasAssumption rdf:resource="#WinningTexas"/>
</Assumptions>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#BackgroundChoosingRunningMate -->

<BackgroundInfo rdf:about="#BackgroundChoosingRunningMate">
    <hasBackground rdf:resource="#RepublicanPresidentialNominee"/>
</BackgroundInfo>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Bentsen_Ties -->


Bentsen’s ties with big business were unusual for a Democratic nominee. Bentsen was one of the best fund raisers around and might be able to eliminate or even reverse the Republicans’ traditional financial advantage. Even if some of the more liberal voters were disenchanted, Bentsen could appeal to a more business-oriented constituency.

The main job of the vice-presidential nominee is to carry his or her home state. Could Betsen carry Texas? Many people thought that Dukakis could win Texas’s 29 electoral votes, however, the gamble would pay off dramatically,
depriving Bush of one of the largest states that he might have taken for granted.

<basedOnCriterion rdf:resource="#AbleToWinState"/>
</Argument>


<Criterion rdf:about="#IsFundRaiser">
  <rdfs:comment>
    Is he or she a good fund raiser?</rdfs:comment>
</Criterion>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Issue_RunningMate -->

<Issue rdf:about="#Issue_RunningMate">
  <rdfs:comment>
    The issue here is to choose a running mate for Dukakis.</rdfs:comment>
</Issue>


<Alternative rdf:about="#John_Glenn">
  <rdfs:comment>
    Senator John Glenn from Ohio</rdfs:comment>
</Alternative>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Lloyd_Bentsen -->

<Alternative rdf:about="#Lloyd_Bentsen">
  <rdfs:comment>
    U.S Senator from Texas</rdfs:comment>
  <owl:sameAs rdf:resource="#Running_Mate"/>
</Alternative>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#RepublicanPresidentialNominee -->

<Background rdf:about="#RepublicanPresidentialNominee">
  <rdfs:comment>
    The Republican presidential nominee was George Bush, whose own adopted
state was Texas.</rdfs:comment>
</Background>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Running_Mate -->

<Decision rdf:about="#Running_Mate"/>

<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#WinningTexas -->

<Assumption rdf:about="#WinningTexas">
  <rdfs:comment>
    Many people thought that Dukakis could win Texas' 29 electoral votes</rdfs:comment>
  </Assumption>
</rdf:RDF>

<!-- Generated by the OWL API (version 2.2.1.941) http://owlapi.sourceforge.net -->