Documents that “Think”

by A. W. Crapo and M. E. Graham, GE Global Research, 1/10/2011

Introduction

You are an expert in your domain. You use a specialized vocabulary: words that would be unfamiliar to people outside your field, or words that have a different or more specialized meaning to you and your colleagues. Perhaps you write technical documents that state precisely what is to be done under various circumstances. These documents might include service bulletins, repair manuals, or legal contracts. The contents of these documents may be interdependent. These documents use your specialized vocabulary and may contain conditional statements such as “Do XYZ if ABC occurs.” Of course the real relationships contained within your documents will probably be much more complex.

We have developed a language and a related tool set that allow you to define the concepts of your domain and the logical implications of various conditions in an English-like but unambiguous form. English, and for that matter all languages, are ambiguous and subject to interpretation. This language does not allow ambiguous statements. This means that once you have described the domain and the logic in our language, someone can supply a description of a specific problem or situation and a computer can tell them any conclusions implied by that description. In other words, the document is “executable”. Our language is called SADL.

What does SADL mean for you? It means that you can author documents that can be directly applied to given situations in a consistent manner. It means that your documents can be used to explain *why* they mean what they mean for a given situation. It means that your documents can be checked for logical consistency. It means your documents can extend the documents of others, in a verifiably consistent manner, and that others can similarly extend your documents to create a network of documents that “think”!

Describing the Domain with SADL

SADL is layered on top of the World Wide Web Consortium’s Web Ontology Language, which was developed to enable the content of the Internet to gradually be transformed into a large, distributed knowledge base. SADL makes it easier to say things and to see and understand what has been said. You begin as you might if you were training someone entirely new to your field of expertise to be your replacement—by describing the categories or types of things that you care about and then further dividing these categories into narrower sub-categories as needed. There is great power in defining categories and sub-categories because it allows you to only say things once. For example, if you say that Mammals are warm-blooded and Chimpanzees are a sub-category of Mammals, then you don’t need to say that Chimpanzees are warm-blooded because that is implied—it has already been said once in the right place.

You can also describe the kinds of relationships that might exist between different types of things. Relationships are very important— they are often the defining characteristic of something. Using relationships, you can describe a category in terms of the conditions necessary to belong to that category. For example, an Animal that has a “child” relationship with another Animal is a Parent. Using these and other ways of describing the concepts important in your field of expertise, you can create a conceptual model that is reusable, and the basis for adding the logic and business rules of your document.

Finally, you can identify instances of things you care about in your documents and the categories to which they belong. These are the actual, often real, things you care about, whereas the categories and relationships are useful abstractions. You also describe these instances in terms of their properties and relationships with other instances of things. In summary, you describe your domain in terms of categories of things, kinds of relationships between things, instances of things and their attributes, and how those things are actually related to other things.

When you describe a concept using another concept which has not been defined, the undefined concept is underlined in red with an explanation, just like if you mispell something in a word processor. However, you can continue to describe your domain and when you do define the concept, the red underline and explanation will be removed. This allows you to describe your domain top-down, defining concepts before you use them, or bottom-up, using concepts to define other concepts and then defining those after they are used, and so on.

The Rules

If you were training your replacement, a lot of what you would say, after you defined the concepts, would be conditional statements, “If this and this and this…. then ….” This kind of conditional statement is sometimes called a rule. Clearly, a rule should not use concepts that are not well understood. In SADL, one can write rules of the form, “Given …, if …, then …” but only if the concepts have been defined. If you use a concept that is not defined it will be underlined in red as described above.

The rules you provide in your document augment the things that are known implicitly. For example, if you are writing a document on zoo management in cold climates, you might have a rule that says, “If something is warm-blooded and native to a tropical climate, it should be kept indoors in the winter.” Then if someone is applying your document to a situation where it is January in Minnesota and Pansy is a Chimpanzee (and, referring to our previous example, Chimpanzee is a sub-category of Mammal), then the computer will be able to tell us that Pansy should be kept indoors. It is implied from the model that Pansy is a Chimpanzee, so Pansy is a Mammal, and that Mammals are warm-blooded, so Pansy is warm-blooded. Then the rule is used to conclude that since Chimpanzees are natives of West and Central Africa, which is a topical climate, and it is January which is winter in Minnesota, therefore Pansy should be kept indoors.

You can augment your documents with examples that can then be used to test the correctness of the models that the documents contain. This not only allows validation and re-validation of your models as you change or enhance them, but it provides examples to help someone understand exactly how your model is applied to real situations. These examples, however, are not just words. They are situations that are processed by your documents as real, working instances of how the set of documents "think."

Modularity and Reuse

You probably wouldn't choose someone new to your field to train as your replacement. Rather you would pick someone who already knows all of the basic concepts and just needs to understand what's unique about your job. Similarly, as more models are built and shared within and across organizations and disciplines, you will not need to start your description at the beginning but will start with existing documents and only add descriptions of the concepts unique to your job. Even if you do start from the beginning, you will want to divide your model up into parts—the general, reusable concepts, the unique concepts of your job, the specific rules by which you work, and examples of how your documents should be applied. When you write a technical paper, you don't repeat what's already been said in other documents. You just reference them and supply a bibliography. In SADL, you reference an existing document with an "import" and then all of the concepts in the imported document, along with any documents it imports, etc., are available to you in this document. Likewise, the documents that you write can be easily used, if appropriate, in other situations and by other writers as they import your documents. Some documents will define general concepts that will be imported and used in lots of places. Other documents will add very specific information for a very specific purpose and are less likely to be reused. The underlying technology is very modular; it has to be as the World Wide Web is very distributed.

For documents to be easily reusable over their entire life, they must be versioned and stored where others can find them, specify a desired version, compare versions, etc. The SADL authoring environment is integrated with systems for storing, versioning, and comparing documents. The ability to manage your documents is critical to their successful use and reuse.

Making SADL Documents Useful

SADL documents are often useful just because they have been created and form an unambiguous model representing the consensus of discussion. However, they can be even more useful when they are "exercised" by combining them with compatible descriptions of specific scenarios or situations. That Pansy is a Chimpanzee, that Pansy is in a particular Zoo that is located in Minnesota, and that the time is January are all information that is part of a specific situation. The document that captures the domain knowledge should not contain the situation-specific instances of Animal, Location, and Time. But once the specifics in a compatible format are combined with the documents describing the concepts and rules, a document processor can easily determine what, if any, conclusions can be logically drawn about the situation. This combining of the concepts and rules with a specific situation is where the "thinking" takes place.

Status

SADL (Semantic Application Design Language) has gone from being a proof-of-concept to being the foundation of a commercial application of significant size and impact. The language and authoring environment has also been used in several smaller projects with positive results. SADL Version 2 improves upon Version 1 in several respects..

1. The SADL V2 language covers all of the desirable constructs in the underlying Web Ontology Language (OWL-DL).
2. The authoring environment is built using tools that facilitate many helps for the author including help in completing statements, lists of available concepts, hyper linking concepts that are used to the definitions of those concepts, etc.
3. To examine the results or conclusions that come from combining a specific situation with a set of models, the SADL V2 language includes a query language of the same form as the declarations and rules and which translates to SPARQL. For complex queries SPARQL may still be used directly.

Most of the current implementation of SADL and its authoring environment has been released to Open Source (http://sadl.sourceforge.net). We seek opportunities to validate the language on new and interesting problems while simultaneously extending the language and authoring environment to be a complete and robust system for the creation and application of "thinking documents."