Abstract

Current practices in industries such as aerospace attempt to aggregate information from a wide area as part of their decision making process. However, collecting knowledge that is critical to a project is often daunting and time consuming. This paper describes the conceptualization and early development of a framework consisting of a semantic knowledge engine, archivist tool, and knowledge-mapping tool using a wiki front-end as a means for users to enter knowledge using a familiar web-based interface.

Categories and Subject Descriptors
1.2.4 [Knowledge Representation Formalisms and Methods]: Relation systems; Semantic networks.

General Terms

Keywords
Wikis; knowledge building; semantic analysis; data aggregation; knowledge management; search and retrieval.

1. Introduction
To be successful in today’s knowledge economy requires that learners, scholars, researchers, and practitioners be able to effectively and efficiently, collect, store, evaluate, connect, and retrieve knowledge and best practices. This is typically achieved with the use of a knowledge management system that, though effective for most purposes, can result in fragmentation of information. As an example, Ash [1] reports on a concern at Boeing. He writes: "what’s happened at Boeing, like many companies, is that the advent of personal computing has fractured the organizational database and the company’s critical knowledge is now scattered among personal computers and in a plethora of formats" (¶ 3).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WikiSym ’09, October 25–27, 2009, Orlando, Florida, U.S.A.
Copyright © 2009 ACM 978-1-60558-730-1/09/10...
$10.00.
2. COREIDEA FRAMEWORK

The componentized approach of the Coreidea framework divides the overall system into a series of separate, independently operating services. Working together, these components provide a coherent knowledge management system, allowing for both automatic higher-level semantic analysis of new content and easy retrieval of existing information, such as knowledge and best practices. The Wiki front-end presents a means for users to enter content using a familiar web-based interface. When the user edits or creates a new Wiki page, the Coreidea system channels their changes through a data aggregation and analysis pipeline. To simplify later processing, the first step involves data translation and enrichment. This it accomplishes using a translation gateway, which converts the data into a form that the rest of the system will understand, and feeds it onto the bus. On the bus, the data is now available for further interpretation. (see Fig. 1).

![Figure 1. Semantic knowledge engine conceptualization model](image)

Both of these document analysis techniques require word statistics, which makes the role of the aggregation service clear. As an aside, from a technical point of view, synchronizing the communication between the bus and aggregator and Archivist and aggregator, while maintaining a strict decoupling between the two services, is one of the challenges.

Both the Archivist and aggregator needs to treat new data in a carefully sequenced manner. The analysis uses data that must be absent from the data matrices it is processing, requiring the use of event based synchronization techniques in the underlying messaging layer.

The system stores a decision network for each known topic. It feeds the product of each separate document analysis into the decision network for each topic yielding an overall probability that that topic is appropriate for the content. The Archivist then presents these choices to the user, who can accept or reject each topic. It uses the decisions that the user made to back-calculate through the decision network, allowing the network to learn. With adequate training, the choices made by this processing pipeline returns expected user results.

Chimera, another front-end application, presents the user with a data entry oriented interface. The user enters in relational data using dynamic fields, which the system will internally store and aggregate. A query language interface provides a means for the user to create expert system style queries against the aggregated store. This allows for the creation of reports and automated analyses that the student, professional, or manager can use to make well-informed decisions. Chimera also feeds this data onto the Coreidea bus, initially through a translation gateway to normalize it, and then following the same process as with the Wiki front-end.

A significant deficit of many wiki systems is the need to perform exact or morphologically similar searches. With the semantic archivist data store, searches using concepts, rather than specific words, become possible.

3. REFERENCES


