Abstract

Electronic mail (e-mail) is an under-utilised resource of information and knowledge. It could be an important part of the larger so-called *organisational memory (OM)*—if it were not so disorganised and fragmented. The OM contains the knowledge of the organisation's employees, written records, and data. This thesis is about organising and managing information in, and about, e-mail so as to make it retrievable and usable for task management purposes.

The approach is user-centred and based on a conceptual model for task management. The model is designed to handle tasks that occur in the communications in an open distributed system, such as Internet e-mail. Both structured and unstructured tasks can be supported. Furthermore, the model includes management of desktop source information, which comprises the different electronically available sources in a user's computer environment. The information from these is used in the model to sort information and thereby handle tasks and related information. Tasks are managed as *conversations*, that is, exchanges of messages.

We present a language called Formal Language for Conversations (FLC), based on speech act theory, which is used to organise messages and relevant information for tasks. FLC provides the container for task-related information, as well as the context for managing tasks. The use of FLC is exemplified in two scenarios: scheduling a meeting and making conference arrangements.

We describe a prototype based on the conceptual model. The prototype explicitly refines and supports the notion of *threads*, which are employed so as to give tasks a *context*. It integrates the use of FLC into the traditional threading mechanism of e-mail, in addition to matching on text in the body. An agent architecture is also described, which is used to harmonise the information in the heterogeneous desktop sources. Finally, human-readable *filtering rules* created by a machine learning algorithm are employed in the prototype. The prototype is evaluated with regard to its thread-matching capability, as well as the creation of usable and readable filtering rules. Both are deemed satisfactory.