A knowledge processing system for data service network design

N J Fiddian, P Marti, J-C Pazzaglia, K Hui, A Preece, D M Jones and Z Cui

The knowledge reuse and fusion/transformation system (KRAFT) is research prototype software for combining and transforming constraint-based knowledge. It is being developed in collaboration with BT by three UK universities — Aberdeen, Cardiff and Liverpool. The KRAFT system is designed to help its users to locate relevant data and constraint knowledge held in distributed heterogeneous data and knowledge bases, convert this knowledge to generate a required composite problem specification, and exploit appropriate constraint solvers in solving the specified problem. The system architecture utilises intelligent software agent technology in the form of wrapper, facilitator and mediator agents for co-operative knowledge processing, and also a shared data model and a shared ontology as the basis for knowledge exchange.

This paper describes the application of KRAFT in the design of data service networks for BT. It presents, in turn, an introduction including an overview of the KRAFT system architecture, a description of the BT network design test application scenario, the application of KRAFT to this scenario, conclusions and further work.

1. Introduction

1.1 Motivation and related work

Agent-based architectures are proving to be an effective approach to developing distributed information systems [1], as they support rich knowledge representations, meta-level reasoning about the content of on-line resources, and open environments in which resources join or leave a network dynamically [2]. KRAFT employs such an agent-based architecture [3] to provide the required extensibility and adaptability in a dynamic distributed environment.

Unlike most agent-based distributed information systems, however, KRAFT focuses on the exchange of data and constraints among agents in the environment it supports.

Recent research in the area of software agent technology offers promising ways of supporting distributed design applications, but the area is still far from mature. Early projects such as PACT [4] and SHADE [5] showed that agent technology could support the exchange of rich business information (using the knowledge interchange format (KIF)) between organisations using heterogeneous technologies, with a limited amount of organisational agility — basic ‘matchmaking’ brokerage connecting suppliers to customers. While demonstrating the promise of the agent-based approach, these projects revealed problems — the complexity of the KIF representation has prevented it from gaining widespread use, while the limited brokerage model hinders the implementation of flexible negotiation schemes.

The design of the KRAFT architecture builds upon recent work in agent-based distributed information systems. In particular, the roles identified for KRAFT agents are similar to those in the InfoSleuth system [1]; however, while InfoSleuth is primarily concerned with the retrieval of data objects, the focus of KRAFT is on the combination of data and constraints.

KRAFT also builds upon the work of the Knowledge Sharing Effort (KSE) [6], in that some of the KSE facilitation and brokerage methods are employed, along with a subset of the 1997 Knowledge Query and Manipulation Language (KQML) specification [7].

Unlike the KSE work, however, which attempted to support agents communicating in a diverse range of knowledge representation languages (with attendant translational problems), KRAFT takes the view that constraints are a good compromise between expressivity and tractability.

1.2 Overview of the KRAFT system architecture

The KRAFT system has an agent-based architecture, in which all knowledge-processing components are realised as software agents. As the benefits and features of agent-based software architectures have been reported widely elsewhere [8—10], these will not be dwelt upon here. However, it is worth noting why an agent-based architecture was chosen for KRAFT:
agent architectures are designed to allow software processes to communicate knowledge across networks, in high-level communication protocols — since constraints are a sub-type of knowledge, this was seen as an important feature for KRAFT,

agent architectures are highly dynamic and open, allowing agents to locate other agents at run time, discover the capabilities of other agents, and form co-operative alliances — as KRAFT is concerned with the fusion of knowledge from available on-line sources, these features were seen as being of great value.

The design of KRAFT is consistent with several emerging agent standards, notably KQML [11] and FIPA [9, 10]. Agents are peers — any agent can communicate with any other agent with which it is acquainted. Agents become acquainted by registering their identity, network location and an advertisement of their knowledge-processing capabilities with a specific type of agent called a facilitator (essentially an intelligent Yellow Pages service). When an agent needs to request a service from another agent, it asks a facilitator to recommend an agent that appears to provide that service. The facilitator attempts to match the requested service to the advertised knowledge-processing capabilities of agents with which it is acquainted. If a match is found, the facilitator can inform the service-requesting agent of the identity, network location and advertised knowledge-processing capabilities of the service provider. The service-requesting agent and service-providing agent can now communicate directly.

It is worth emphasising that, while this model is superficially similar to that used in distributed object architectures such as CORBA [12] and DCOM [13], the important difference is the semantic level at which interactions take place. In distributed object architectures, objects advertise their presence by registering method signatures with registry services, and communicate by remote method invocations. In agent-based systems, advertisements of capabilities are much richer, being expressed in a declarative knowledge representation language, and communication uses a high-level conversational protocol built from primitive conversational actions such as ‘ask’, ‘tell’, ‘advertise’ and ‘recommend’. Distributed object architectures are in fact highly suitable for implementing agent-based architectures (for example, the ADEPT system used CORBA [8]), but the converse is not true.

A conceptual view of the KRAFT architecture is shown in Fig 1. KRAFT agents are shown as shaded ovals or rectangles according to whether they are internal or external to the KRAFT domain, respectively. There are four kinds of these — user agents, wrappers, mediators and facilitators. All of these are in some way knowledge-processing entities. The (external) knowledge resources with which they work are shown as unshaded rectangles in Fig 1.

User agents provide end users with entry-points into a KRAFT knowledge-processing system. A user agent will offer some kind of user interface, through which the user will present queries to the KRAFT network. The user agent wrapper will transform user queries into the internal knowledge representation language of the KRAFT system.