

From Agent Oriented Abstraction to Decision Making Under Uncertain Knowledge

Jacques Calmet and Yi Yang

University of Karlsruhe (TH),
Institute IAKS, Am Fasanengarten 5
76131 Karlsruhe, Germany
calmet@ira.uka.de, yangyi_sh@gmx.de

Abstract. We illustrate how a model of abstraction for the concept of agents, based upon a well-defined paradigm of what abstraction means in the context of artificial intelligence (AI), leads to a very practical application. The theoretical requirements for this application are to design and implement a decision support system in an environment of uncertain knowledge in a dynamically changing environment. Here uncertain means probabilistic. In practice, the resulting system is specialized to buy insurance covering risks in the area of natural catastrophies. Several tools are designed and implemented. Among them are the OntoBayes model which links Bayesian networks to influence diagrams and Virtual Knowledge Communities (VKC) based upon ontologies which enable the sharing and exchange of knowledge. The exchange of decision models is also a requirement in such a dynamical framework. This paper proposes an overview of the methodologies being designed.

Keywords: *Abstraction, Agent Oriented Abstraction, Decision support system, Ontology, OWL, Bayesian nets, Influence diagrams.*

1. Introduction

This introduction for this very short paper outlines the main concepts that are mentioned in the abstract and upon which the application is built. Abstraction is a word very often used but with very different meanings and contexts. In the domain of artificial intelligence, founded in 1956, there are even fewer attempts to define what an abstraction is and how to model it. The first one is due to Newell and Simon. Then, Wellman proposed to see a computer as a decision machine. Giunchiglia and Walsh introduced theories of abstraction for mechanized reasoning in a context of logic. An almost direct application was OMRS, Open Mechanized Reasoning System. Calmet and Homann used a similar approach for computer algebra. The two attempts were combined for symbolic computation systems (OMSCS). Relevant references are found in [Calmet 2004]. Recently there has been a renewal of interest in abstraction in AI through the SARA workshops held at IJCAI 2005 and 2007.

The open mechanized approach defines abstraction as a three levels structure. One needs first to have a theory (algorithms for instance), then one adds control (a programming language for instance) and finally one considers the environment in which this controlled theory will be active. This looks trivial but in fact it is possible to prove that computations do terminate, are unique and correct. Extension to numerical computations has been investigated and an extension to legal reasoning is under way [Calmet 2008]. Although the Open Mechanized approach looks mainly meaningful for computational processes, its three levels structure is suitable to abstract the concept of multi-agent systems (MAS) when one adopts the Agent Oriented Abstraction (AOA) paradigm. AOA consists of 6 definitions [Calmet 2004]. The first four ones deal respectively with the definition of an agent (annotated knowledge coupled to a decision mechanism), the decision mechanism, the knowledge annotations and the utility of a decision mechanism. This first group defines a theory. Definition 5 introduces the concept of a society of agent. This corresponds to the control imposed on a theory. The final definition is related to the specialization of the theoretical concepts defined so far and illustrates the interaction of a controlled theory with its environment.

2. An abstract framework of decision support systems

One of the most daunting challenges for decision making is uncertainty. Nowadays there are too many kinds of decision support systems [Power 2003]. Most of them are designed and developed based on precision and certainty. They are infeasible to work under uncertainty. They provide unreliable solutions based on the assumption of closed environments. For most real applications, uncertainty is an inevitable feature and can not be ignored. Agents do not act within a static and close environment but within a dynamic and open one. The available information is mostly incomplete and often imprecise because agents almost never have access to the whole truth implied by their environment. Agents must therefore act under uncertainty, and must be able to make optimal decisions with limited computational resources.

In order to make decision under uncertainty possible, an abstract framework was proposed in [Yang 2007]. This framework touches the following research fields: ontologies, Bayesian networks, influence diagrams, virtual knowledge communities, multiagent systems and web services. As depicted in Figure 1, a DSS rests on a 4-pillar framework and each pillar serves as a building block of the system. We can group all pillars into two classes —either theoretical or technical foundations—in the left-to-right sequence. The first two pillars are the theoretical foundations for decision making, while the last two pillars are considered as the most important technical foundations to implement the system.

The core work of this framework is an ontology-driven uncertain model, *OntoBayes*, which is designed with two parts: a knowledge part and a decision model part. The former is an integration of certain and uncertain knowledge based on ontologies and BNs respectively, while the latter can describe different decision models based on IDs. The design of *OntoBayes* is inspired by AOA. The knowledge part and the decision part are abstracted within different layers.

In order to facilitate the use of *OntoBayes* in DSSs, particularly to facilitate the sharing and reuse of knowledge and decision models for decision makers, a formal language for representing the knowledge part and the decision model part is mandatory. OWL was selected as the underlying knowledge representation language for *OntoBayes*. Therefore the design of *OntoBayes* must be implemented based on OWL. For this goal, OWL was extended with the features of BNs and IDs. The extensions are mainly based on different upper ontologies. Each of them can abstractly describe and specify different methods, either Bayesian networks or influence diagram, in a formal way. This model facilitates the representation of uncertain knowledge which is one of the most important building blocks for decision making under uncertain knowledge.

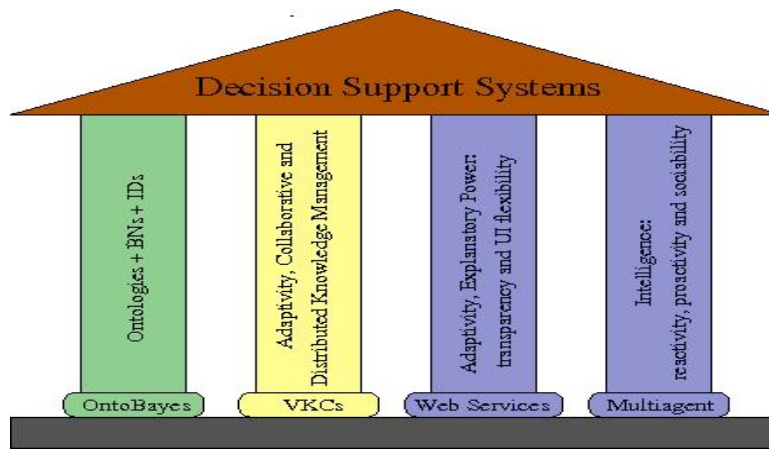


Figure 1: An abstract framework of Decision Support Systems.

Another building deals with the transfer of uncertain knowledge. For such a goal the methodology of VKC is used. VKCs can be abstractly defined as a mean for agents to share knowledge about a topic [Maret 2004]. It aims to increase the efficiency with which information is made available throughout a society of agents. The original design and implementation of VKC was concerned with certain ontological knowledge in form of RDF. In our framework VKC is used as the platform of decision support systems for facilitating knowledge exchange between different agents and it aims not only at certain knowledge, but also at uncertain knowledge supported by OntoBayes. To make this feature possible, the VKC's approach has been extended and optimized, especially through the integration with OWL.

The Multiagent paradigm is one of the selected overall technical foundations in DSSs. It is used to address the challenge of system intelligence. This introduces intelligent agents as a powerful metaphor in the field of DSSs. These agents link certain decision problem types and the decision environment in which they operate. They possess the following basic characteristics: reactivity, pro-activity and sociability. These basic characteristics make them suitable as software entities for the delegation of diverse decision making tasks and for collaborative work with each other.

Web Services as another selected technical foundation of DSSs is used to address the challenges of system adaptability, explanatory power and UI flexibility. We design the system architecture in which all components are built in as web services so that users or system agents integrate them and perform agent tasks helping users according to various knowledge sources. The whole system is developed as SOA (Service Oriented Architecture), which provides the system with the following features: loose coupling, implementation neutrality, flexible configurability, eextensible lifetime and granularity [Singh 2005].

3. Architecture of the System and Implementation

In this section the system architecture will be briefly described to outline the implementation of the framework proposed in the previous section. As depicted in Figure 2, this architecture has many components and each one of them can be assigned to a layer of the framework according to the color representing it. Components in cyan, magenta, yellow, and green belong to the repository, management, collaboration and application layer respectively.

In the repository layer, there are at least two kinds of repositories: a service repository and an OntoBayes repository. The *Service repository* is used to store service descriptions. All published services can be retrieved from this repository. The *OntoBayes repository* is in fact the public knowledge and decision model storage of the system. In the OntoBayes model there are three kinds of OWL files: the ontological knowledge, the Bayesian knowledge and decision models. They can be distinguished with different upper ontologies. Other repositories can be designed either as part of an extension of the system or on the demand of users. For example, this covers repositories for storing policy or for new business processes.

In the management layer there are several types of management components and each of them has its own functionalities and provides different services.

- A *repository management* provides basic services to manage all kinds of repositories, e.g. operations for access, recovery and long term preservation of repositories.
- A *decision management* takes responsibility for managing all activities directly related to decision making. These activities are for instance to manage different decision making processes, to distinguish different decision support levels (simple query vs. decision analysis) and so on.
- A *knowledge management* should provide services to manage the knowledge life cycle: knowledge generation, knowledge codification, knowledge transfer (also known as knowledge sharing), and knowledge application [AL01]. We take only knowledge generation and codification into account in this system component, because in this work knowledge transfer is theoretically designed based on VKCs .

- A *reasoner management* has the responsibility for assigning algorithms for reasoning about a BN or evaluating an optimal policy of an ID. Building a reasoner management component in a DSS separately is useful if there are many algorithms available for solving a decision problem. For example agents can make use of an exact reasoning to query a Bayesian variable, but they can also make use of approximate reasoning to perform this work.
- A *user management* provides authentication and authorization services for users. Each user has a different role in the system and has also different rights. For example, there are knowledge workers, domain experts, knowledge engineers, administrators, decision makers (i.e. the end users) and so on.

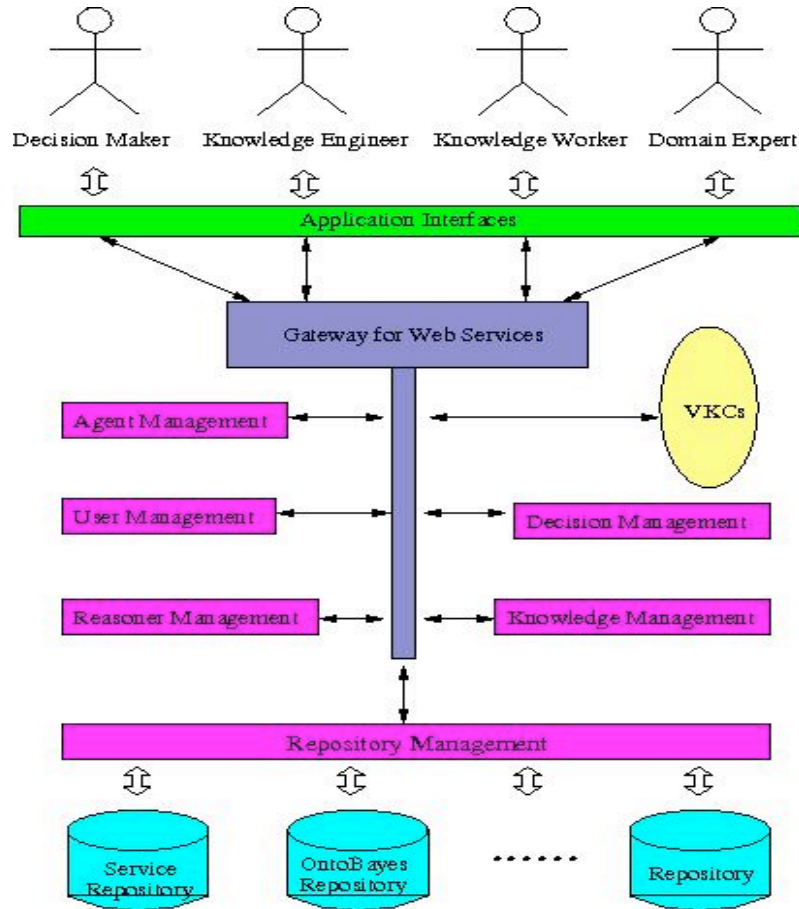


Figure 2. An Architecture for the system implementation

- An *agent management* takes responsibility to manage all agents. For example, to create agents, to terminate agents and so on. There are two kinds of agents running in the system: system agents and user agents. The former run on each management component and have its functionalities. They are created by the system. The latter are created by users with assigned tasks.

In the collaboration layer there are VKCs which provide a virtual place for agents to meet and work together. Two main functionalities will be provided in this layer: corporate knowledge support for decision making through knowledge exchange and collaborative working of virtual teams.

The front end of the system, the application layer, provides user interfaces based on web services to support decision problem solving. Decision support in the application front end will be realized through the utilization of web services provided by the system back end.

4. Conclusions

The main goal of this very short paper was to illustrate that a sound theoretical approach leads to design a system that can be used for very practical application purposes. Because of the format of the paper, it was only possible to outline very roughly the features of the system. For the same reason, there was no room left to introduce the application domain. This domain is the insurance industry and the specific task was to decide on buying “flooding insurance” in the area of Shanghai. Another goal was to emphasize that to cover uncertain information (here flooding risks) in DSSs is very challenging. This is probably why no satisfactory commercial system is on offer nowadays.

5. References

J. Calmet, P. Maret, and R. Endsuleit (2004), *Agent-Oriented Abstraction*, Revista Real Academia de Ciencias (Madrid), Special Volume on Symbolic Computing and Artificial Intelligence, Vol. 98, No. 1 (pp. 77-83).

J. Calmet (2008), *Open Mechanized Legal Reasoning*, in preparation.

P. Maret, J. Subercaze, and J. Calmet (2008), *Peer-to-Peer Model for Virtual Knowledge Communities*, to appear in the proceedings of AIKED 2008, Cambridge, February 2008.

Yi Yang (2007), *A Framework for Decision Support Systems Adapted to Uncertain Knowledge*, Ph. D thesis, university of Karlsruhe (TH).

Daniel J. Power (2003), *A Brief history of Decision Support Systems*. *DSSResources.COM*.

P. Maret, M. Hammond, and J. Calmet (2005), *Virtual Knowledge Communities for Corporate Knowledge Issues*, in *Proc. of 5th International Workshop on Engineering Societies in the Agents World (ESAW'04)*, Springer, LNAI, Vol. 3451, pp. 33–44.

M. Alavi and D. Leidner (2001). *Knowledge Management and Knowledge Management Systems: Conceptual foundations and research issues*, *MISQ Review*, Vol. 25, No.1 (pp. 107–136).