Legal knowledge based systems
JURIX '95
Telecommunication and AI & Law

The Foundation for Legal Knowledge Systems
Editors:
J.C. Hage
T.J.M. Bench-Capon
M.J. Cohen
H.J. van den Herik

Constantijn Heesen, Vincent Homburg and Margriet Offereins, LACA: An architecture for legal agents, in: J.C. Hage, T.J.M. Bench-Capon, M.J. Cohen, H.J. van den Herik (eds.), *Legal knowledge based systems JURIX '95: Telecommunication and AI & Law*, Lelystad: Koninklijke Vermande, 1995, 23-32, ISBN 90 5458 252 9.

More information about the JURIX foundation and its activities can be obtained by contacting the JURIX secretariat:

Mr. C.N.J. de Vey Mestdagh
University of Groningen, Faculty of Law
Oude Kijk in 't Jatstraat 26
P.O. Box 716
9700 AS Groningen
Tel: +31 50 3635790/5433
Fax: +31 50 3635603
Email: sesam@rechten.rug.nl

 http://jurix.bsk.utwente.nl/

LACA: AN ARCHITECTURE FOR LEGAL AGENTS

Constantijn Heesen, Vincent Homburg and Margriet Offereins
University of Groningen, Faculty of Management and Organisation,
P.O. Box 800, 9700 AV Groningen

Abstract

In this paper we propose an architecture for legal agents: autonomous legal knowledge-based systems with facilities to communicate their intentions to other legal agents, enabling the spanning of legal tasks over different co-operating legal agents. This architecture (LACA: Legal Agents Communication Architecture) is based on agent theories and the Dutch General Administrative Law (*Algemene Wet Bestuursrecht*). LACA consists of specific communication primitives and conversation classes.
Keywords: Legal KBS, Agents, Communication, LACA.

1 Introduction

Information technology (IT) is becoming increasingly crucial to the viability of organisations (Simons, 1992). Through the years, different types of IT can be distinguished. In general, it can be said that the early visions of information systems which were based on the database concept have been expanded through the addition of knowledge bases and, more recently, communication facilities (Table 1).

1970s	*DB* technology	*database* oriented systems
1980s	*KB-DSS* technology	*knowledge-based decision-support systems*, that add user interfaces and knowledge-base concepts to the data-base systems
1990s	*Agent* technology	*agent systems*, that add communication facilities to the KB-DSS concept

Table 1: Trends in IT

To our knowledge, there have not yet been examples of legal systems with communication facilities. In this paper, we use the notion of an 'agent' to model the communication between legal knowledge-based systems. An 'agent' can be defined as "an integrated entity involving a computer system and its user" (Huang *et al*., 1994, p. 221). In the next section, specific characteristics of agents will be described.

1.1 Agent characteristics

Using a weak notion of agents (Jennings and Wooldridge, 1994), one can identify characteristics that distinguish agent-based systems from knowledge-based systems or knowledge-based decision-support systems (Table 2).

These agent characteristics make agent-based systems especially appropriate for legal organisations with distributed knowledge, problem-solving capabilities, resources and responsibilities. In these organisations, co-operation has to be managed in order to fulfil tasks. Current legal knowledge-based systems only support very specific tasks. In situations where problems are considered that span different legal areas (and different knowledge-based systems), some kind of co-operation between legal knowledge-based systems must be achieved in order to solve problems that are beyond the capabilities of individual knowledge-based systems. Therefore,

conventional legal knowledge-based systems should be augmented with communication facilities.

autonomy	agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state
social ability	agents interact with other agents via some kind of agent-communication language
reactivity	agents perceive their environment and respond in a timely fashion to changes that occur in it
pro-activeness	agents do not simply act in response to their environment; they are able to exhibit goal-directed behaviour by taking the initiative

Table 2: Agent characteristics

An example of this situation occurs in the networks of government and executive organisations that have been privatised for broader participation in the marketplace, for example in the field of education in the Netherlands (apart from the Ministry, there is the *Informatie Beheer Groep*, the C*f*i, *Inspectiedienst*, etc.).

To provide support for the design and management of knowledge-based systems in a network of organisations, one can think of adopting an agent-based approach because of the properties of autonomy, social ability, reactiveness and proactiveness that are both associated with agents as well as with organisations in a network (Huang *et al*., 1994).

1.2 The General Administrative Law (Algemene Wet Bestuursrecht)

In this paper, we will apply ideas from the network- or agent-oriented view to the field of administrative law in the Netherlands, specifically the AWB (*Algemene Wet Bestuursrecht* or General Administrative Law). The AWB regulates the relations between government and civilians by providing a.o. general rules both for the preparation, the motivation and the announcement of administrative orders, and for the objection and the appeal against administrative orders. These regulations describe the steps each party involved should take in respectively the normal procedure in which an administrative order is requested, the objection procedure in which an objection is made against an administrative order, and the appeal procedure in which an appeal is made against an administrative order. The general rules for each of these procedures structure the communication process of the parties involved and provide the communication procedures that parties should use in order to communicate properly according to the AWB.

Depending on the kind of procedure that is performed in the field of administrative law, different types of parties are involved. Thus, a conversation involves different kinds of parties communicating with each other. For instance, we distinguish legal organisations that issue administrative orders, such as adjudicators of social security benefits, the courts that handle appeals, and the objection committees that handle objections.

Below, we present an approach that helps to see legal organisations as being part of organisational networks, we propose a general architecture for agents in distributed organisational settings, and we show how the AWB structures the communication process between agents.

2 Legal Agents

In section one, we described a weak notion of an agent. A stronger notion of agents is that agents have mentalistic notions, such as knowledge, belief, intentions, and obligations. By describing agents in these terms, one takes an *intentional stance.* In general, one can say that for simple systems, a more mechanistic description fits the job. However, with more complex systems, even if a complete, accurate picture of the system's architecture and working is available, a mechanistic *design stance* explanation of its behaviour may not be practicable (Wooldridge and Jennings, 1994).

From the stronger notion of agents, it is clear that building legal knowledge-based systems and providing them with some sort of communication protocol does not make the system an agent. Interaction between agents is more than the simple exchange of messages. In summary, issues associated with agent theories are (Labrou and Finin, 1995): models of agents (beliefs, goals, representation and reasoning), interaction protocols (an interaction regime that guides the agents) and interaction languages (languages that introduce standard message types that all agents interpret identically).

A *legal* agent can be described as a special kind of agent (as described above) that is able to handle legal knowledge, has legal intentions and can participate in legal conversation. We limit the use of the term 'legal agent' to agents that operate within the scope of the AWB: e.g., governmental executive organisations, organisations or civilians, and administrative courts.

3 LACA

3.1 Background

We model the communication and intention part of a legal agent by using speech act theory (Searle, 1969). Speech act theory is a high-level theoretical framework, developed by philosophers and linguists to account for human communication. It has been extensively used, formalised and extended within the fields of Computer Linguistics and AI as a general model of communication between arbitrary agents (Labrou and Finin, 1995). As such, speech act theory helps us to model the relationships various legal agents can participate in, including interaction patterns (conversations) and belief structures.

In speech act theory, action is tightly connected with speech. There are three distinct actions in speech act theory:

- locutions: the actual physical expressions,
- illocutions: the communication of the speaker's intention, and
- perlocutions: actions that occur as a result of the illocution.

A legal agent communication language could use *primitives* in the form F(p) to express illocutionary force from a legal agent to another one. In this speech act, F indicates illocutionary force and p expresses propositions. The *perlocutionary* effects are the changes in the state of both sender and receiver, and could produce some new, expected responses.

After analysis of the basic procedures of the AWB (Ministerie van Justitie and Ministerie van Binnenlandse Zaken, 1994), a set of communication primitives[1] in the tradition of speech act theory has been defined. We believe these primitives can be used in the context of legal knowledge-based systems that have to communicate with each

[1] Labrou and Finin (1995) propose the term 'performatives' in their agent communication language 'KQML'. KQML is intended to be a universal interaction language, that supports communication through explicit linguistic actions.

other to solve problems that are beyond their own capabilities. These communication primitives are described as speech acts in terms of an illocution and a perlocution (Table 3).

Illocution type	**Propositional Content (AdO = Administrative Order)**	**Perlocution (RAg = ReceiveAgent, SAg = SendAgent)**
request	AdO; response_by date	RAg evaluates whether to accept the request, and informs SAg of decision. If RAg decides to accept the request, it becomes committed to the request and will issue an AdO.
accept	request; response_by date	RAg knows that SAg is committed to the request and that SAg will issue an AdO.
reject	request	RAg knows SAg will not commit to the request.
summon	date	RAg comes to a hearing.
alter	request	RAg knows SAg passed on the request.
query	a request for information; response_by date	RAg must answer the query.
inform	any information: data, domain knowledge	RAg uses the information.
consult	AdO; response_by date	RAg gives an advise on the AdO.
notify	state	RAg knows what's going on.
grant	request; response_by date	RAg knows the request has been granted.
refuse	request; response_by date	RAg knows the request has been refused.
object	AdO	RAg should reconsider the AdO.
judge	request	RAg knows what AdO should be issued.
appeal	AdO	RAg should reconsider the AdO.
advise	AdO	RAg may use the advise on the AdO (for example to motivate a decision).
acknowledge	any message	RAg is aware of the successful transmission of the message.
cancel	any message	RAg should ignore the earlier message.
resist	legal procedure	RAg should reconsider the legal procedure.

Table 3: LACA communication primitives

3.2 Legal Agent Architecture

The architecture of the LACA Legal Agents (Figure 1) is based on the expertise model of CommonKADS and the agent model of (Huang *et al.*, 1994). CommonKADS is a model-based development approach for knowledge-based systems.

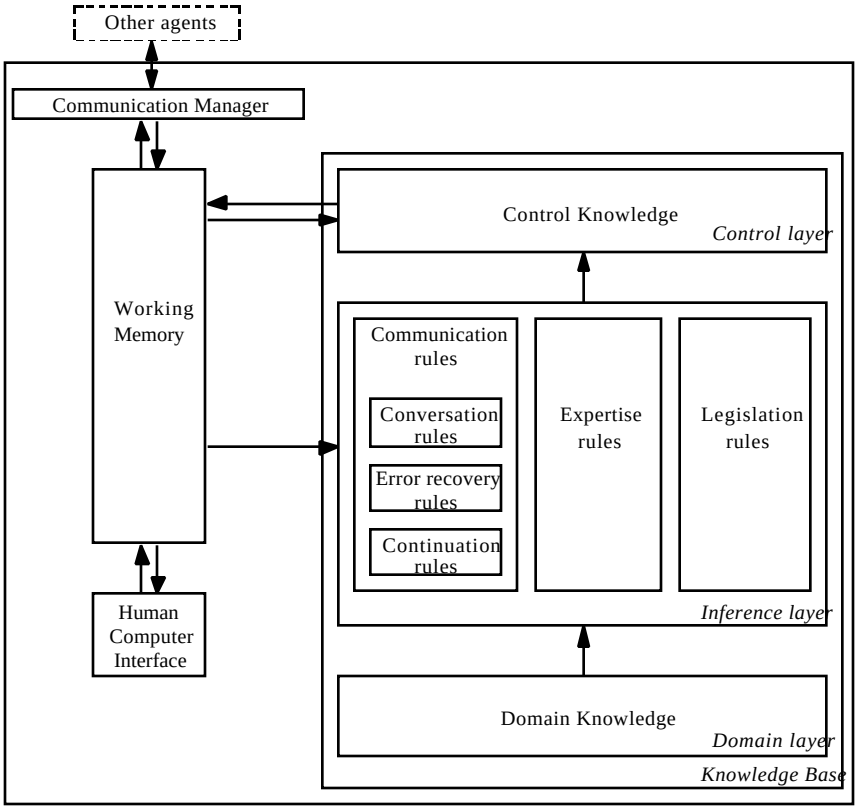

Figure 1: LACA architecture

An important element of this model set is the Expertise model. In the Expertise model the knowledge of the system is specified in three layers, the domain layer, the inference layer and the task layer. The counterparts of these layers in the Legal Agent Architecture are the domain layer, the inference layer and the control layer, together constituting the knowledge base of the Legal Agent. There are various reasons for the separation of domain, inference and control knowledge. We name two: First it facilitates reuse and maintenance of knowledge, because the knowledge in the inference and control layer is generic, i.e., not dependent on the domain knowledge. A second reason is to distinguish the world knowledge from the regulation knowledge (Breuker and den Haan, 1991).

The AWB is a typical example of a generic, domain-independent regulation. Besides the knowledge base, the LACA legal agent architecture consists of three other parts (inspired by the Huang architecture), viz. a working memory, a human-computer interface and a communication manager. All communication primitives and state descriptions are represented in the *working memory*. Through the working memory the control layer is activated and results are put back. One can think of the working memory as a blackboard. Communication primitives coming from other agents via the communication manager or from the user via the human-computer interface, are posted on it. The control layer (see section 3.3) is triggered by incoming communication primitives and reacts in an appropriate way. The *human-computer interface* facilitates the communication between the user and the support system. In this way the user can

control the processes of the system and can interact with the system, for example, to authorise an action before the system sends it to another agent. Communication with other agents is provided by the *communication manager*. The communication manager sends messages, with communication primitives out of the working memory, which were put there by the control layer, to other agents and sets the communication primitives of incoming messages back into the working memory. A message consists of one or more communication primitives (see Table 3), the address of the recipient and the name of the sender. As we will see in section 3.4 a set of dependent messages constitutes a conversation.

LACA is comparable to other legal knowledge-based systems, such as LOD3 (Taylor, 1991). The agent architecture differs on two points from 'normal' legal knowledge-based systems. These differences are the addition of a communication manager and of communication rules inside the knowledge base. In the next sections we take a closer look at the knowledge base of the LACA agent architecture by describing the domain, inference and control layer.

3.3 Legal Agent Conversation

In section 3.2 we described the intentional level of agent interaction by means of communication primitives. In this section we describe the coordination level of agent interaction using the coordination language COOL (Barbuceanu and Fox, 1995). This language has been used particularly for describing coordination in the supply chain of an enterprise, but is also generally applicable as a coordination specification language for a multi-agent system. Below, we describe the basic components of this language: conversation classes, conversation rules, error recovery rules and continuation rules. Figure 1 shows how we incorporated these components into the legal agent architecture.

Domain knowledge
The domain knowledge is a specification of the application specific concepts and the relationships between the concepts. Examples of concepts from the AWB domain in the domain layer are a.o. an administrative order, an organisation or a conversation class. A conversation class (Figure 2) is a vital concept of LACA. It specifies the states, conversation rules and error rules that are specific to a type of conversation (Barbuceanu and Fox, 1995).

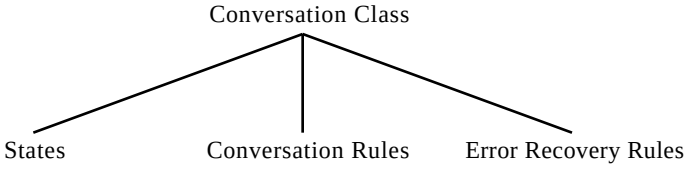

Figure 2: Conversation class

A conversation is coupled to a procedure. The AWB describes several types of procedures: the normal procedure, the objection procedure and the appeal procedure (see section 1.2). The conversation belonging to a procedure is specified in the domain layer, except for the conversation and error rules, which are specified in the inference layer. Agents may be engaged in several conversations at the same time. In the next section we explain the conversation and error rules. Section 3.4 describes the states a conversation can be in, for example, an agent can be in a state waiting for extra information before accepting a request.

Inference knowledge
The inference knowledge consists of generic, declarative inference rules. These rules specify the relations between objects in the domain layer. The inference knowledge is divided into three parts: the legislation rules, the expertise rules and the communication rules.

The *legislation rules* are a representation of the law. The *expertise rules* represent additional case law and interpretative material. This distinction between legislation and expertise rules is made more often in literature (e.g., Bench-Capon, 1991; Taylor, 1994). For example, Bench-Capon states that at the core of a system we find legislation, but this must be supplemented with knowledge about the interpretation of the law (expertise), in order to make a useful system.

The third part of the inference layer is a set of the *communication rules.* These rules control the communication between other legal agents. The communication rules are an implementation of the AWB, because this law has, seen from an agent perspective, the function of regulating the communication between legal agents. The rules are distinguished into, again, three parts, the conversation rules, the error recovery rules and the continuation rules (Barbuceanu and Fox, 1995). A set of *conversation rules*, belonging to a conversation class, specify how an agent in a given state receives a message of a specified type, performs local actions (e.g., updating local data), sends out messages, and switches to another state. The next rule is an example of a conversation rule.

```
IF    state(3, request_extra_data) AND
      deadline(3, request_extra_data, passed)
THEN
      reject(request)
```

This AWB-based rule states that the communication primitive *reject* must be sent, if the conversation is in state 3 (the adjudicator waits for extra data) and the deadline for the submission of extra data is passed (see also Figure 3).

If there are incompatibilities among the state of a conversation and the incoming messages, the *error recovery rules* are invoked. The *continuation rules* do not belong to a conversation class but are specific to an agent. They specify how an agent accepts new requests for a conversation or select a conversation to continue from among the existing ones. One can also think of continuation rules as task management rules. We can now interpret the AWB as a set of communication rules prescribing patterns for conversations between Legal Agents.

Control knowledge
The control layer is a meta-level that applies the inference layer to the domain layer in order to generate new inferences whenever new data are added to the working memory. It is at the control level that the actual execution of the inference rules is carried out.

For example, within the context of the extra data request, once the data

```
state(3, request_extra_data)
```

and

```
deadline(3, request_extra_data,passed)
```

are asserted to the working memory, the control layer applies the given inference rule and domain knowledge to add a new piece of data into the working memory:

```
reject(request).
```

3.4 An example of Legal Agent Conversation

As we pointed out in section 1, three different types of conversation can be distinguished in the field of administrative law. Each type of conversation corresponds

to one of the procedures that may be applied. In the domain layer we define these as different conversation classes, each specifying the specific states, conversation rules and error recovery rules for a type of conversation. An agent has several conversation classes which it can use when communicating with other agents (Barbuceanu and Fox, 1995).

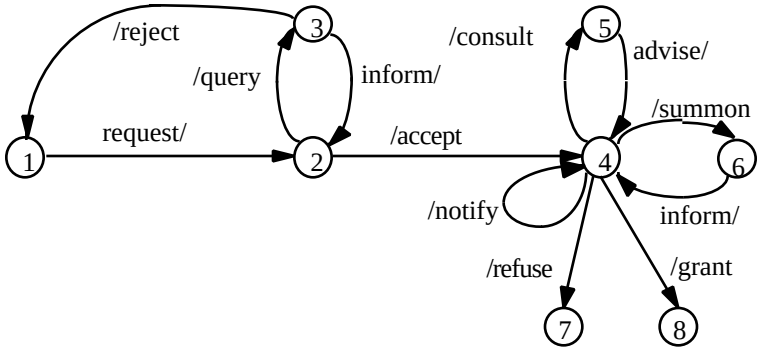

Figure 3: FSM of conversation for normal procedure

An example of a conversation is described by a Finite State Machine (FSM) representing the states that a conversation can be in. The transition from one state to another is triggered by a speech act.

Figure 3 shows the FSM of the normal procedure for issuing an administrative order by an executive organisation on a request from a civilian. In this figure the notation <received speech act> / <sent speech act> is used to label edges in the graph. When the conversation is in a given state and a speech act is received, the receiving Legal Agent performs some local inferencing, sends out a speech act (see Table 3) and switches to the next state. State 1 is the initial state and states 7 and 8 are final states. The conversation starts in state 1 and switches to state 2 on a request for an administrative order (AWB section 4.1.1., art. 4:4). If there is something wrong with the request, a query for information is sent out by the Legal Agent to the citizen. If the information is not received in time or if the request is still not valid, a reject is sent to the citizen (AWB section 4.1.1., art. 4:5). In accepting the request, the Legal Agent commits itself to issuing an administrative order and will proceed to state 4. In state 4 the administrative order is being prepared. The Legal Agent may summon the citizen for a hearing to get more information (AWB section 4.1.2) or may consult an external advisor (AWB section 3.3). If the issuing of the administrative order is being delayed, it should be notified (AWB section 4.1.3, art 4:14). The result of state 4 is the announcement of the administrative order (AWB section 3.6, art. 3:40-3:44) that is either a refusal or a grant of the request. One can make an objection against the administrative order, and eventually an appeal. Then a new conversation starts that is described by the corresponding conversation class and corresponding FSM representations.

4 Conclusion and discussion

In this paper, we have adopted an agent-oriented view on organisations in the field of administrative law and we have proposed an agent architecture called LACA. We used the AWB to structure interaction patterns, but interaction between legal agents consists of more aspects than specified by this law. However, the AWB provided us with sufficient insights to illustrate the formal communication and interaction between legal agents.

With LACA, we illustrate the possibilities of using agent theories in the field of legal knowledge-based systems to augment current legal knowledge-based systems with communication facilities at a conceptual level. Interaction is more than a simple exchange of messages. Legal agents built according to the LACA architecture resemble the architectures of traditional (legal) knowledge-based systems, but add speech act based communication primitives to enable knowledge-based system to communicate their intentions to other agents that could be located elsewhere, for example by using the Internet.

In short, adding communication facilities to (legal) knowledge-based systems asks for adopting a (legal) agent approach.

In this way, for example giving administrative orders as a legal task spanning multiple legal areas, in which various legal knowledge-based systems exist, can be fulfilled by co-operating legal agents, each having the characteristics of autonomy, social ability, reactiveness and proactiveness. The administrative orders could be sent automatically too, using telecommunication media, and received, processed and reacted upon (by means of objections or appeals) by other legal agents that participate in a relationship with a government legal agent (for example, by requesting an administrative order). In this way, using LACA as an architecture for legal agents, relationships between administrative bodies and organisations or citizens can be modelled and parallels can be drawn between requesting an administrative order and new concepts like banking-by-phone and video-on-demand.

References

Barbuceanu, M., and Fox, M.S. (1995). COOL: A language for coordination in multi agent systems. *Proceedings of the First International Conference on Multiagent Systems* (ed. V. Lesser). MIT Press, Cambridge.

Bench-Capon, T.J.M. (ed.) (1991). *Knowledge-Based Systems and Legal Applications*. Academic Press, London.

Breuker, J.A., and Haan, N. den (1991). Separating world and regulation knowledge: where is the logic? *Proceedings of the third international conference of AI and Law* (ed. M. Sergot). ACM, New York.

Huang, J., Jennings, N.R., and Fox, J. (1994). An Architecture for Distributed Medical Care. *Intelligent Agents* (eds. M.J. Wooldridge and N.R. Jennings). Springer-Verlag, Berlin.

Jennings, N.R. (1993). Commitments and conventions: the foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, Vol. 8. No. 3.

Jennings, N.R., and Wooldridge, M. (1994). Applying Agent Technology. *Intelligent Agents* (eds. M.J. Wooldridge and N.R. Jennings). Springer-Verlag, Berlin.

Jorna, R.J., and Simons, J.L. (eds.) (1992). *Kennis in organisaties*. Coutinho, Muiderberg.

Labrou, Y., and Finin, T. (1995). A semantics approach for KQML - a general purpose communication language for software agents. *CIKM. Proceedings of the Third International Conference on Information and Knowledge Management.* Springer-Verlag, Berlin.

Ministerie van Justitie and Ministerie van Binnenlandse Zaken (1994). *Algemene Wet Bestuursrecht* (brochure). SDU, The Hague.

Searle, J.R. (1969). *Speech Acts: an essay in the philosophy of language*. Cambridge University Press, London.

Simons, J.L. (1992). Kennismanagement, de strategische rol van kennis in organisaties. In: *Kennis in organisaties* (eds. R.J. Jorna and J.L. Simons). Coutinho, Muiderberg.

Taylor, A. (1991). A Support Environment for Adjudicators. *Knowledge-Based Systems and Legal Applications* (ed. T.J.M. Bench-Capon). Academic Press, London.

Wielinga, B.J. (ed.) (1994). *Expertise Model Definition Document*. Working paper ESPRIT project P5248 KADS II. University of Amsterdam, Amsterdam.
Wooldridge, M.J. and Jennings, N.R. (eds.) (1994). *Intelligent Agents*. Springer-Verlag, Berlin.