Computational Models of Argument
S. Modgil et al. (Eds.)

doi:10.3233/978-1-61499-906-5-361

Argumentation Schemes for Data Access Control

Alison R. PANISSON [a,1], Asad ALI [b], Peter MCBURNEY [b], and Rafael H. BORDINI [a]

[a] *School of Technology, PUCRS – Porto Alegre, Brazil*
[b] *Department of Informatics, KCL – London, UK*

Abstract. One of the main challenges in integrating different smart applications is security. Among the problems related to security, data access control is one of the most important, given that it involves end-users' privacy. In this paper, we propose an approach to the modelling of data access control interfaces using argumentation-based agents. In particular, we introduce argumentation schemes for data access control, which are based on some of the most relevant models for data access control currently available. Our approach considers not only the usual access control policies, describing which category of agents has access to which category of information, but also emergency policies, describing situations where special emergency access control rules apply. Using argumentation-based agents as access control interfaces allows us to deal with the uncertainty about external information, allowing a correct categorisation of agents that request access to information, as well as allowing agents to expose emergency situations in which emergency access control rules may apply.

Keywords. Data Access Control, Argumentation Schemes, Multi-Agent Systems

1. Introduction

Security concerns are increasingly important, mainly regarding sharing and using data given the increasing use of IoT (*Internet of Things*) devices and the proliferation of big data mechanisms [1]. Confirming the IoT vision, the number of devices connected to the *Internet* has been increasing and consequently the amount of data available on the internet has also been increasing [2]. This creates opportunities for studies in areas such as big data analysis, machine learning, and many others.

Data access/sharing control were not sufficiently studied in previous decades, considering that data were shared within boundaries of organisations, e.g., companies and universities. In such organisations, trust and security issues were easily solved [1]. However, with the current computing trends, including the integration of different smart applications (e.g., healthcare, smart cities, smart home/building, etc.), sharing information between different systems has become mandatory [3]. Consequently, the problems of data sharing/access control and privacy protection have become challenging issues when integrating different smart applications [4].

[1] Corresponding Author: Alison R. Panisson – e-mail: alison.panisson@acad.pucrs.br

System of Systems (SoS) [5] is a natural way of thinking about the modelling of these current ideas on the integration of different smart applications, in which heterogeneous systems (smart applications) are modelled as subsystems, which cooperate to achieve the higher purposes of the entire system (e.g., a global IoT). Multi-Agent Systems (MAS) [6] provide an interesting paradigm to implement SoS models; in particular, MAS provides a suitable approach to implement smart applications in IoT[2] [7].

In this particular piece of work, we are interested in the communication interface that implements data access control between different smart applications, which is currently a challenge in the integration of different systems, mainly because of the uncertainty implicit in the information used during this decision-making process [1,8]. In that respect, we propose the modelling of data access control interfaces using argumentation-based agents. Also, we propose two argumentation schemes (i.e., reasoning patterns) for data access control. These reasoning patterns take into consideration the most relevant models for data access control in the literature [9,10,11], generalising the reasoning an agent needs to carry out when treating data access requests. Guided by the critical questions in the schemes, agents are able to deal with the uncertainty in the information used during this decision-making process. Furthermore, using our approach to data access interfaces using argumentation-based agents, agents are able to understand why a request has been denied to it. Understanding this answer allows agents to provide additional information in order to be correctly categorised as well as to expose emergency situations in which emergency access control rules may apply.

2. Access Control Models

In [9], the author proposes the Category-Based Access Control (CBAC) meta-model, in which a category is any class or group to which entities are designated. The CBAC model is defined as a countable set of categories $\mathcal{C}$; a set of principals $\mathcal{P}$ (the entities that are able to require access to resources); a set of actions $\mathcal{A}$; a set of resources $\mathcal{R}$; a set of possible *answers* to access request $Auth$, and a set $\mathcal{S}$ of *situations identifiers* to denote environment information, which are application dependent [9], for instance representing time instants, system state, external state, etc.

The CBAC meta-model defines the following relations [9]: (i) *principal-category assignment*: $\mathcal{PCA} \subseteq \mathcal{P} \times \mathcal{C}$, such that $(p, c) \in \mathcal{PCA}$ iff a principal $p \in \mathcal{P}$ is assigned to category $c \in \mathcal{C}$; (ii) *permission-category assignment:* $\mathcal{ARCA} \subseteq \mathcal{A} \times \mathcal{R} \times \mathcal{C}$, such that $(a, r, c) \in \mathcal{ARCA}$ iff action $a \in \mathcal{A}$ on resource $r \in \mathcal{R}$ can be performed by principals assigned to category $c \in \mathcal{C}$; (iii) *authorisations*: $\mathcal{PAR} \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{R}$, such that $(p, a, r) \in \mathcal{PAR}$ iff a principal $p \in \mathcal{P}$ can perform action $a \in \mathcal{A}$ on resource $r \in \mathcal{R}$. Thus, a principal p is authorised to perform action a on a resource r only if p belongs to a category c such that for some category below c in the hierarchy (and including c itself) action a on r is authorised, otherwise the request is `denied`. The general idea is summarised by the following rule, in which `subc(c',c)` checks whether `c'` is a subcategory of `c`:

$$(p, a, r) \in \mathcal{PAR} \Leftarrow (p, c) \in \mathcal{PCA} \wedge subc(c', c) \wedge (a, r, c') \in \mathcal{ARCA}$$

[2]We refer the reader to the IoA (*Internet of Agents*) workshop series for an overview about this topic `http://ioa.alqithami.com/`.

In [11], the authors extend the CBAC model in order to consider policy composition, where an access control policy is combined with an emergency policy that specifies how various emergency situations affect the rights of users to access resources. Thus, two access control policies π_1 and π_2 are considered. Policy π_1 describes the usual access control rules, and π_2 the emergency policy, describing emergency situations $\mathtt{s}$ in which access may be granted for a particular category of principals $\mathtt{c}$.

$$(p, a, r) \in \mathcal{PAR} \Leftarrow (p, c) \in \mathcal{PCA} \wedge \mathtt{subc}(\mathtt{c}', \mathtt{c}) \wedge (a, r, c') \in \mathcal{ARCA}_{\pi_1}$$
$$(p, a, r) \in \mathcal{PAR} \Leftarrow (p, c) \in \mathcal{PCA} \wedge \mathtt{subc}(\mathtt{c}, \mathtt{c}') \wedge \mathtt{emrg}(\mathtt{s}) \wedge (a, r, c') \in \mathcal{ARCA}_{\pi_2}$$

3. An Architecture for Data Access Control

In our approach to data access control, the whole system is composed of subsystems (SoS) and each subsystem is implemented as a MAS. First, we describe how we define the access control policy, considering not only the usual access control rules from [9], but also considering the idea of emergency access control rules from [11]. Afterwards, we describe how we structured the hierarchies of categories for information and agents, using not only the idea of categories from [9], but also categories of information, which allows us to use categories to classify information. Finally, we describe how external requests are treated by a subsystem, considering the access control policy and the external information available to agents dealing with the request.

3.1. Access Control Policy

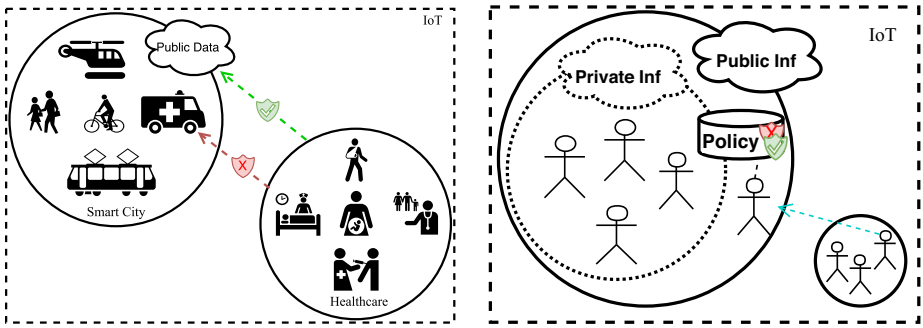
An access control policy is defined by a set of access control rules, which are specified according to the application needs[3]. Here, access control rules specify which categories of agents have access to which categories of information. Thus, we use two distinct kinds of categories in this work. The first, *access-category*, is the usual concept for categories in CBAC [9], describing categories of principals, here agents, which request access to resources, here information. The second, *information-category*, is used for categories of information. Using information-categories not only allows us to group information of similar privacy settings but also allows us to describe hierarchies of information categories. Both characteristics provide a semantic description of the information belonging to each category, which is essential when applying argumentation-based techniques in such domains. Besides the access control rules from CBAC [9], we also consider emergency access control rules from [11] as part of the access control policy. Thus, the resulting access control rules have the following format:

$$(\textit{access-category}(r_i) \xRightarrow{\texttt{access}} \textit{information-category}(c_i))$$
$$(\textit{access-category}(r_i) \wedge \textit{emergency}(s_i) \xRightarrow{\texttt{access}} \textit{information-category}(c_i)).$$

meaning that "access-category r_i has access to information-category c_i" and "during an emergency situation s_i, access-category r_i has access to information-category c_i", respectively.

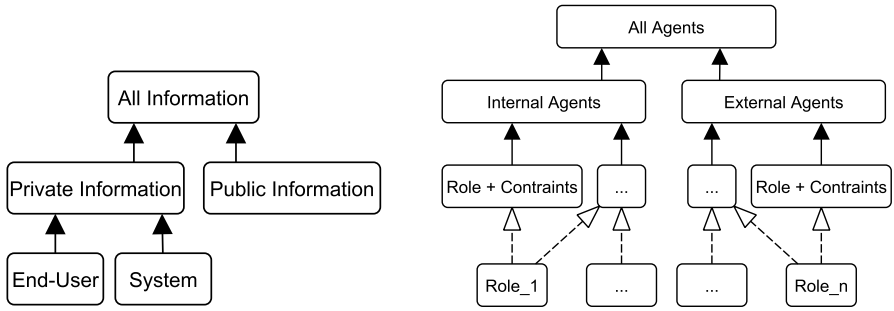
While the assignment of information to an information-category and the access control rules are internally defined in the multi-agent system that owns the information using only information from the system itself, external requests will require additional in-

[3]Normally the access control policy is specified during the design of the system [12].



(a) Data Access Between Different Applications. (b) Interface Agents for Data Access Control.

Figure 1. An Architecture for Data Access Control.



(a) *Information-Category* Hierarchy. (b) *Access-Category* Hierarchy.

Figure 2. A Structure for Categories in Data Access Control.

formation from the requester in order to categorise it according to the local definition of access-category and, consequently, to support decisions on whether to grant access to the information or not. Each category in access-category is defined by a set of constraints that the requester needs to satisfy in order to be so categorised. Constraints can include different characteristics of the requester, such as its role and reliability, as well as characteristics of its environment, organisation, domain, and so forth [9].

3.2. A Structure for Categories

When considering different smart applications, within the IoT domain for example, we are able to think of the information generated by each application as being categorised into two major categories, *public* and *private* information. Public information is the contribution that a subsystem makes to the whole system, resulting from the processing of other information in order to avoid disseminating private information, e.g., about the end-users and/or the system. Figure 1(a) illustrates this idea, in which *public* data are available for external requests, but *private* data (e.g., the location of a vehicle in a smart city) are not available for external requests. In summary, all applications might have access to public data, and access control policies may have an access control rule that provides access to *public* information for all agents in the system. Differently, *private* information may be granted access only to an access-category for which the access to such information is specified in the access control policy, i.e., there is an access control rule granting access for the access-category of the requester to the information-category of that information.

In Figure 2(a), we show a hierarchy of information-category. We consider that information is either *private* or *public*. Also, private information is categorised as end-user and system information. Note that an access control rule that allows members of a particular access-category to access the *private* information-category also grants them access to the information-categories *end-user* and *system*.

In Figure 2(b), we show an access-category hierarchy. At the root, we have the general access-category of all agents, which includes both subcategories of *internal* and *external* agents, representing agents playing some role within that subsystem, and external agents which do not play any role in that subsystem. For both *internal* and *external* agents, a number of subcategories can be defined, based on the different roles considered and other constraints required for each category in that system. Thus, an agent can be categorised in more than one access-category, and two agents playing the same role may not be categorised in the same access-category, given that the role is only part of the definition of the categories. Note that a subcategory inherits granted access from superior categories, so if there is a rule granting access for the access-category *all agents* to the information-category *public information*, that means it grants access for access-category *external agents* to *public information* too.

3.3. Interface Agents for Data Access Control

In SoS, heterogeneous systems cooperate to achieve a higher purpose [5]. During the development of SoS, communication interfaces are considered one of the most complex and difficult tasks [5]. When implementing SoS using the MAS paradigm, the communication interface with external systems is implemented using specialised agents. Thus, a communication interface for data access control corresponds to agents that are responsible for making a decision about sharing or not the information according to who is the requester and the access control policy of that particular system. Figure 1(b) illustrates interface agents for data access control.

When the agent responsible for the communication interface receives a request from an external agent a_1 to access information i_1, it carries out the argumentation-based reasoning process we will describe in the next section in order to construct an acceptable argument that grants access for a_1 to i_1, considering the access control policy (the information-category of i_1, the access control rules, and the access-category constraints). When this agent is able to construct an acceptable argument granting access for requester a_1 to information i_1, that information access request is granted, otherwise it is denied.

4. Argumentation Schemes for Data Access Control

Based on the meta-model introduced in [9], as well as the emergency policies introduced in [11], we introduce an Argumentation Scheme for Data Access Control named AS4DAC.

> **[premise]** Information I has security information-category C. **[premise]** Agent A belongs to an access-category R which has access to information with security information-category C. **[conclusion]** Agent A has access to information I.

This conclusion is reached unless the answer to any of the following questions is *no*:

CQ1 Does information I belong to information-category C?

CQ2 Does agent A belong to access-category R?

CQ3 Is there an access control rule that grants access for R to I?
CQ4 Is this conclusion free from conflict with any other information-category C_i to which information I is also assigned, which is not a super-category of C, and for which there is no access control rule that grants access for R to C_i?
CQ5 Is this conclusion free from conflict with any other access-category R_i to which agent A is also allocated, which is not a super-category of R, and for which there is no access control rule that grants access to C?
CQ6 In the case where this conclusion is based on an emergency access control rule that grants access for R to C during an emergency situation S_i, is S_i the case?

External requests will require additional information from the requester in order to categorise it according to the internal definitions of access-category and, consequently, to make a decision about grating access to the requested information or not. Also, given the defeasibility of such information, it is necessary to give special attention to the access-category assignment when receiving a request. Therefore, we introduce the Argumentation Schemes for Access-Category Assignment, named AS4ACA, which allows agents to investigate the assigning of a requester to an access-category in more depth.

[premise] An access-category R is defined by a set of constraints S. **[premise]** Agent A satisfies the constraints S. **[conclusion]** Agent A belongs to the access-category R.

CQ1 Does agent A satisfy all constraints s_i in the set of constraints S?
CQ2 Is R the more specific access-category for which A satisfies the constraints?

Note that not only should an agent satisfy all constraints required by a particular access-category in order to be so categorised, but it should be also categorised in the most specific access-category it satisfies the constraints, given that more specific categories inherit access from the super-categories in the hierarchy of access-categories.

4.1. Argumentation-Based Reasoning for Data Access Control

Some approaches in the argumentation literature show that argumentation schemes [13] can be translated into defeasible inferences [14,15,16], and the acceptability of arguments, instantiated using these rules, can be checked through frameworks such as ASPIC+ [17], DeLP [18], and others [19].

Definition 1 (Argumentation Scheme) *An argumentation scheme is a tuple $\langle \mathcal{SN}, \mathcal{C}, \mathcal{P}, \mathcal{CQ} \rangle$ with $\mathcal{SN}$ the argumentation scheme identifier (name), $\mathcal{C}$ the conclusion of the argumentation scheme, $\mathcal{P}$ the premises, and $\mathcal{CQ}$ the associated critical questions.*

Definition 2 (Argument) *An argument is a tuple $\langle S, c \rangle$, with S the set of premises and inference rules of the scheme used to draw c (the conclusion of the argument). That is, S includes all instantiated premises from $\mathcal{P}$ — i.e., considering a most general unifier θ, for all $p \in \mathcal{P}, p\theta \in \mathcal{S}$ — and the inference rule corresponding to the scheme ($\mathcal{P} \Rightarrow \mathcal{C}$); the conclusion c is the instantiation $\mathcal{C}\theta$ such that $S \models c$.*

An example of argument, instantiated from AS4DAC, is given below, with $\theta = \{\mathtt{I} \mapsto \mathtt{i}_1, \mathtt{C} \mapsto \mathtt{c}_1, \mathtt{A} \mapsto \mathtt{a}_1, \mathtt{R} \mapsto \mathtt{r}_1\}$:

```
⟨ {inf_category(i1,c1), ac_category(a1,r1), access(r1,c1),
  [inf_category(I,C), ac_category(A,R), access(R,C) ⇒ access(A,I)]},
  access(a1,i1) ⟩
```

Considering our approach to data access control, an agent may grant access for a requester a_1 to information i_1 only if there is an *acceptable* argument concluding $\texttt{access}(a_1, i_1)$, considering the existence of an access control rule that grants access for the access-category of a_1 to the information-category of i_1.

Definition 3 (Acceptable Arguments) *An argument $\langle S, c\rangle$, instantiated from an argumentation scheme $\mathcal{SN}$, is acceptable to an agent ag (where $\Delta_{\texttt{ag}}$ is its knowledge base) iff: (i) all premises in S are acceptable to ag, i.e., $\forall p\theta \in \mathcal{S}, \Delta_{ag} \models p\theta$, either because p is asserted in its knowledge base, or because p is the conclusion of an acceptable argument; and (ii) all critical question related to the argumentation scheme $\langle \mathcal{SN}, \mathcal{C}, \mathcal{P}, \mathcal{CQ}\rangle$ are positively answered by ag, i.e., $\forall Cq_i \in \mathcal{CQ}, \Delta_{ag} \models Cq_i\theta$.*

Considering our example above, that argument is acceptable to an agent ag when ("$\neg$" represents strong negation and "not" negation as failure): (i) $\Delta_{ag} \models \texttt{inf\_category}(i_1, c_1)$, which is asserted in Δ_{ag}; (ii) $\Delta_{ag} \models \texttt{ac\_category}(a_1, r_1)$, which requires ag to instantiate an acceptable argument from the argumentation scheme AS4ACA; (iii) $\Delta_{ag} \models \texttt{access}(r_1, c_1)$, which is asserted in Δ_{ag}; and (iv) all critical questions are also positively answered by ag: $\Delta_{ag} \models \texttt{inf\_category}(i_1, c_1)$, $\Delta_{ag} \models \texttt{ac\_category}(a_1, r_1)$, $\Delta_{ag} \models \texttt{access}(r_1, c_1)$, $\Delta_{ag} \models \{\texttt{not}(\texttt{inf\_category}(i_1, c_2), \neg\texttt{subc}(c_1, c_2), \neg\texttt{access}(r_1, c_2))\}$, $\Delta_{ag} \models \{\texttt{not}(\texttt{ac\_category}(a_1, r_2), \neg\texttt{subc}(r_1, r_2), \neg\texttt{access}(r_2, c_1))\}$, $\Delta_{ag} \models \{\texttt{not}(\texttt{emrg}(s_1, \texttt{access}(r_1, c_1)), \neg s_1))\}$.

Note that an agent will always be able to categorise an information i_1 to an information-category and a requester a_1 to an access-category[4], given the hierarchy of categories defined in Figure 2. Thus, when an access to i_1 is denied, i.e., the agent is not able to construct an acceptable argument for $\texttt{access}(a_1, i_1)$, that means (i) there is no access control rule granting access for the access-category of a_1 to the information-category of i_1 (**CQ3** in AS4DAC); (ii) there exist a counter-example for $\texttt{access}(\texttt{a}_1\texttt{,i}_1)$ (**CQ4** and **CQ5** in AS4DAC); or, (iii) the emergency situation considered, if any, is not true (**CQ6** in AS4DAC). In all cases, the agent denies access for a_1 to i_1 with the following argument:

$\langle$ {inf_category(i$_1$,c$_1$), ac_category(a$_1$,r$_1$), ¬access(r$_1$,c$_1$),
[inf_category(I,C), ac_category(A,R), ¬access(R,C) ⇒ ¬access(A,I)]},
¬access(a$_1$,i$_1$) $\rangle$

When an agent a_1 receives this argument, it is able to respond with additional information. This information may clarify to the agent dealing with the request the correct access-category to categorise a_1, answering differently **CQ2** in AS4DAC (i.e., the conclusion of AS4CAC) and, possibly, answering positively **CQ3**, **CQ4**, and **CQ5** in AS4DAC. Also, a_1 may provide information about an emergency situation that may grant access to this information, considering an emergency access control rule.

5. Final Remarks

Our approach was built based on: (i) the models for data access control from [9,11]; (ii) approaches that apply argumentation-based techniques considering the specification of argumentation schemes in their conception, e.g., [20,21]; (iii) the problem of data access

[4]However, this categorisation may change as the agent acquires more information about a_1.

control, e.g., [1,4]; and (iv) structured argumentation frameworks/work that suggest the representation of argumentation schemes using a logic with similar notation to agent-oriented programming languages, which can be naturally interpreted and manipulated by agents, for example [14,16].

To the best of our knowledge, our work is the first to propose argumentation schemes for data access control. We specified those reasoning patterns considering characteristics of some of the most relevant current models for data access control, providing a general approach for reasoning about data access control. Our work differs from the literature particularly on its generality. Using our approach, agents are not only able to carry out a detailed reasoning process in order to deal with the uncertainty about the information used to make a decision on granting access to requests, but also to communicate additional information to confront the information used to categorise agents, as well as to expose emergency situations in which emergency access control rules may apply.

References

[1] E. Karafili, K. Spanaki, and E.C. Lupu. An argumentation reasoning approach for data processing. *Computers in Industry*, 94:52–61, 2018.

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things* 3(3):34–36, 2010.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[4] K. Zhao and L. Ge. A survey on the internet of things security. In *Computational Intelligence and Security (CIS)*, pgs 663–667, 2013.

[5] G. M. Puppi Wanderley, M. Abel, J. Barthès, and E. C. Paraiso. A core architecture for developing systems of systems. In *Int. Conf. on Systems, Man, and Cybernetics*, 2017.

[6] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[7] V. Gorodetsky. Internet of agents: From set of autonomous agents to network object. *IoA'17*, 2017.

[8] E. Karafili, A. C. Kakas, N. I. Spanoudakis, and E. C. Lupu. Argumentation-based security for social good. In *AAAI*, 2017.

[9] S. Barker. The next 700 access control models or a unifying meta-model? In *ACM symposium on Access control models and technologies*, pgs 187–196, 2009.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[11] S. Alves and M. Fernández. A framework for the analysis of access control policies with emergency management. *ENTCS*, 312:89–105, 2015.

[12] A. Ali and M. Fernández. A programming language with role-based access control. *Proc. of URC* 2010, UG Research in Computer Science - Theory and Applications*, London, IFCOLOG, 2010.

[13] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[14] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2011.

[15] A. R. Panisson and R. H. Bordini. Uttering only what is needed: Enthymemes in multi-agent systems. In *AAMAS*, pgs 1670–1672, 2017.

[16] A. R. Panisson and R. H. Bordini. Argumentation schemes in multi-agent systems: A social perspective. In *Int. Ws. on Engineering Multi-Agent Systems*, 2017.

[17] S. Modgil and H. Prakken. The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[18] A. J. García and G. R. Simari. Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. *Argument & Computation*, 5(1):63–88, 2014.

[19] A. R. Panisson and R. H. Bordini. Knowledge representation for argumentation in agent-oriented programming languages. In *Brazilian Conference on Intelligent Systems*, 2016.

[20] P. Tolchinsky, U. Cortés, J. C. Nieves, A. López-Navidad, and F. Caballero. Using arguing agents to increase the human organ pool for transplantation. In *Ws. on Agents Applied in Health Care*, 2005.

[21] S. Modgil, P. Tolchinsky, and U. Cortés. Towards formalising agent argumentation over the viability of human organs for transplantation. In *MICAI*, pgs 928–938, 2005.