Modeling Data Access Legislation with Gorgias

Nikolaos I. Spanoudakis[1], Elena Constantinou[2], Adamos Koumi[2], and Antonis C. Kakas[2]

[1] Applied Mathematics and Computers Lab., Technical University of Crete, Greece
[2] Department of Computer Science, University of Cyprus, Cyprus

Abstract. This paper uses argumentation as the basis for modeling and implementing the relevant legislation of an EU country relating to medical data access. Users can consult a web application for determining their allowed level of access to a patient's medical record and are offered an explanation based on the relevant legislation. The system can also advise a user on what additional information is required for a higher access level. The system is currently in the process of an extensive evaluation through a pilot trial with a special focus group of medical professionals. The development methodology that we have used is generally applicable to any other similar cases of decision making based on legislative regulations. The main advantage of using argumentation is the ability to explain the solutions drawn and the high modularity of software facilitating the extension and adaptation of the system when new relevant legislation becomes available.

Keywords: Argumentation, Legal Systems, Modular Software

1 Introduction

Modern systems aim to automate compliance to laws, policies (or business rules) and regulations. In many cases the problem would involve several of such policies to be applied together creating the need for internal coherence amongst the different policies of the integrated system. The main challenge in building such systems is to develop software that is close to the high-level specification of the policies involved so that (1) the information can be easily acquired and faithfully represented, and, (2) changes in the policies could be easily propagated to the software. The resulting software should also be able to provide information explaining why a particular case is compliant or not, how its compliance is affected by the various policies involved and how new information about the case at hand can change the degree of compliance.

A particular case of the problem of policy compliance is that of *data sharing*. In such problems data may belong and be private to a particular owner or institution but, yet, it is often necessary to share (at least part of) this data. Data sharing agreements are enforced when the data is used to identify if a user/application is granted access to the data and at what level of access. The problem of such data access and usage control is well studied [16, 11, 18], but

existing solutions are restricted in allowing conflicting rules, together with a solution to the conflicts. Recent projects like *CoCo Cloud*[3] aim to automated data sharing activities by analyzing the various policies involved in order to identify possible conflicts and then propose algorithms for conflict resolution.

An important case of data sharing is that of *accessing a patient's medical data* where, although the data belongs to the patient, it is necessary for doctors or other medical staff to access parts of this data when the patient needs treatment. The decision of what data can be shared should follow legal regulations that pertain on the one hand to the general data protection and privacy rights of individuals and on the other hand to rights and obligations that are specific to medical data.

In this paper we study the problem of medical data access as specified by the relevant European Union and national regulations in one of its member states (Cyprus). These regulations are modeled in terms of argumentation drawing from the theory and practice of argumentation in Artificial Intelligence (see e.g. [8, 5, 17]). Compliance of access with respect to the regulations is thus mapped into a decision problem of what level of access has, according to the argumentation theory that models the legislation, an acceptable argument that supports the option to grant this level of access. Arguments that support different levels of access dialectically compete with each other and only the stronger argument(s) are used to grant access. Our approach follows a long tradition of linking argumentation in AI with Law (see [15, 6, 14] for reviews) but where the emphasis is on the development of a practical system for a relatively simple, yet real-life, piece of legislation.

In contrast with conventional approaches to data sharing the approach through argumentation is not based on a procedural analysis for finding and resolving conflicts but on a high-level declarative representation of the policies themselves. Through a systematic evaluation that we are currently carrying out using an appropriate focus group for this real-life application of medical data access we aim to examine and understand the possible added value of argumentation for this type of applications.

The next section presents the legislation that regulates medical data access and analyses this in a suitable way for our model. In section 3 we briefly review the argumentation framework and the methodology we will use in modeling the problem. Section 4 shows how the legislation is mapped into argumentation and how the application system is build. The final section concludes with our plans for future work of more extensive evaluation and the development of other similar applications.

2 Legal framework for Medical Data Access

In this section we will present the legal framework that we aim to model. For this we had to consider two law documents, one for personal data protection [2]

[3] http://rissgroup.org/coco-cloud-confidential-and-compliant-clouds/

and one on the rights of the patients [1]. We will start by defining some domain knowledge that will aid in the development of the argumentation theory. Then we will present the different users and types of access. Finally we will outline the policies defined in the legal framework.

2.1 Definitions

We will start by defining what is/constitutes a *medical record*. The medical record contains data related to the mental and physical health of the owner in the past, the present, and, sometimes, the future. Specifically, it contains:

- *Demographic data*, used to identify the owner, e.g. name, surname, date of birth, telephone number, address, identity and social security numbers.
- *Socioeconomic data*, personal data, such as marital status, profession, employer, religion, nationality, personal habits (e.g. smoking).
- *Clinical data*, such as illnesses endured, lab tests, x-rays, drug prescriptions, surgeries, temperature and blood pressure readings.

The type of access to the medical record depends on the following concepts:

- *Patient*: An individual that requests/receives medical service.
- *Medical service provider*: Medical doctor, pharmacist, dentist, nurse, obstetrician, paramedical or administrative staff working for a medical institution.
- *Personal data*: Any information related to an individual whose identity is known or can be established.
- *Data Processing and archiving*: Any series of activities applied to medical or personal data, including: *collection, modification, storage, transformation, retrieval, search, use, transfer, copy, encryption, deletion or destruction.*
- *Medical data*: Information about the health of an individual, also information in close connection with the medical domain.
- *Medical files*: Files produced by a medical service provider in printed or digital form related to the health of an individual, containing information that can be used to establish the identity of the individual.
- *Third party*: A legal or physical entity, public authority, service or any other body other than the person to whom the data refers.
- *Legal representative*: An individual hired to perform an action in place of someone else or to represent someone in a transaction with a third party.
- *Consent*: The owner of the personal data gives clearly and in full knowledge consent for their processing.
- *Controller*: Decides on the purpose and means of processing a data file.

The users of the medical data are those with the right to process them. They are expected to be medical doctors, nurses, paramedical and administrative personnel of state-owned or private medical institutions and hospitals. The administrative personnel can also use the system aiming to provide access to a patient, the patients family or a legal representative.

Before the user can access a medical record of a patient, he is expected to establish his/her identity and explain the circumstances under which he/she is requesting access. There are several types of access granted to a specific user:

- *Full access*: The user can add, remove data of a medical record. The user can access all the medical files in the record and the personal data.
- *Limited plus access*: Limited plus access aims to allow access to data for determining the general status of the owners health without much detail but allowing a good diagnosis and drug prescription. The user can have limited access to the medical files, i.e. those related to the current treatment of the owner and to personal data. The user can access information related to the allergies of the owner, chronic diseases and medication received. The user can add a medical file related to the current treatment of the owner.
- *Limited plus, read-only access*: Same as previous, with the exception that the user cannot add a medical record.
- *Limited access*: The user can have limited access to the medical files related to the medical history of the owner with respect to a specific therapy followed in the past. Specifically, the user can access information relating to the treating medical personnel, the diagnosis, medication received, results of clinical examinations and the resulting conclusions.
- *Suspended access*: This type of access is only valid for the owner of the medical record. Access to specific data is refused for a specific time-span determined by a medical doctor (who has determined that the patient must not know yet a specific issue regarding his/her health because this might be a hazard for his/her health).
- *No access*: No information is disclosed to the user.

2.2 Policies for Determining Access Type

The access type depends on three main contexts. Firstly it depends on *who is asking to get access* to the medical record. According to that we have the following types of users and **default** access types:

- {owner} → full access
- {family doctor} → full access
- {doctor} → limited plus access
- {family member} → limited plus access
- {legal representative person} → full access
- {patient involved to owners treatment} → limited plus read only access
- {person holding order from the high court} → limited plus read only access
- {other person} → no access

Then, access type depends on *the purpose of asking for access* (that the user must disclose along with his/her identity) posing limitations:

- {research purpose} → limited access
- {processing purpose} → limited plus access
- {for publishing purposes in medical journals} → limited access
- {treatment purpose} → limited plus read only access
- {teaching purpose} → limited read only access
- {order from the Medical Association} → limited plus read only access

Thirdly, access type depends on other *specific circumstances* :

- {written consent from owner} → full access
- {owner is dead} → no access
- {owner, doctor restriction} → suspended access

These factors are considered together, generally from the first (person asking) to the third (circumstances).

3 Argumentation Theory for Policy Applications

In this section we review the basic theory of argumentation which we will use to model regulation and other policies. The theory will be presented from a general point of view of applying argumentation to real-life compliance problems viewed as decision problems under an argumentation policy. We will also overview the *Gorgias* system as an environment for developing applications of argumentation and on which our case study of medical data access will be based.

Policies will be represented within the preference-based argumentation framework proposed in [9]. In this, application problems are captured via argumentation theories composed of different levels. **Object level arguments support** the possible decisions, or **options**, in a specific application domain, while **first-level priority arguments** express preferences on the object level arguments in order to resolve possible conflicts. **Higher-order priority arguments** are also used to resolve potential conflicts between priority arguments of the first (or subsequent) levels.

Formally, an **argumentation theory** is a pair $(\mathcal{T}, \mathcal{P})$ whose sentences are formulae in the background monotonic logic, $(\mathcal{L}, \vdash)$, of the form $L \leftarrow L_1, \ldots, L_n$, where $L, L_1, \ldots, L_n$ are positive or negative ground literals. The derivability relation, $\vdash$, is given simply by the inference rule of modus ponens. The head literal L can also be empty. Rules in $\mathcal{T}$ capture argument schemes for building **object level arguments**, or denials when the head is empty. On the other hand, rules in $\mathcal{P}$ represent argument schemes for building **priority arguments**. The head L of these rules has the general form, $L = h\_p(rule1, rule2)$, where $rule1$ and $rule2$ are atoms naming two rules and h_p refers to an (irreflexive) *higher priority* relation amongst the rules of the theory.

The semantics of an argumentation theory is defined via an abstract argumentation framework $< Args, Att >$ associated to any given theory $(\mathcal{T}, \mathcal{P})$. The **arguments** in $Args$ are given by the composite subsets, (T, P) , of the given theory, where $T \subseteq \mathcal{T}$ and $P \subseteq \mathcal{P}$. An argument (T, P) **supports** its conclusions, of either a literal, L, or a priority (ground) atom, $h\_p(r, r')$, where r and r' are the names of two rules in the theory, when $T \vdash L$ or $T \cup P \vdash h\_p(r, r')$.

The **attack relation**, Att, allows an argument, (T, P), to attack another argument, (T', P'), when (i) these arguments derive contrary conclusions (i.e. derive L and $\neg L$, or $h\_p(r, r')$ and $h\_p(r', r)$) and (ii) (T, P) makes the rules of its counter proof at least "as strong" as the rules of the proof of the argument (T', P') that is attacked. The detailed formal definition of the attacking relation

can be found in [9]. The **admissibility** of (sets of) arguments, Δ, is defined in the usual way [8], i.e. that Δ does not attack itself and that it attacks back any argument that attacks it.

It is important to note that typically for an argument (T, P) to be *admissible* its object level part, T, has to have along with it priority arguments, P (from $\mathcal{P}$), in order to make itself at least "as strong" as its opposing counter-arguments. This need for priority rules can repeat itself when the initially chosen ones can themselves be attacked by opposing priority rules. In that case, higher-order priority rules need to be used to make these priority rules at least "as strong" as their opposing priority ones.

The multi-layered nature of an argumentation theory $(\mathcal{T}, \mathcal{P})$ and the process of deciding on the admissibility arguments mirror the structure of the legislation and the process of legal reasoning to decide on a valid legal position. Basic articles of the law give the information for object-level rules, in the argumentation theory, for general default decisions while articles describing contextual-based exceptional decisions are captured via the priority rules of the theory.

The above theoretical framework of argumentation has been implemented in the open source *Gorgias* system (http://www.cs.ucy.ac.cy/ñkd/gorgias/). *Gorgias* has been, since 2004, successfully applied by different users for developing real life applications (see e.g. portfolio management [13], provision of services in ambient intelligence [12], management of firewall policies [3], conflicts resolution in pervasive services [4]) (see http://gorgiasb.tuc.gr/Apps.html for a list of applications).

Based on this experience a new software methodology [19] and tool to support this, called *Gorgias-B* (http://gorgiasb.tuc.gr), has been recently developed so that such applications of argumentation can be developed in a systematic and principled way. The proposed *"Software Development for Argumentation"(SoDA)* methodology aims to provide a general software development framework that can be used by application domain experts, with little or no knowledge of argumentation theory, to develop application software based on argumentation. The methodology guides the developer through his/her application problem by an incremental refinement of application scenarios, where he/she considers the several (usually conflicting) alternatives (e.g., different diagnostic results, judicial decisions, recommendation options, risk management decisions, etc) and evaluates them according to some criteria/features of the problem and context dependent (meta)-knowledge. The *Gorgias-B* tool helps the developer to consider his/her application according to the *SoDA* methodology and offers a high-level environment through which the software code of the underlying argumentation theory is automatically generated. The *Gorgias-B* system also supports *abductive reasoning* integrated with argumentation [7, 10] thus enabling the possibility for solving, using the same application software, *reverse decision problems* of identifying extra information needed to make a certain desired decision possible, i.e. supported by an admissible argument.

4 Medical Data Access System

In this section we will describe how we have modeled the legislation for medical data access using *Gorgias* according to the *SoDA* methodology and give the high-level architecture of the developed application system.

4.1 Decision Policy Development

The first task is to model the options of the problem, i.e. the different types of access (see section 2.2). We use the option predicate *access(User, Data, Level)* where the first parameter denotes the user, the second the data (or file) asking for permission and the third the permission type for this data. The next task is to define the contextual hierarchy from the most general to the most specific, of the various application scenarios. In our case (see section 2) this is given by:

1. Person requesting access
2. Purpose of access
3. Special circumstances

We then proceed to define the Gorgias rules that defined the argumentation theory starting from general scenarios and considering refinements of these. For example these object rules will be generated for two different types of access:

$r_1(P,F,T) : access(P,F,no\_access) \leftarrow true$
$r_2(P,F,T) : access(P,F,limited\_plus\_access) \leftarrow true$

According to the *SoDA* methodology, we consider conflicting options in pairs. For the pair *no_access* and *limited_plus_access* we have at the second level two possibilities. One defaulting to the *no_access* captured by the rule $c^2_{1,2}$ and one selecting *limited_plus_access* when the person requesting it is a medical doctor, captured by $c^2_{2,1}$:

$c^2_{1,2}(P,F,T) : h\_p(r_1(P,F,T), r_2(P,F,T)) \leftarrow true$
$c^2_{2,1}(P,F,T) : h\_p(r_2(P,F,T), r_1(P,F,T)) \leftarrow doctor(P)$

At a higher-level of priority we capture that $c^2_{1,2}$ is generally stronger, and we move to consider the purpose of requesting access. If a doctor wants access for medical purpose then the *limited_plus_access* is granted:

$c^3_{1,2}(P,F,T) : h\_p(c^2_{1,2}(P,F,T), c^2_{2,1}(P,F,T)) \leftarrow true$
$c^3_{2,1}(P,F,T) : h\_p(c^2_{2,1}(P,F,T), c^2_{1,2}(P,F,T)) \leftarrow medical$

At yet a higher-level of priority we capture that $c^3_{2,1}$ is stronger and that it forms the default at this higher level. However, in the special circumstance that the owner is dead, this priority is again reversed:

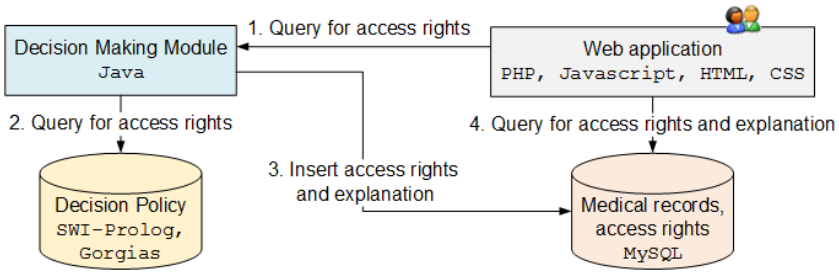

Fig. 1. The ***Medical Data Access Control*** application's architecture

$c^4_{2,1}(P,F,T) : h\_p(c^3_{2,1}(P,F,T), c^3_{1,2}(P,F,T)) \leftarrow true$
$c^4_{1,2}(P,F,T) : h\_p(c^3_{1,2}(P,F,T), c^3_{2,1}(P,F,T)) \leftarrow owner(X,F), dead(X)$

Finally, at a fifth level of priority we capture that $c^4_{2,1}$ is generally stronger:

$c^5_{1,2}(P,F,T) : h\_p(c^4_{1,2}(P,F,T), c^4_{2,1}(P,F,T)) \leftarrow true$

When all such pairs have been considered for all possible ranked contexts the argumentation theory is ready and the *Gorgias-B* tool automatically generates the *Gorgias* Prolog source code.

4.2 System Design/Architecture

The Prolog source code is invoked by a Java module for getting the facts for any situation for which we want to find the access rights. This Java module is used by a web application built using standard HTML/CSS and PHP technologies[4]. The architecture of our system is depicted in Figure 1 where the numbers on the arrows show the sequence of execution for each user query.

A user typically logs in and uses a form to request for the access rights to a patient's record. The Java module then writes the result of the Prolog query to a database that is used by the web application along with the explanation in predicate form. In the database the predicates are mapped to legislation articles and paragraphs (e.g. the request by a doctor for medical reasons will respond with *limited_plus_access* rights based on article 15, paragraph 2b of [1]) so that a user-understandable text is shown to the user by the web-application.

A user can also use the system to ask what circumstances should hold so that he/she can have a different access level. To support this query, facts are defined as abducibles, i.e. unless the system is informed otherwise, they can be assumed as true, and the system can reply with the possible context for such use. For example if a doctor wants to access a file he gets *no_access*. However, if he asks whether he could get *limited_plus_access* rights he will get the answer that he needs to have a *medical* reason.

[4] The MEDICA web application has been deployed at: http://medica.cs.ucy.ac.cy

4.3 First Evaluation of System

The developed system is the result of Elena Constantinou's diploma thesis for her Computer Science B.Sc. degree. A first evaluation with (23) classmates was carried out, aiming to determine how easy it was to learn and use it. Through a questionnaire, 78% of the students agreed that the system is easy to learn, 70% agreed that the system menus and functionality is well designed. Moreover, through this survey a number of ideas to enhance the system were suggested.

We are currently in the process of evaluating the system with a focus group of specialists to assess the applicability of the system for usage in hospitals and health centers. We are working closely with a team of medical informatics which advises the government of Cyprus on IT systems for the national health service. This team will first evaluate our system from their own IT perspective and then proceed for an evaluation through pilot trial at appropriate medical centers.

5 Conclusions

We have shown how the technology of argumentation can be used to model and implement the real-life legal regulations pertaining to access to patient data. Using the high-level declarative approach of argumentation we can develop in a principled way application software that is modular and flexible in accommodating changes in the problem requirements. In contrast to a carefully crafted set of rules for capturing the legislation, argumentation provides a direct mapping of the legislation where the representation of one part of legislation does not need to explicitly safeguard against the other parts of legislation that might be in conflict with this. We claim that this results in application software where the effort required to update the software to changes in the legislation is comparable to that of changing the old legislation document to the new one.

The direct representation of the legislation is particularly facilitated by the simplicity of the representation language of the *Gorgias* framework (not found in other structured argumentation frameworks). This simplicity allows the developer to follow the *SoDA* methodology where s/he does not need to consider the application at the lower-level of the structured argumentation representation language but rather only at the higher level of application scenarios and the possible relative preferred decisions available in the different scenarios.

Apart from a more extensive evaluation of the system, including the strengthening of confidence in its legal correctness, we are working to provide a natural dialogue interface with the user for explaining and guiding as to the available levels of access. We are also considering, in collaboration with the *RISS* group at Imperial College, other real-life applications of data sharing where a more heterogeneous set of policies is involved such as business policies, shared party agreements, as well as national and international legal regulations.

Acknowledgements: We thank the *RISS* group at Imperial College for useful discussions.

References

1. Cyprus law on patient rights, 1(I)/2005
2. Cyprus law on personal data protection, 138(I)/2001
3. Bandara, A.K., Kakas, A.C., Lupu, E.C., Russo, A.: Using argumentation logic for firewall configuration management. In: Integrated Network Management, IM 2009. 11th IFIP/IEEE International Symposium on Integrated Network Management, Hofstra University, Long Island, NY, USA, June 1-5, 2009. pp. 180–187 (2009)
4. Benazzouz, Y., Boyle, D.: Negotiation and Argumentation in Multi-Agent Systems: Fundamentals, Theories, Systems and Applications, chap. Argumentation-Based Conflict Resolution in Pervasive Services, pp. 399–419. Bentham Science (2014)
5. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in Artificial Intelligence. Artificial Intelligence 171(10–15), 619–641 (2007)
6. Bench-Capon, T.J.M., Prakken, H., Sartor, G.: Argumentation in legal reasoning. In: Argumentation in Artificial Intelligence, pp. 363–382 (2009)
7. Demetriou, N., Kakas, A.C.: Argumentation with abduction. In: 4th Panhellenic Logic Symposium (PLS2003) July 7-10, Thessaloniki, Greece. pp. 38–43 (2003)
8. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person Games. Artificial Intelligence 77, 321–357 (1995)
9. Kakas, A.C., Moraitis, P.: Argumentation based decision making for autonomous agents. In: The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings. pp. 883–890 (2003)
10. Kakas, A.C., Moraitis, P.: Adaptive agent negotiation via argumentation. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006. pp. 384–391 (2006)
11. Matteucci, I., Petrocchi, M., Sbodio, M.L.: CNL4DSA: a controlled natural language for data sharing agreements. In: Proceedings of the ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26. pp. 616–620 (2010)
12. Moraitis, P., Spanoudakis, N.I.: Argumentation-based agent interaction in an ambient-intelligence context. IEEE Intelligent Systems 22(6), 84–93 (2007)
13. Pendaraki, K., Spanoudakis, N.I.: Portfolio performance and risk-based assessment of the PORTRAIT tool. Operational Research 15(3), 359–378 (2015)
14. Prakken, H., Sartor, G.: Law and Logic: a Review from an Argumentation Perspective. Artificial Intelligence 227, 214–245 (2015)
15. Prakken, H.: Ai & law, logic and argument schemes. Argumentation 19(3), 303–320 (2005)
16. Pretschner, A., Hilty, M., Basin, D., Schaefer, C., Walter, T.: Mechanisms for usage control. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security. pp. 240–244. ASIACCS '08, ACM (2008)
17. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence. Springer Publishing Company, Incorporated, 1st edn. (2009)
18. Ruiz, J.F., Petrocchi, M., Matteucci, I., Costantino, G., Gambardella, C., Manea, M., Ozdeniz, A.: A Lifecycle for Data Sharing Agreements: How it Works Out, pp. 3–20. Springer International Publishing, Cham (2016)
19. Spanoudakis, N.I., Kakas, A.C., Moraitis, P.: Applications of argumentation: The SoDA methodology. In: 22nd European Conference on Artificial Intelligence, 29 Aug.-2 Sep., The Hague, The Netherlands (ECAI 2016). pp. 1722–1723 (2016)