Applying Metalevel Argumentation Frameworks to Support Medical Decision Making

Nadin Kökciyan, Isabel Sassoon, Elizabeth Sklar, Sanjay Modgil and Simon Parsons

Abstract—People are increasingly employing artificial intelligence as the basis for decision-support systems (DSSs) to assist them in making well-informed decisions. Adoption of DSS is challenging when such systems lack support, or evidence, for justifying their recommendations. DSSs are widely applied in the medical domain, due to the complexity of the domain and the sheer volume of data that render manual processing difficult. This paper proposes a metalevel argumentation-based decision-support system that can reason with heterogeneous data (e.g. body measurements, electronic health records, clinical guidelines), while incorporating the preferences of the human beneficiaries of those decisions. The system constructs template-based explanations for the recommendations that it makes. The proposed framework has been implemented in a system to support stroke patients and its functionality has been tested in a pilot study. User feedback shows that the system can run effectively over an extended period.

Index Terms—Computational argumentation, decision-support systems, explainability, healthcare.

1 Introduction

ARTIFICIAL intelligence (AI) has great potential to assist people in making well-informed decisions. In decision-support systems (DSSs), this assistance can take the form of a set of recommendations; the final decision is then made by a human. However, this final decision-making step is not a trivial task, especially when the rationale underlying those recommendations is not transparent. Our focus is on decision making in healthcare, a domain where DSSs can usefully fulfill two roles. Firstly, DSSs can provide valuable time-saving support by highlighting key factors underlying decisions for healthcare professionals. Secondly, patients are often in need of routine information, for which DSSs can provide details supplementing treatment plans proposed by (human) clinicians. Hence, DSSs may help patients self-manage their health conditions and so contribute to improved health outcomes.

As a result of focus groups conducted with patients and healthcare professionals [1], we identified three stakeholder criteria for a DSS, which should: (1) represent information coming from heterogeneous sources (e.g. a wellness sensor gives blood pressure readings), (2) consider stakeholder preferences in its decision-making (e.g. a patient who prefers one treatment over another), and (3) provide explanations for the decisions it makes (e.g. a patient wants to know why a specific painkiller was recommended).

In Figure 1, we propose a DSS architecture (satisfying the three criteria) that can be specialized for different domains. The input data layer includes multiple information sources, which can provide a stream of data (i.e. live data), as in the case of wellness sensors; or more static data such as clinical guidelines providing advice regarding a condition or a symptom. The preferences can be integrated into the system according to specific needs of the domain in question. For example, the stakeholders can share their preferences while interacting with the system, or preferences can be embedded in the system if they are known a priori.

In this general framework, we focus on computational argumentation [2] as the reasoning mechanism. Argumentation has been applied in AI and multi-agent systems as a formal approach, where evidence is represented as *arguments* for and against particular *conclusions*. We apply metalevel argumentation frameworks (MAFs) [3] that empower systems to encode and reason with possibly conflicting rationales for preferences and attacks and with the relationships among arguments that attacks and preferences define. Thus, one can reason about information that is relevant to decisions, which is not typically accommodated in object-level approaches to argumentation. In other words, standard (object-level) argumentation frameworks include attacks and preferences, but they typically cannot be reasoned about; MAFs overcome this limitation.

All data provided by the input data layer is translated into a chosen formal language and stored as specifications. The *Meta Engine* processes the specifications to construct a MAF in support of actions while using knowledge from the *Schemes* repository. This repository provides domain-specific information such as how to build arguments, attacks or explanations. The *Solver* component evaluates the 'winning' or 'justified' arguments in the constructed MAF to make recommendations, and explanations provided by the *Explanation Generator*.

We introduce an instance of our proposed framework as part of the CONSULT[1] project. We describe this instance, and the assessment of its usability and acceptability, in the rest of the paper.

- *N. Kökciyan is with the University of Edinburgh. I. Sassoon is with Brunel University London. E. Sklar and S. Parsons are with the University of Lincoln. S. Modgil is with King's College London.*

Manuscript received XXX; revised XXX.

[1] Collaborative Mobile Decision Support for Managing Multiple Morbidities, http://consultproject.co.uk

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/MIS.2021.3051420

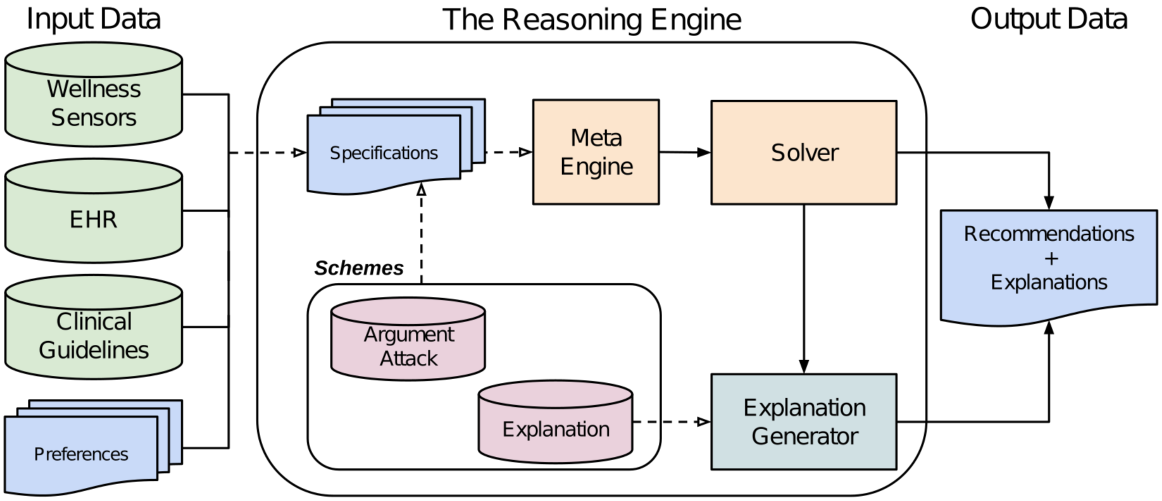


Fig. 1. A general framework for domain-specific decision-support systems. The data is provided by various information sources. The *Schemes* are templates for structuring and representing arguments, attacks and explanations. A formal language is used to encode the knowledge in terms of *Specifications*. The *Meta Engine* uses these specifications to construct a metalevel argumentation framework, and the *Solver* evaluates this framework to compute recommendations. The *Explanation Generator* constructs textual explanations for the recommendations according to explanation schemes.

2 Background

An argumentation framework (AF) is a directed graph consisting of arguments (the graph's nodes) related by attacks (the edges) denoting that one argument is a counterargument to another [4]. The resulting AF is evaluated under a chosen *semantics*, thus identifying one or more sets of acceptable (justified) arguments. In preference-based AFs (PAFs) [5], a preference ordering over arguments can then be used to identify when one argument successfully attacks (i.e. 'defeats') another; an attack succeeds if the target argument is not preferred to the attacking argument. Extended AFs (EAFs) [6] can incorporate arguments claiming preferences over other arguments, and so enable reasoning about possibly conflicting preferences. MAFs [3] formalize the idea that one can argue *about* the status of, and relations among, arguments in object-level AFs (such as *justified*, *rejected*, *defeat*, *preferred*).

We use MAFs for the following reasons: (1) Object-level AFs assume a given fixed preference ordering over arguments. Moreover, evaluating the acceptability semantics for EAFs is more involved than for standard AFs and hence requires the development of new algorithms if they are to be used directly in practical applications. (2) The attacks in an object-level AF are defined in different ways. For example, arguments for conflicting decision options attack each other, but the fact that these decision options are mutually exclusive may not be agreed upon by all stakeholders. In MAFs, the rationales for attacks in object-level AFs are represented as meta-arguments; hence, challenging the rationale for any given attack becomes possible. (3) MAFs provide a general unifying formalism in which one can encode a wide range of object-level AFs and their developments (i.e. not just PAFs and EAFs).

Our work is closely related to other work that uses argumentation to support medical decision-making. Cyras *et al.* [7] aim to build an argumentation-based DSS that uses a Transition-based Medical Recommendation model to represent conflicting clinical guidelines, and they map other data such as medical records and preferences to a formal language; the development of the proposed model is in progress. Glasspool *et al.* [8] propose an argumentation-based planning support system to help patients in exploring different treatment options. The arguments for and against various claims are predefined and they are used to inform patients. ArguEIRA [9] is a clinical DSS aimed at detecting a patient's anomalous reaction to a medication. Some work focus on dialogue-based systems to assist humans in making decisions. Tolchinsky *et al.* [10] propose an approach where physicians argue over the viability of organs for transplantation by using domain-specific argument schemes. Yan *et al.* introduce an argumentation-based DSS for dementia diagnosis where they use possibilistic logic to deal with uncertainty and inconsistency in data [11]. Different from the aforementioned work, we use MAFs to accommodate reasoning about information relevant to decisions—which is typically not part of the domain of the discourse. We formally represent the internal structure of arguments, attacks and explanations that are instantiated by static or dynamic data. As a result, recommendations and explanations can be constructed automatically and interrogated dynamically.

Another line of research focuses on updating the existing AFs as a result of updates to the knowledge base. Some approaches use belief revision techniques (e.g. [12]) while some others study the impact of adding new arguments to existing abstract AFs (e.g. [13]) or to structured argumentation AFs [14]. In our work, we generate a MAF from scratch by considering the current state that depends on dynamic data. Updating existing MAFs is an important future direction to explore for improving the performance in real-world applications.

3 The Consult System

The CONSULT system is an instance of the proposed framework in Figure 1. It is designed to help stroke patients in

TABLE 1
A partial view of the knowledge base that includes information from two guidelines: the hypertension guideline NG136 and NHS website. Arguments are generated according to scheme representations instantiated by static (e.g. guidelines) and dynamic data (e.g. sensor data).

1	**Actions**
2	*# NG136 Hypertension guideline – ccb is an option for people aged over 55, or African/Caribbean of any age Step 1.*
3	h_1: *action*(ccb) :- *patient*(P), *age*(P, A), A>55, not *not_tolerates*(P, ccb).
4	h_2: *action*(ccb) :- *patient*(P), *ethnic_origin*(P, african), not *not_tolerates*(P, ccb).
5	h_3: *action*(ccb) :- *patient*(P), *ethnic_origin*(P, caribbean), not *not_tolerates*(P, ccb).
6	*# thiazide becomes an option when ccb is not tolerated.*
7	h_4: *action*(thiazide) :- *patient*(P), *age*(P, A), A>55, *not_tolerates*(P, ccb).
8	h_5: *action*(thiazide) :- *patient*(P), *ethnic_origin*(P, african), *not_tolerates*(P, ccb).
9	h_6: *action*(thiazide) :- *patient*(P), *ethnic_origin*(P, caribbean), *not_tolerates*(P, ccb).
10	*# NHS Recommendations NSAIDS – ibuprofen and naproxen are two painkillers.*
11	n_1: *action*(ibuprofen) :- *patient*(P), not *not_recommend*(P, ibuprofen).
12	n_2: *action*(naproxen) :- *patient*(P), not *not_recommend*(P, naproxen).
13	**Goal Mapping**
14	*# If the patient suffers from hypertension, the goal is reduce blood pressure (rbp).*
15	g_1: *goal*(rbp) :- *suffers_from*(P, hypertension).
16	*# If the patient suffers from backpain, the goal is reduce pain (rp).*
17	g_2: *goal*(rp) :- *suffers_from*(P, backpain).
18	**Action - Goal Mapping**
19	*# ccb and thiazide are two actions that promote the goal rbp.*
20	f_1: *promotes*(ccb, rbp).
21	f_2: *promotes*(thiazide, rbp).
22	*# ibuprofen and naproxen are two actions that promote the goal rp.*
23	f_3: *promotes*(ibuprofen, rp).
24	f_4: *promotes*(naproxen, rp).
25	**Argument constructed according to schemes**
26	*# If there is an action that promotes a goal, an ASPT argument is constructed.*
27	s_1: a(aspt([*goal*(G), *action*(A), *promotes*(A, G)], *action*(A))) :- *goal*(G), *action*(A), *promotes*(A, G).
28	*# If systolic and diastolic measurements are in a certain range, an amber alert argument is constructed.*
29	s_2: a(amber([*systolic*(P, S),'<150', '>134'], *flag*(amber))) :- *systolic*(P, S), S<150, S>134.
30	s_3: a(amber([*diastolic*(P, D),'<95', '>84'], *flag*(amber))) :- *diastolic*(P, D), D<95, D>84.

self-managing their condition and to adhere to agreed-upon treatment plans, in collaboration with healthcare professionals; hence the framework is specialized to accommodate the needs of this application. A healthcare professional in the project team assessed the correctness of the recommendations as well as the explanations provided to the patients.

3.1 Data Inputs

Wellness sensors provide live data on health parameters and vital signs of a patient. We use sensors to collect data on: heart rate, heart rhythm and blood pressure. A patient's *electronic health record (EHR)* details their medical history. *Clinical guidelines* are official documents published by medical organisations. We use the UK hypertension treatment guideline NG136[2] published by NICE, and follow National Health Service (NHS) recommendations publicly available online (e.g. to recommend a painkiller).

The CONSULT system also considers preferences of stakeholders. For example, a patient can declare her preferences between different painkillers while interacting with the system. Such information is useful in making personalised recommendations.

The way that raw data from wellness sensors is processed in the CONSULT system is determined by clinical guidelines. For example, NG136 suggests that: 1) ambulatory blood pressure monitoring (ABPM) blood pressure measurements should be taken at least twice a day; and 2) the ABPM daytime average over at least 14 measurements should be used when considering raising any alerts. The CONSULT system computes the ABPM daytime averages in line with this clinical information. These values are then passed to the reasoning engine for further processing (lines 29–30, Table 1). Similarly, we use tailored algorithms to map the medical history of the patient and their preferences into the formal language used by the reasoning engine. Note that the formalisation of clinical guidelines is difficult, and it should be done in conjunction with domain experts. Representing the heterogeneous data coming from various information sources in a structured way makes it possible for DSSs to reason about data (while inferring more data) and make recommendations.

3.2 Knowledge Representation

We use first-order logic (FOL) to represent data in terms of facts and rules, stored in a knowledge base (KB). FOL was expressive enough to represent the guideline NG136

[2] https://www.nice.org.uk/guidance/ng136

and NHS recommendations that describe treatments in our specific use-cases.[3]

Table 1 shows a partial view of the KB. This language consists of unary and binary predicates. Variables are denoted by capital letters, predicates are written in *italics*, and constants are in `lowercase fixed-width` font. For example, the unary predicate *action*(`ccb`) states that `ccb` is an action; the binary predicate *age*(`P`, `A`) denotes that the age of the patient `P` is `A`, where `P` and `A` will be instantiated with actual values. Each rule is of the form $head$:- $body$, which means that if the antecedent $body$ holds, then the consequent $head$ holds. In our work, $head$ consists of one predicate whereas $body$ is a conjunction of predicates. Each fact and rule in the KB has a label in the form of x_i.

3.2.1 Specifications

The EHR data provides information such as *ethnic_origin* or *age* of a patient. The hypertension guideline considers such information to recommend treatments. For example, when a person is older than 55 or is of African/Caribbean ethnicity, `ccb` becomes a possible treatment, unless not tolerated. In the example below, a patient and general practitioner (GP) discuss treatment options.

Example 1. Jane is a 60-year-old female stroke survivor, and she suffers from hypertension. Jane's medical history indicates she should not use `ccb`.

The reasoning engine uses Jane's medical record to represent the following facts: *patient*(`jane`), *age*(`jane, 60`), *not_tolerates*(`jane, ccb`) and *suffers_from*(`jane, hypertension`). A treatment will be recommended regarding rules h_1–h_6 (Table 1, lines 2-9). Since `ccb` is not tolerated by Jane, the only available action is `thiazide` as inferred by h_4. This example shows how clinical guidelines, such as NG136, can be represented formally in order to recommend treatments to support patients and healthcare professionals in decision-making. The NHS website is another medical resource that provides advice specific to certain conditions such as symptoms. For example, when a patient suffers from backpain, `ibuprofen` or `naproxen` can be considered unless they are not recommended due to the patient's medical history. We capture this in rules n_1–n_2 (Table 1, lines 10-12).

Our approach applies goal-driven reasoning, treatments are recommended to satisfy goals. For example, if the patient suffers from hypertension, a goal becomes reducing blood pressure (`rbp`); or if the patient has backpain, a goal becomes reducing pain (`rp`) (rules g_1–g_2). The predicate *promotes* connects actions and goals as defined in the guidelines (facts f_1–f_4). For example, the use of `ccb` promotes the goal `rbp`; or the use of `ibuprofen` promotes the goal `rp`.

TABLE 2
A subset of ASP rules to translate object-level AFs to MAFs [16]

r_1: *arg*(*justified*(X))	:- a(X)
r_2: *arg*(*rejected*(X))	:- a(X)
r_3: *arg*(*defeat*(X, Y))	:- r(a(X), a(Y))
r_4: *att*(*defeat*(X, Y), *justified*(Y))	:- *arg*(*defeat*(X, Y)), *arg*(*justified*(Y))
r_5: *att*(*rejected*(X), *defeat*(X, Y))	:- *arg*(*defeat*(X, Y)), *arg*(*rejected*(X))
r_6: *att*(*justified*(X), *rejected*(X))	:- *arg*(*justified*(X)), *arg*(*rejected*(X))
r_7: *att*(*prefer*(X, Y), *prefer*(Y, X))	:- *arg*(*prefer*(X, Y)), *arg*(*prefer*(Y, X))
r_8: *arg*(*prefer*(X, Y))	:- p(a(X), a(Y))
r_9: *att*(*prefer*(X, Y), *defeat*(Y, X))	:- *arg*(*prefer*(X, Y)), *arg*(*defeat*(Y, X))

3.2.2 Schemes

Argument, attack and explanation schemes are templates representing common patterns of reasoning and relate a set of premises to a conclusion, all of which are sentences that can be represented in FOL. Each sentence includes variables, instantiated by a KB. These domain-specific schemes are kept in the *Schemes* repository. We represent them as rules while retaining the internal structure of the constructed argument. Formally, each scheme rule consists of a *rule body*, a conjunction of predicates (premises of the scheme), and a *rule head*, of the form a($sname$(P,c)); where 'a' stands for an argument, $sname$ is the scheme name, P is the set of predicates used in the rule body and c is the conclusion of the scheme.

Table 1 shows three scheme rules: s_1, s_2 and s_3. Argument Scheme for a Proposed Treatment (ASPT) [17] is an argument scheme to construct an argument in support of an action promoting a specific goal; s_1 represents ASPT formally. In Example 1, applying the rules g_1, f_2 and s_1 will result in one argument A_1: a(aspt([*goal*(`rbp`), *action*(`thiazide`), *promotes*(`thiazide, rbp`)], *action*(`thiazide`))).

In the following example, ASPT is instantiated to recommend a painkiller this time.

Example 2. Jane is experiencing backpain. She needs a painkiller. Jane uses the CONSULT system to get some recommendations.

As Jane interacts with the CONSULT system, the reasoning engine adds a new fact to its KB: *suffers_from*(`jane, backpain`). According to g_2, a new goal of reducing pain (`rp`) will be inferred. `ibuprofen` and `naproxen` are two possible treatments to offer (n_1–n_2). Together with the existing facts f_3 and f_4, s_1 will instantiate two ASPT arguments: A_2: a(aspt([*goal*(`rp`), *action*(`ibuprofen`), *promotes*(`ibuprofen, rp`)], *action*(`ibuprofen`))) and A_3: a(aspt([*goal*(`rp`), *action*(`naproxen`), *promotes*(`naproxen, rp`)], *action*(`naproxen`))).

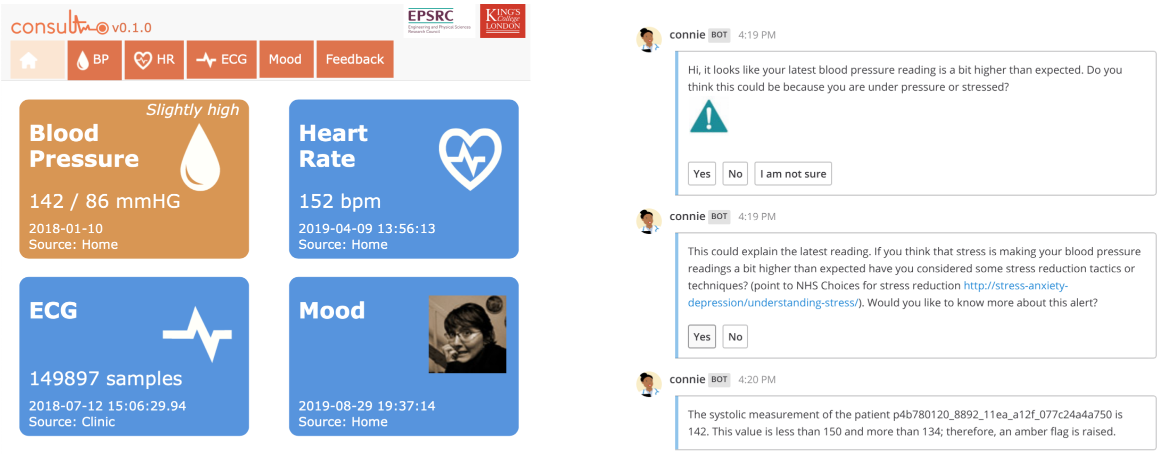
Another scheme, AMBER, constructs arguments to represent alerts if blood pressure readings are in the range specified in the hypertension guideline NG136. In Table 1, s_2 and s_3 represent AMBER arguments formally.

Example 3. Jane uses her blood pressure monitor to check her blood pressure. The CONSULT system displays her readings on a dashboard (Figure 2a). The weekly average value is 142/86 mmHg, which is somewhat high.

The reasoning engine represents this information with the following facts: *systolic*(`jane, 142`) and *diastolic*(`jane, 86`). According to s_2, Jane's average systolic reading triggers an AMBER argument that is instantiated as: A_4: a(amber([*systolic*(`jane, 142`), '<150', '>134'],

[3] Other applications may require languages that can represent temporal information and uncertainty. These would need to be able to build arguments using these more complex languages (e.g. [11], [15]).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/MIS.2021.3051420



Slightly high
Blood Pressure
142 / 86 mmHG
2018-01-10
Source: Home

Heart Rate
152 bpm
2019-04-09 13:56:13
Source: Home

ECG
149897 samples
2018-07-12 15:06:29.94
Source: Clinic

Mood
2019-08-29 19:37:14
Source: Home

(a)

connie BOT 4:19 PM

Hi, it looks like your latest blood pressure reading is a bit higher than expected. Do you think this could be because you are under pressure or stressed?

Yes | No | I am not sure

connie BOT 4:19 PM

This could explain the latest reading. If you think that stress is making your blood pressure readings a bit higher than expected have you considered some stress reduction tactics or techniques? (point to NHS Choices for stress reduction http://stress-anxiety-depression/understanding-stress/). Would you like to know more about this alert?

Yes | No

connie BOT 4:20 PM

The systolic measurement of the patient p4b780120_8892_11ea_a12f_077c24a4a750 is 142. This value is less than 150 and more than 134; therefore, an amber flag is raised.

(b)

Fig. 2. Two views of CONSULT system; (a) The dashboard showing a blood pressure amber alert; and (b) An alert dialogue in the chatbot.

flag(amber))). s_3 will lead to the construction of an AMBER argument for Jane's average diastolic reading similarly.

3.3 Meta Engine and Solver

Scheme rules provide a way to construct object-level arguments automatically. *Meta Engine* translates object-level arguments and attacks into a MAF by using our developed encoding [16]. This encoding can be used in conjunction with an answer set programming (ASP) approach to evaluate the justified arguments in a MAF. The code for our implementation, and examples including the three in this paper, are available online[4].

In Table 2, we provide a subset of ASP rules that translate object-level AFs into MAFs. At the object-level, a(X) and r(a(X), a(Y)) represent the argument X and the attack relation between the arguments X and Y, respectively. At the metalevel, the meta-argument A and the meta-attack between the meta-arguments A and B are described by *arg*(A) and *att*(A, B), respectively. Rules r_1–r_2 assign a *justified* or *rejected* status to an object-level argument at the metalevel; whereas r_3 translates an object-level attack to a *defeat* meta-argument. Rules r_4–r_6 generate the meta-attacks between meta-arguments. MAFs can represent preferences or attacks on attacks. For example, r_7–r_9 define the semantics of including preferences in a MAF. In r_8, p(a(X), a(Y)) represents a preference stating that X is being preferred over Y, which is explicitly represented as a meta-argument. In Example 2, if Jane has a preference for ibuprofen over naproxen, such a preference could be represented as p(A_2,A_3). Rule r_7 ensures that a meta-attack exists between two conflicting preference meta-arguments. When there is a preference meta-argument that supports X, any attacks on X will be attacked as stated in r_9.

In order to demonstrate our approach at the metalevel, we focus on Example 3. Our encoding will translate A_4 into two meta-arguments: M_1: *arg*(*justified*(amber([*systolic*(jane, 142), '<150', '>134'], *flag*(amber)))) and M_2: *arg*(*rejected*(amber([*systolic*(jane, 142), '<150', '>134'], *flag*(amber)))). r_6 will generate a meta-attack from M_1 to M_2, D_1: *att*(*justified*(amber([*systolic*(jane, 142), '<150', '>134'], *flag*(amber))), *rejected*(amber([*systolic*(jane, 142), '<150', '>134'], *flag*(amber)))). The resulting MAF can then be represented as: $\langle\{M_1, M_2\}, \{D_1\}\rangle$.

We use DLV[5] as the ASP-solver to compute extensions under the grounded semantics (the acceptability semantics that yields a single set of justified arguments). The reasoning engine evaluates the constructed MAF and returns the set of justified meta-arguments. In Example 3, there is one justified meta-argument, M_1, which is not attacked by any other meta-argument.

3.4 Explanation Generator

The results are then processed by the *Explanation Generator* component, which has access to a set of explanation schemes, to generate semi-structured explanations [18]. If an explanation scheme exists for an argument, a textual explanation is also generated. For example, an explanation for an AMBER argument is constructed according to the explanation scheme $\langle AMBER$, *"The systolic measurement of the patient* $\{P\}$ *is* $\{S\}$*, this value is less than 150 and more than 134 and therefore an Amber flag is raised."*$\rangle$. The variables in the textual explanation (P and S) are replaced by actual values as they appear in the *justified* AMBER argument. In Example 3, the explanation "The systolic measurement of the patient jane is 142, this value is less than 150 and more than 134 and therefore an Amber flag is raised." will be constructed by the *Explanation Generator*.

3.5 Recommendations and Explanations

The CONSULT system was implemented as a mobile application running on an Android tablet device, Figure 2 depicts two views of the system. Recommendations and explanations are provided to the patients through a simple chatbot (Figure 2b), working in conjunction with the dashboard (Figure 2a). The dashboard flags issues that the patient may wish to attend to; for example, Figure 2a shows the amber blood pressure alert discussed in Example 3. In addition, the dashboard enables users to visualise their data

[4] https://git.ecdf.ed.ac.uk/nkokciya/metalevel-aspartix/

[5] http://www.dlvsystem.com/dlv/

in context, access advice and input information (e.g. mood tracking). The chatbot enables users to interactively access more targeted support as discussed in Example 2.

3.6 Pilot study

We conducted a formative pilot study to assess the usability and acceptability of the CONSULT system with two different versions: one version had the dashboard only, the other also had the chatbot. Six healthy volunteers were recruited for a seven day within-subjects mixed-methods study run in-the-wild. Participants started with either a dashboard-only version or one that had both dashboard and chatbot, and then swapped halfway through.

As part of the pilot study, participants were asked to collect measurements (blood pressure, heart rate and ECG) from wellness sensors and to input data (such as mood) on a regular basis. They were encouraged to interact with the interfaces (dashboard and chatbot). The system would display alerts when their blood pressure was raised and initiate a chatbot dialogue (during the days when they had access to the chatbot). The results of the pilot study show that participants were able to use the CONSULT system for a week, and to interact with the chatbot when a dialogue was initiated automatically.

4 Conclusion

In this paper, we propose an argumentation-based DSS that: (i) can reason with heterogeneous data such as static data coming from clinical guidelines or dynamic data provided by sensors; (ii) considers user preferences as part of the reasoning process; and (iii) provides textual explanations to explain automated decisions. The core reasoning is based on MAFs that offer a uniform encoding of object-level AFs and relevant meta information. We implemented an instance of such a DSS, the CONSULT system, which was successfully deployed. The pilot study demonstrated that real users could use the system over an extended period. We have received ethics approval for two further user studies, which will assess the effectiveness of the system in clinical terms. In the first study, stroke patients will use the CONSULT system to self-manage their health conditions, as in our pilot study. The second study will focus on the reasoning aspects of the system when using knowledge from clinical guidelines (e.g. NG136); recommendations generated for sample cases will be evaluated by a panel of general-practitioner experts.

Acknowledgments

This work was supported by the UK Engineering & Physical Sciences Research Council (EPSRC), grant #EP/P010105/1.

References

[1] T. Porat, N. Kökciyan, I. Sassoon, A. P. Young, M. Chapman, M. Ashworth, S. Modgil, S. Parsons, E. Sklar, and V. Curcin, "Stakeholders' views on a collaborative decision support system to promote multimorbidity self-management: barriers, facilitators and design implications," in *American Medical Informatics Association Annual Symposium*, 2018.

[2] I. Rahwan and G. R. Simari, *Argumentation in Artificial Intelligence*. Springer, 2009.

[3] S. Modgil and T. Bench-Capon, "Metalevel Argumentation," *Journal of Logic and Computation*, vol. 21, no. 6, pp. 959–1003, 2011.

[4] P. M. Dung, "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–358, 1995.

[5] L. Amgoud and C. Cayrol, "A reasoning model based on the production of acceptable arguments," *Annals of Mathematics and Artificial Intelligence*, vol. 34, no. 1-3, pp. 197–215, 2002.

[6] S. Modgil, "Reasoning about preferences in argumentation frameworks," *Artificial Intelligence*, vol. 173, no. 9, pp. 901–934, 2009.

[7] K. Cyras, B. Delaney, D. Prociuk, F. Toni, M. Chapman, J. Dominguez, and V. Curcin, "Argumentation for explainable reasoning with conflicting medical recommendations," in *Reasoning with Ambiguous and Conflicting Evidence and Recommendations in Medicine Workshop*, 2018, pp. 14–22.

[8] D. Glasspool, J. Fox, A. Oettinger, and J. Smith-Spark, "Argumentation in decision support for medical care planning for patients and clinicians." in *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*, 2006, pp. 58–63.

[9] M. A. Grando, L. Moss, D. Sleeman, and J. Kinsella, "Argumentation-logic for creating and explaining medical hypotheses," *Artificial Intelligence in Medicine*, vol. 58, no. 1, 2013.

[10] P. Tolchinsky, U. Cortes, S. Modgil, F. Caballero, and A. Lopez-Navidad, "Increasing human-organ transplant availability: Argumentation-based agent deliberation," *IEEE Intelligent Systems*, vol. 21, no. 6, 2006.

[11] C. Yan, H. Lindgren, and J. C. Nieves, "A dialogue-based approach for dealing with uncertain and conflicting information in medical diagnosis," *Autonomous Agents and Multi-Agent Systems*, vol. 32, no. 6, pp. 861–885, 2018.

[12] M. A. Falappa, G. Kern-Isberner, and G. R. Simari, "Explanations, belief revision and defeasible reasoning," *Artificial Intelligence*, vol. 141, no. 1-2, pp. 1–28, 2002.

[13] C. Cayrol, F. D. de Saint-Cyr, and M. Lagasquie-Schiex, "Change in abstract argumentation frameworks: Adding an argument," *Journal of Artificial Intelligence Research*, vol. 38, pp. 49–84, 2010.

[14] G. Alfano, S. Greco, F. Parisi, G. Simari, and G. Simari, "An incremental approach to structured argumentation over dynamic knowledge bases," in *Conference on Principles of Knowledge Representation and Reasoning*, 2018, pp. 78–87.

[15] J. Fox, N. Johns, and A. Rahmanzadeh, "Disseminating medical knowledge: the proforma approach," *Artificial Intelligence in Medicine*, vol. 14, no. 1-2, pp. 157–182, 1998.

[16] N. Kökciyan, I. Sassoon, A. Young, S. Modgil, and S. Parsons, "Reasoning with metalevel argumentation frameworks in aspartix," in *Conference on Computational Models of Argument*, 2018, pp. 463–464.

[17] N. Kökciyan, I. Sassoon, A. P. Young, M. Chapman, T. Porat, M. Ashworth, V. Curcin, S. Modgil, S. Parsons, and E. Sklar, "Towards an Argumentation System for Supporting Patients in Self-Managing their Chronic Conditions," in *Proceedings of the AAAI Joint Workshop on Health Intelligence*, 2018.

[18] N. Kökciyan, S. Parsons, I. Sassoon, E. Sklar, and S. Modgil, "An argumentation-based approach to generate domain-specific explanations," in *Multi-Agent Systems and Agreement Technologies*, 2020, pp. 319–337.

Nadin Kökciyan is a Lecturer in Artificial Intelligence in the School of Informatics at University of Edinburgh; and a Visiting Research Fellow in the Department of Informatics at King's College London. Her primary focus is developing AI techniques to support decision-making in multi-agent systems. Nadin received her PhD degree in Computer Engineering from Bogazici University. Contact her at nadin.kokciyan@ed.ac.uk.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/MIS.2021.3051420

Isabel Sassoon is a Lecturer in Computer Science (Data Science) in the Department of Computer Science at Brunel University London; and a Visiting Research Fellow in the Department of Informatics at King's College London. Her research interests are in data-driven automated reasoning and modelling, and its transparency and explainability. Isabel received her PhD degree in Informatics from King's College London, UK. Contact her at `isabel.sassoon@brunel.ac.uk`.

Elizabeth Sklar is Professor in Agri-Robotics at the University of Lincoln. She worked as a software engineer in industry research labs for 10 years before shifting to academia in the late 1990's. Her main focus is on interaction in multi-robot and human-robot systems, including the use of argumentation-based dialogue for reaching decisions and structuring evidence to explain those decisions. She received her PhD from Brandeis University. Her email address is `esklar@lincoln.ac.uk`.

Sanjay Modgil is a Reader in Artificial Intelligence and Head of the Reasoning and Planning Group at King's College London. His research interests are in development of logic-based models of argumentation and dialogue, rationality under resource bounds, and the ethics of AI. He received his PhD in logic at Imperial College London. Contact him at `sanjay.modgil@lkcl.ac.uk`.

Simon Parsons is a Professor in AI and Machine Learning at the University of Lincoln. He obtained his PhD from Queen Mary, University of London, and has been a researcher in AI for more than 30 years. His research is primarily concerned with decision making in complex environments, particularly focusing on data-driven methods that use computational argumentation. His email address is `sparsons@lincoln.ac.uk`.