Argumentation-Based Health Information Systems: A Design Methodology

Helena Lindgren, Timotheus Kampik, Esteban Guerrero Rosero, Madeleine Blusi, and Juan Carlos Nieves, *Umeå University, 90187 Umeå, Sweden*

In this article, we present a design methodology for argumentation-based health information systems. With a focus on the application of formal argumentation, the methodology aims at eliciting requirements in regard to argumentation reasoning behavior, knowledge and user models, and business logic on levels below and above the argumentation layer. We highlight specific considerations that need to be made dependent on the system type, i.e., for clinical decision-support systems, patient-facing systems, and administration systems. In addition, we outline challenges in regard to the design of argumentation-based intelligent systems for healthcare, considering the state of the art of argumentation research, health information systems, and software design methods. For each challenge, we outline a mitigation strategy.

Formal argumentation has emerged as a promising method for automated reasoning. While a large body of works exists on theoretical aspects of formal argumentation,[1] the application of the method to real-world use cases is—despite some success stories—still at an early stage. Hence, it is important to advance research that closes the gap between the theoretical knowledge the community is accumulating and real-world applications.

Formal argumentation approaches are frequently proposed in the context of health information systems[2] to derive conclusions from conflicting, inconsistent, or uncertain information. The following (simplified) example highlights the usefulness of argumentation in healthcare (using the *abstract argumentation* approach).[3] A patient shows symptoms that could either indicate attention-deficit/hyperactivity disorder (ADHD) (the treatment of which we denote by argument *c*) or depression (whose treatment we denote by *d*). ADHD can be treated with stimulant medication (argument *b*), depression with antidepressants (argument *a*). A decision support system based on standardized clinical pathways recommends the intake of *a*, based on treatment plan *d*. In contrast, a medical specialist recommends the intake *b*, based on treatment plan *c*. Because stimulate medication and antidepressants counteract each other, only one of the treatment options can be chosen. The practitioner who is responsible for treating the patient needs to decide which advise to follow. Figure 1 shows the argumentation graph of the example.

In formal argumentation, problems of this type can be expressed in a mathematical model, which can be solved using a formal method, e.g., a so-called *argumentation semantics*. Note that in the context of formal argumentation, *arguments* can model any type of knowledge, and are not necessarily based on natural language. To solve the framework in Figure 1, we first need to answer some (potentially use case-specific) questions, for instance:

1) Are some arguments stronger than others, for example because they are more likely to be true or come from a more authoritative source?
2) What medical knowledge should inform the internal structure of an argument, and how should its relation to other arguments be generated?
3) How should cycles of arguments that can arise from dependencies between different sources of information (e.g., medical guidelines from different organizations, diverging opinions of medical practitioners) be resolved?

Digital Object Identifier 10.1109/MIS.2020.3044944
Date of publication 15 December 2020; date of current version 14 May 2021.

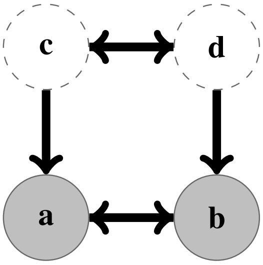

FIGURE 1. Example of an argument graph. Treatment *c* implies abstaining from the intake of medicine *a*, whereas treatment *d* implies not taking *b*. Hence, only one of the treatment options can be chosen.

In the past, formal argumentation has been highlighted as a potential solution to a set of common challenges that arise when designing clinical decision support systems.[4] Still, and although the formal argumentation community is thriving, no success stories that report on the wide-spread adoption or large-scale clinical trial of argumentation-based systems in healthcare exist. Indeed, much of the research on formal argumentation and healthcare is limited to the definition and (sometimes) the implementation of running examples, without the involvement of domain experts, the creation of open-sourced software artifacts, and the empirical evaluation of the developed prototype applications.

This article presents a design methodology for argumentation-based health information systems that can facilitate stronger applied research in this domain in the future. The design methodology is informed by our own research experiences, and in particular, the lessons we have learned during the past 15 years of research at the intersection of formal argumentation and artificial intelligence for healthcare. Table 1 provides an overview of some application scenarios of argumentation to the healthcare domain, during which the methodology was defined and refined, and highlights some design aspects (which are explained in this article) for each use case. The methodology can be considered a complementary, argumentation-centered perspective on a software development process that assumes a somewhat *agile*, iterative approach to software development,[5] which we consider a reasonable assumption in a research-intense development context; however, it can be adjusted to better integrate with other, noniterative software development approaches.

DESIGN METHODOLOGY

The design methodology can be divided into following three phases.

1) Use case and architecture identification.
2) Iterative system design and implementation.
3) Empirical evaluation.

The end of each phase (and end of the Phase 2 iteration cycle) represents an inflection point, at which a preliminary evaluation of the system design is conducted that informs the decision on how to proceed further. Also, each phase results in the creation of distinct artifacts, which can, for example, be presented in a dissemination or handed over to third parties. Before the first phase, focus groups of stakeholders (potential users, domain experts, etc.) should be set up that accompany the design process.

Use Case and Architecture Identification

The first phase is concerned with use case identification and high-level application architecture definition.

Identify use case

Right from the start, the system's use case should be defined in close collaboration with relevant medical experts. Adopting an activity-centric perspective is important to specify the type of support the system should provide in the application scenario, as well as the way the system and its users are supposed to

TABLE 1. Medical use case examples, their use case types, knowledge bases, conflict types, and argumentation approaches.

Medical Use Case	Use Case Type	Knowledge Base	Conflict Types	Argumentation Approach	Reference
Decision-support for dementia diagnosis and treatment	Clinical decision-support system	Possibilistic answer-set programs	Guideline inconsistencies	Inquiry dialogues	Yan *et al.*[6]
Self-management application for mental/social well-being	Patient-facing system	Extended logic programs	Human goal conflicts	Abstract argumentation	Guerrero *et al.*[7]
Fall prevention application for the elderly	Patient-facing system	Extended logic programs	Inconsistencies in human activities	Abstract argumentation	Guerrero *et al.*[8]

collaborate to work toward the users' and the healthcare providing organization's goals.

Define high-level architecture

After the use case has been defined, high-level requirements should be specified, and the preliminary architecture should be designed. It is important that the architecture primarily serves the use case; the alignment with a basic research purpose should merely be a desirable side effect. At this point, it is typically sufficient to model the architecture using general graphical diagramming tools; more detailed specifications in standardized modeling notations such as the Unified Modeling Language or ArchiMate[9] can follow later.

Inflection point. After this phase, it should be clear whether an argumentation-based system can, indeed, serve the use case at hand. If this is not the case, a different type of system (for example: a simple rule-based system or a machine learning classifier) can be implemented, or—if this is not feasible—the project can be abandoned. In particular, the application of formal argumentation as an *agreement technology* implies that the use case requires the management of potentially inconsistent or uncertain information from multiple sources in decision processes. *Artifacts.* Results of this phase are a preliminary feasibility analysis, a high-level architecture and requirements specification, and an activity analysis specifying the work and decision-making processes that are to be supported.

Iterative System Design

The second phase implements a system prototype that is based on the artifacts that result from the previous phase. The implementation is conducted iteratively in collaboration with relevant stakeholders.

Design knowledge model

A domain-specific knowledge model should be devised, again in collaboration with domain experts. The knowledge model should be based on existing models of the corresponding domain, for example, on standardized data models like Health Level Seven (HL7),[10] clinical paths that have been specified by the relevant authorities, or international disease classification standards. However, it is important to consider that local realities might diverge from the standardized specifications. For example, the information scheme that a specific electronic health record system uses might not be standard-compliant, and even if it is, information-completeness is practically not always given. Indeed, in this fact lies strength of the argumentation-based approach: Inconsistencies between ideal standards and local realities can be explicitly modeled and resolved at run-time. Another important aspect when designing the knowledge model is the knowledge modeling language. Because medical professionals are not necessarily well-versed in knowledge modeling languages like the Web Ontology Language, it is important to use a high-level, potentially informal language to specify the rough model, and, when agreement on the most important aspects is reached, iteratively refine details.

Cocreate interactive prototypes

In parallel to building the knowledge model, interactive prototypes are created; again, domain experts should be involved, and in addition, potential nonexpert stakeholders like patients. Knowledge model and user interface depend on each other. On the one hand, the user interface provides abstractions of the knowledge model that compromises between accuracy and conciseness. On the other hand, the knowledge model needs to consider user interaction needs. During the initial prototype design phase, knowledge model and interactive prototype should be only loosely coupled, to ensure that user interaction needs are treated as first-class citizens. Tools that allow for the rapid creation of prototypes can already be used in this phase. However, the employment of such tools at a too early stage can hamper creativity, as a rather strict frame for the system's user interface (UI) design is dictated by these platforms.

Elicit arguments

After the general knowledge model and UI have been designed, argument elicitation can begin. As the first step in this activity, one needs to distinguish between elicitation at design-time and elicitation at run-time.

1) At design-time, arguments are manually curated, mined from an unstructured dataset, or automatically transferred from another already well-structured knowledge base. Arguments and argumentation frameworks can be refined and sanity-checked before deployment, which places less strict requirements on the algorithms for argument generation.
2) At run-time, arguments can be derived directly from user-interactions, or from additional data that is uploaded to the system; i.e., in this case, algorithms for the autogeneration of arguments need to be defined and properly tested to ensure they perform as intended when the system is deployed.

To construct arguments and detect conflicts, the formal logic that formalizes the semantics of the knowledge specification language should provide polynomial-time inference operators. This means it is feasible to define efficient algorithms for constructing arguments and detecting attacks between arguments in a medical knowledge base. To *mine* arguments from unstructured or poorly structured data, machine learning techniques can be applied.[11] During elicitation, it is recommendable to assign each argument to one or several groups; for example, one argument can be assigned to the group *end-user preference*, whereas another one is assigned to the groups *expert diagnosis* and International Classification of Diseases. No matter whether an argument is elicited at run-time or design-time, the argument's *strength* needs to be considered. The strength can be derived from the groups the argument is in, or be inferred from data. For instance, in case a diagnosis-support system is implemented, expert opinions might be considered stronger than end-user self-assessments, because the latter can be considered as less reliable.[6] However, when designing a self-management mobile application, end-user preferences can be considered more important than expert opinions in some contexts, for example with regards to the configuration of daily schedules and motivational recommendations.[7] In formal argumentation, strength can be modeled qualitatively, for example by constructing preference orders over arguments,[12] or quantitatively, for example, using probabilistic approaches.[13]

Also, it is crucial to consider the *type(s) of argumentation dialogues* the system should support; following are the examples.[14]

1) Inquiry dialogues: The system uses (multiagent) argumentation for knowledge-seeking purposes, for example, to derive new conclusions by eliciting arguments from distributed and potentially inconsistent knowledge bases.
2) Deliberation dialogues: The system facilitates tradeoffs between the positions of different parties, for example, by consolidating conflicting opinions of experts or conflicts between medical guidelines on regional and national level.
3) Persuasion dialogues: The system uses formal argumentation to persuade a user, for example, by exchanging arguments with a patient to motivate them to work toward a specific goal.

The selection of dialogue type is done based on the activity analysis that specifies phases and decision points in the clinical pathway and decision processes. Alongside with making design decisions with regards to argument strength and dialogue type, the *structure* of the arguments should be defined. The structure is dependent on the knowledge model, i.e., on the nature and structure of the knowledge base(s) from which conclusions are to be derived. For this, one can rely—to some extent—on the argument interchange format,[15] which is an early stage effort to provide guidelines and best practices for the elicitation of arguments from knowledge bases, as well as for argument exchange. In conjunction with the design of argument structure and argument strength, the argumentation *semantics* that determines how an argumentation graph is resolved needs to be defined; in this context, the argumentation principles[16] that a specific semantics fulfills should be considered. In particular, principles that are aligned with the requirements of the application scenario should be identified.

An important concern in argumentation semantics is their *computational complexity*. The decision problems of the well-accepted argumentation semantics range from NP-complete to $\Pi_2^{(p)}$-complete.[17] In this context, tradeoffs may need to be made; for instance, argumentation semantics like the grounded semantics that are of comparably low computational complexity do not allow for particularly nuanced conflict resolution (colloquially speaking). To enhance computational performance, some applications of formal argumentation place conditions on the structure or size of the argumentation graph and, for example, only construct noncyclic argumentation graphs.[18] Hence, use case-specific running examples should be constructed to evaluate if a semantics' output is reasonable from a computational complexity and a subject-matter expert perspective.

Design knowledge model UI

In parallel to the argument elicitation step, the abstraction the user interface provides on system data and processes, i.e., on argumentation frameworks, knowledge bases, and reasoning methods, should be designed. In particular, the following questions should be answered.

1) What abstractions should the user interface provide on the knowledge base and how detailed should these abstractions be?
2) Are there data that should under no circumstances be exposed to a user, for example, for data-privacy reasons?
3) When should the user be able to add additional knowledge as a means of providing feedback, how should the knowledge be integrated into

the existing knowledge base, and what actions should be triggered by such user feedback?

As a rule of thumb, the level of detail the user interface provides in terms of both data view abstractions and data input and feedback opportunities should be more fine-grained for medical expert users than for patients.

Implement system prototype

In parallel to the previous four design steps, the system prototype should be iteratively implemented. As mentioned earlier, rapid prototyping tools can facilitate the implementation. In particular, data schemes that underlie the system should ideally be (auto)generated directly from the models that have been defined in collaboration with the focus group(s). Given the safety-critical nature of medical information systems, even for the prototype implementation, quality assurance best practices like test-driven development and continuous integration should be followed. Expert-validated running examples can serve as test cases.

Evaluate qualitatively

To assess whether the system is ready to be deployed for a long-running empirical study, a preliminary, qualitative evaluation should be conducted. In contrast to the previous stages of the iterative design process, this evaluation should be conducted using a deployed, running, and stable system instance. Ideally, a new set of experts and end-users is involved in the evaluation. This avoids biased feedback from persons who are already invested from a design perspective. As a first evaluation step, the initial knowledge base should be validated. For example, the system's outputs can be compared to clinical guidelines and protocols, and be evaluated by medical experts and patients. Additional feedback can be solicited on the system itself and the conclusions derived from it in example scenarios, and by conducting trial runs of the system in a carefully controlled real-life environment. Also, observations should be made on how the use of the system affects decision-making and human behavior. For example, given a decision support system, it should be documented to what extent users follow the system's recommendations.

Inflection point. When an implementation and design cycle has been absolved, the system designers should decide whether 1) further iterations are necessary, 2) the system is sufficiently mature for an empirical study, or 3) the system prototype should be discontinued without an empirical evaluation being feasible. *Artifacts.* The resulting artifacts of this phase are a detailed system specification, a system prototype, and a preliminary qualitative evaluation of the system.

Empirical Evaluation

In the final phase, the system is empirically evaluated with the objective to provide a strong assessment of the system's usefulness in real-world medical application scenarios.

Evaluate empirically

To assess the medical efficacy of the developed system, an empirical evaluation in practice is necessary, if possible in a randomized-controlled trial setting. Ideally, the medical practitioners who participate in the empirical evaluation of the system are not (at least not exclusively) the ones who have helped design it. Otherwise, there is a risk that the participating medical experts are 1) biased toward the efficacy of the system and 2) have a level of expertise in working with the system that is hard to obtain by third parties.

To enable a real-world study, the system needs to be integrated into the corresponding healthcare process or *clinical pathway*. If the study is conducted across regions with different local routines, the study should ideally be detached from these local routine variants to ensure comparability across regions. In the empirical evaluation, a mixed-method approach is recommended. Generally, the system's recommendations or decisions can be quantitatively evaluated. Unexpected system behavior can be analyzed qualitatively to find the reason for the deviation and to determine whether the unexpected behavior is indeed *undesirable*.

Inflection point. When the empirical evaluation has been concluded, further steps can be planned based on the implications of the results. In particular, follow-up studies can be conducted, for which the system might need to be customized to fit a new context (for example, to the needs of medical practitioners in a different country). In successful cases, the hand-over of the system to parties that can ensure long-term operations can be initiated; for instance, local health authorities or other healthcare-providing organizations may have an interest in taking over the maintenance and operations of a system that is evidently useful in a particular medical context. *Artifacts.* The resulting artifact of this phase is an analysis document that contains an empirical evaluation of the system.

Figure 2 shows the process diagram of the design methodology.

APPLICATION SUBDOMAINS

The context in which a health information system is applied should self-evidently inform its design. A particularly important distinction is whether the system is primarily used by medical experts, by patients, or by administration staff.

Clinical Decision-Support Systems

In recent years, it has been acknowledged that the handling of inconsistent knowledge, for example, of diverging expert opinions, is crucial in many medical decision-support scenarios. However, existing, industry-scale products do not provide first-class abstractions for managing inconsistent or conflicting knowledge. Argumentation-based clinical decision-support systems can help facilitate decision-making by enabling the management of these inconsistencies that are common in medical decision-making, and indeed often part of the medical process by design, e.g., when the opinions of several medical experts are solicited with regards to a specific case. An instance of an argumentation-based decision support system is a dementia diagnosis and management support application as presented by Yan *et al.*[6] In a decision-support context, it is important that arguments from different types of sources are marked according to the corresponding source type's *strength*. For instance, the argument strength assignment can reflect that the opinion of a single practitioner cannot invalidate well-established domain knowledge as specified in national or international guidelines. A potential hierarchy of arguments could be as follows (assuming a total ordering), but is most certainly scenario-dependent.

1) Arguments derived from global guidelines and standards take precedence over all other arguments.
2) Local guidelines define aspects that have not been specified in sufficient detail by global guidelines and standards, but arguments derived from local guidelines are weaker than arguments derived from global guidelines.
3) Arguments derived from practitioner opinions can inform decision-making on individual cases but are weaker than local and global guidelines.

From a formal perspective, a plethora of potential *argumentation strength* design mechanism exist[19]; so far, there are no best practices as to when to choose which mechanism.

Patient-Facing Systems

When implementing patient-facing health information systems, it is crucial that tradeoffs are made between the advice and recommendations that the system provides based on existing medical knowledge and patient data, and the patient's self-assessment and personal preferences. To facilitate these compromises, formal argumentation methods can be considered a natural fit. A particularity of patient-facing systems is that the recommendations provided and the actions executed by these systems must not only fulfill the quality standards of the healthcare domain, but also allow for informed dissent if the assumptions

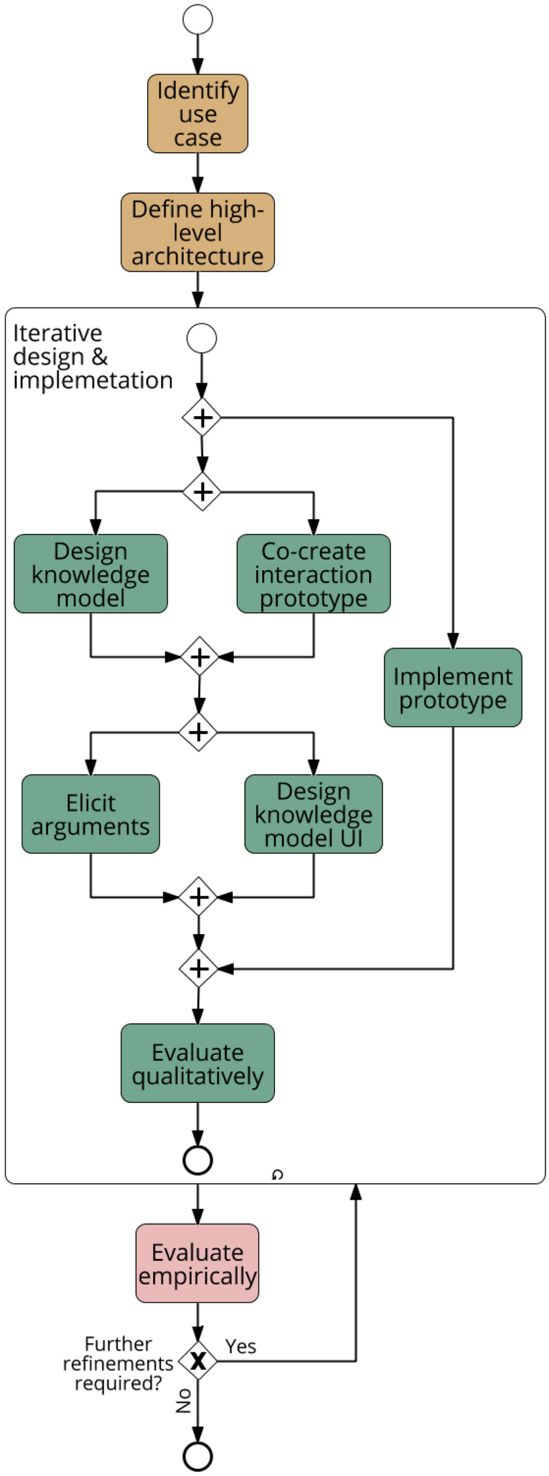
Identify use case
Define high-level architecture
Iterative design & implemetation
Design knowledge model
Co-create interaction prototype
Implement prototype
Elicit arguments
Design knowledge model UI
Evaluate qualitatively
Evaluate empirically
Further refinements required?
Yes
No

FIGURE 2. Design methodology process diagram.

the system makes are not aligned with a patient's personal preferences or life circumstances. An instance of this system type is an argumentation-enabled behavior change management personal assistant.[7] If the assistant recommends a user to do more physical exercise, the user should have the possibility to provide the feedback to the system that in the current situation, family responsibilities do not allow to act according to the recommendation. The possibility for dissent is not only important from the perspective of medical efficacy, but also for the sake of personal autonomy: A user should be empowered to make tradeoffs between an optimally healthy lifestyle and other aspects of life, and not have a system dictate all details. From the perspective of argumentation, this means that users should be able to add new arguments to the system that *defeat* the *conclusions* the system provides; i.e., the system should be able to provide alternative conclusions based on the new arguments while maintaining the evidence-based knowledge foundation. This behavior allows for verifiable compliance with guidelines and regulations, which is, for example, a requirement for regulatory approvals like CE-marking. In this regard, the testability and verifiability of argumentation-based systems is a key advantage, for example, in comparison to machine learning-based systems.

Administrative Health Information Systems

A third type of healthcare information system in which formal argumentation can be applied are administrative systems, for example, enterprise resource planning systems that are used in the healthcare domain. To our knowledge, little research exists on the application of argumentation-based methods for this healthcare information system type; one exception is an argumentation-based scheduling tool for nurses.[20] A possible reason for the sparsity of related research is that administrative systems are tightly integrated into the information system landscape of an organization; this increases the engineering effort and organizational overhead of implementing prototypes that are sufficiently mature for pilot studies. However, from a use-case perspective, the management of conflicting information may be considered useful from an administrative planning perspective; for example, in the wake of the COVID-19 crisis, decisions on how to distribute medical equipment to different administrative regions could be informed by infection spread prognoses that are provided by disagreeing epidemiology experts or based on conflicting base scenarios. Besides the aforementioned peculiarities, the implementation of argumentation-based healthcare administration systems can have similar implications as the implementation of systems for medical experts in that conflicts between *global* (e.g., governmental) guidelines, local implementation details, and diverging expert assessments need to be made.

CHALLENGES AND RESEARCH STRATEGIES

The design methodology may be considered an idealized approach that cannot always be complied with in reality, for example, because of constraints in time, budget, or expertise. Indeed, when examining existing literature on the topic, we observe a set of shortcomings, from which we derive new research strategies.

Lack of integration with theoretical foundations

A large part of the theoretical research results on formal argumentation has not been examined from an application perspective. This applies in particular to the broad range of works that exist on argument strength[19] and argumentation principles.[16]

When deciding on the knowledge model, argument structure and strength, as well as on the argumentation reasoner, works that provide concise overviews of the corresponding basic research foundations should be considered in an as systematic manner as possible; also, the practical considerations that guide this process step in particular instances should inform theoretical research going forward and not only vice versa.

Lack of strong empirical research

Research on argumentation-based health information systems lacks strong empirical evaluation results. This is in contrast to research on traditional decision-support systems, but can potentially be explained by the relative novelty of the argumentation-based approach. Another possible reason for the lack of strong empirical evaluations of argumentation-based health information systems is the fact that the artificial intelligence researchers who are most engaged with the design and implementation of these systems typically lack experience in running long-term empirical assessments of complex sociotechnical systems. Hence, before the last stage of the design process, a handover of the developed system prototypes from artificial intelligence researchers to research groups that focus on the evaluation of information systems can be considered recommendable.

Short-lived software artifacts

A related problem is the lack of openly shared and well-documented software artifacts (as well as of commercial systems) that emerge from research on the topic. This hinders the development of systems as joined community efforts, and the study and application of the systems by information systems researchers who do not have low-level engineering expertise. To increase the chances that the developed systems' lifetimes do not end when the results are disseminated, a technology transfer plan should be integrated into the initial research outline. Such a plan can, for example, entail the handover of the artifact to healthcare authorities, or the extraction of generically useful system components as open-source libraries and frameworks. Realities in the healthcare sector that are prone to limit the adoption of new technology must be acknowledged, i.e., limited resources and healthcare professionals' tight schedules. Also, healthcare professionals may be skeptical when technology is introduced as a replacement and not an augmentation of human care.

CONCLUSION

The presented design methodology is a point of departure toward a more structured and rigorous application of argumentation-based approaches for intelligent health information systems. The methodology is an important tool to bridge the divide between the communities that advance the theoretical and engineering foundations of formal argumentation, and researchers who study the application of intelligent systems in the healthcare domain. We expect that the methodology will further mature as the body of research that studies the application of formal argumentation to health information systems growths. Also, the presented methodology can be transformed into a domain-independent model, or be adjusted to other system types.

ACKNOWLEDGEMENTS

The authors thank the anonymous referees. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation and by the European Union's Horizon 2020 programme, under Grant 825619.

REFERENCES

1. P. Baroni, D. M. Gabbay, M. Giacomin, and L. van der Torre, *Handbook of Formal Argumentation*. London, U.K.: College Publications, 2018.
2. L. Longo, B. Kane, and L. Hederman, "Argumentation theory in health care," in *Proc. 25th IEEE Int. Symp. Comput.-Based Med. Syst.*, 2012, pp. 1–6.
3. P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artif. Intell.*, vol. 77, no. 2, pp. 321–357, 1995.
4. J. Fox *et al.*, "Delivering clinical decision support services: There is nothing as practical as a good theory," *J. Biomed. Informat.*, vol. 43, no. 5, pp. 831–843, 2010.
5. R. Hoda, N. Salleh, and J. Grundy, "The rise and evolution of agile software development," *IEEE Software*, vol. 35, no. 5, pp. 58–63, Sep./Oct. 2018.
6. C. Yan, H. Lindgren, and J. C. Nieves, "A dialogue-based approach for dealing with uncertain and conflicting information in medical diagnosis," *Auton. Agents Multi-Agent Syst.*, vol. 32, no. 6, pp. 861–885, 2018. [Online]. Available: https://doi.org/10.1007/s10458-018-9396-x
7. E. Guerrero, J. C. Nieves, and H. Lindgren, "An activity-centric argumentation framework for assistive technology aimed at improving health," *Argument Comput.*, vol. 7, no. 1, pp. 5–33, 2016.
8. E. Guerrero, J. C. Nieves, M. Sandlund, and H. Lindgren, "Activity qualifiers using an argument-based construction," *Knowl. Inf. Syst.*, vol. 54, no. 3, pp. 633–658, 2018.
9. A. Q. Gill, "Agile enterprise architecture modelling: Evaluating the applicability and integration of six modelling standards," *Inf. Softw. Technol.*, vol. 67, pp. 196–206, 2015.
10. B. G. M. E. Blobel, K. Engel, and P. Pharow, " Semantic interoperability—Hl7 Version 3 compared to advanced architecture standards," *Methods Inf. Med.*, vol. 45, no. 4, pp. 343–353, 2006.
11. M. Lippi and P. Torroni, "Argument mining: A machine learning perspective," in *Theory and Applications of Formal Argumentation*, E. Black, S. Modgil, and N. Oren, Eds. Cham, Switzerland: Springer, 2015, pp. 163–176.
12. L. Amgoud and C. Cayrol, "Inferring from inconsistency in preference-based argumentation frameworks," *J. Automated Reasoning*, vol. 29, no. 2, pp. 125–169, 2002.

13. H. Li, N. Oren, and T. J. Norman, "Probabilistic argumentation frameworks," in *Theory and Applications of Formal Argumentation*, S. Modgil, N. Oren, and F. Toni, Eds. Berlin, Germany: Springer, 2012, pp. 1–16.
14. D. Walton and E. C. W. Krabbe, *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. Albany, NY, USA: State Univ. New York Press, 1995.
15. I. Rahwan and C. Reed, *The Argument Interchange Format*. Boston, MA, USA: Springer, 2009, pp. 383–402.
16. L. van der Torre and S. Vesic, "The principle-based approach to abstract argumentation semantics," *IfCoLog J. Logics Appl.*, vol. 4, no. 8, pp. 2735–2778, Oct. 2017.
17. P. E. Dunne and M. Wooldridge, *Complexity of Abstract Argumentation*. Boston, MA, USA: Springer, 2009, pp. 85–104. [Online]. Available: https://doi.org/10.1007/978-0-387-98197-0_5
18. A. J. Garcóa and G. R. Simari, "Defeasible logic programming: An argumentative approach," *Theory Pract. Logic Program.*, vol. 4, no. 1/2, pp. 95–138, 2004.
19. M. Beirlaen, J. Heyninck, P. Pardo, and C. Straûer, "Argument strength in formal argumentation," *J. Appl. Logics—IfCoLog J. Logics Appl.*, vol. 5, no. 3, pp. 629–675, 2018.
20. K. Čyras, D. Letsios, R. Misener, and F. Toni, "Argumentation for explainable scheduling," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 2752–2759.

HELENA LINDGREN is currently a Professor with Umeå University, Umeå, Sweden. She received the Ph.D. degree from Umeå University in 2007. She has combined clinical expertise with artificial intelligence research, for which she received the Nordea Scientific Award 2018. She serves in committees of several large-scale research programs on AI. Contact her at helena@cs.umu.se.

TIMOTHEUS KAMPIK is currently working toward the Ph.D. degree with Umeå University, Umeå, Sweden. His research interests are the intersection of non-monotonic reasoning and decision theory, and the engineering of multi-agent systems. He is an avid open-source software contributor and has a decade of experience in the enterprise software industry. Contact him at tkampik@cs.umu.se.

ESTEBAN GUERRERO ROSERO is currently a Researcher with Umeå University, Umeå, Sweden. His research interests include the theory and application of formal argumentation and multiagent systems. He received the Ph.D. degree from Umeå University in 2016. Contact him at esteban@cs.umu.se.

MADELEINE BLUSI is currently a Researcher with the Department of Community Medicine and Rehabilitation and an Associate Professor with the Department of Nursing, Umeå University, Umeå, Sweden. Her research interests include intelligent systems and patient-centered healthcare. She received the Ph.D. degree from Mid Sweden University, Östersund, Sweden, in 2014. Contact her at madeleine.blusi@umu.se.

JUAN CARLOS NIEVES is currently an Associate Professor with Umeå University, Umeå, Sweden. He received the Ph.D. degree from Universitat Politècnica de Catalunya—Barcelona Tech (UPC), Barcelona, Spain, in 2008. His research interests include the theory and application of nonmonotonic reasoning and multiagent systems. Contact him at juan.carlos.nieves@umu.se.