Rannsóknasjóður

2018

Antonios Achilleos

Epistemic Logic for Distributed Monitoring

The Icelandic Research Fund 2018

Postdoctoral fellowship – New proposal

Detailed project description

rannís

Guidelines

Please note that *all proposals and appendices* should be in English.

The highlighted text (<grey>) in this document is provided for guidance purposes only, and requested information should be added by applicant. Please insert appropriate material (i.e. text, pictures and/or tables) and add the principal investigator's name to the header and the project's name to the footer.

The detailed project description should provide the following information, and be divided into the following sections (the order and titles should not be changed):

a. *Specific aims of the project, research questions/hypotheses, feasibility, originality and impact*
b. *Present state of knowledge in the field*
c. *Research plan (time and work plan, methodology, milestones, present status of project, etc.) and deliverables. Explain if consents and/or permits are needed*
d. *Management and co-operation (domestic/foreign)*
e. *Proposed publication of results and data storage (including open access policy)*
f. *Contribution of doctoral and master's degree students to the project*
g. *Career development plan*

The project description should not exceed **12 pages** plus front page and guidelines (1.5 line spacing, 12 point Times/Times New Roman, or similar, 2.5 cm side margins). The bibliography is submitted in a separate file with no length limits. Convert the file to pdf before submitting. Please note:

→ ***Incomplete proposals will be rejected***
→ ***Proposals which exceed length limitations will be rejected without review***
→ ***Corrections or amendments after the previously announced deadline will be rejected***
→ ***The clarity and the overall quality of the presentation are taken into consideration when proposals are reviewed***
→ ***Applications that are submitted not using the most recent application form available will be rejected***

For further information please refer to the ***Icelandic Research Fund Handbook 2018.***

Please do not delete, overwrite, or amend this Guideline page in your submission

a. SPECIFIC AIMS OF THE PROJECT, RESEARCH QUESTIONS/HYPOTHESES, FEASIBILITY, ORIGINALITY AND IMPACT

Our goal for this project is to develop a framework for distributed runtime verification with monitor components that can detect, tolerate, and extract information from failures of the monitored system *and* the monitoring system itself. This framework will be based on a logic that combines μHML to describe desired system behavior; Epistemic Modal Logic to describe the system components' knowledge of the global configuration of the monitored system; Justification Logic to track evidence; Abstract Argumentation Theory to resolve conflicting information; and a component to keep track of the order of events. The resulting logic and its variations and counterparts will naturally raise computability and complexity questions with respect to their satisfiability and model-checking problems, which we intend to solve.

Project Background

In the setting of Runtime Verification (RV), a simple computational machine called a monitor is used to observe a system's behavior as the system runs [1]. For a certain property that is required of the system, the monitor may observe the violation or, possibly, the satisfaction of the property, in which case, it reports so. One of the goals of RV is to be able to automatically generate monitors from specification properties, as long as these properties are expressed in an appropriate formal language. A monitoring system is a family of monitors tasked to monitor for a certain specification language.

A strength of RV is that the monitoring system can usually be agnostic of the monitored system's structure, as a run of the system is abstracted as a possibly infinite sequence of events that interact with the environment, called actions. However, when the monitored system is distributed, we may need to use a distributed monitoring system to observe the system's behavior at different locations [2]. A distributed monitoring system may obtain a clearer picture of the observed system's behavior, but at the same time, additional challenges arise. These may include: additional overhead due to necessary communication between monitor components; balancing the computational cost among the monitor components; failures of the monitoring system itself due to faults of the communication medium; the possibility that the distributed system's topology may vary during runtime; and others. Furthermore, if a monitor component monitors the behavior of a monitored system component s, it makes sense that the behavior of s should be affected in a concrete way from the information that s has received from other system components and from its input.

Epistemic Modal Logic and μHML for Specifying Distributed Behavior

Specification languages for RV are usually variants of LTL and regular expressions (see, for example, [3, 4, 5]), and, more recently, μHML [6]. In particular, we consider μHML to be a particularly appropriate language for RV, as it is particularly expressive and (certain fragments of) it can be used to clearly specify monitor behavior, as Francalanza et al. showed in [6]. The formulas of μHML are formed by using the constants $\top, \bot$, variables, Boolean connectives, including negation, dual modalities $[\alpha], \langle\alpha\rangle$, where a is from a set of actions Act, and maximal and minimal fixed point operators max and min. We plan to use μHML as a base logic for RV. In a distributed setting, we want to extend μHML by combining it with a logic that can express the notion that a component is aware of certain pieces of information. For this purpose, we plan to use Multi-agent Epistemic Modal Logic.

Epistemic Logic provides a framework for studying knowledge and beliefs, especially when more than one agent is involved. Modal Logic is probably the most well-known family of logics used in Epistemology (see [7]). Its areas of application are numerous, ranging from Epistemology to Formal Verification and Artificial Intelligence. Modal Logic has been studied extensively (see [8]), from both a computational and a mathematical logic perspective. Modal formulas are formed by introducing the unary operators box (□) and diamond (◊) to the syntax of propositional logic; thus, if φ is a modal (or propositional) formula, then □φ and ◊ φ are also modal formulas. For Multi-agent Modal Logic, instead of □ and ◊ we have $\Box_i$ and $\Diamond_i$ for every agent i. Epistemic Modal Logic allows us to formalize notions of knowledge and belief when $\Box_i\varphi$ is interpreted as "agent i knows/believes that φ is true" and $\Diamond_i\ \varphi$ as "agent i considers φ to be possible" (or as “agent i does not know/believe that φ is false”). We note that μHML is a modal logic, though not an epistemic one. A property of Modal Logic that makes it popular for applications is that it is a flexible, expressive fragment of First-order Logic, with decidable provability (equivalently, satisfiability). Specifically, provability for the logics in this family is usually coNP-complete, PSPACE-complete, or EXP-complete [9, 10].

Our first goal is to develop a framework to examine the monitorability of distributed systems using a logic that combines epistemic operators and μHML. We consider Epistemic Modal Logic to be an appropriate language to express desired specification properties of distributed systems, paired with μHML, which provides a clear, expressive language for specifying the system’s behavior.

For example, consider a distributed system that may receive a request (we encode this event using the proposition req) and then output a reply (encoded as $resp$), as long as the

request has been approved by an appropriate component (encoded as apr). We want the component responsible for outputting the reply, which we call out, to give a reply only if it knows that a request was made and approved. We can formalize this requirement as

$$\varphi_1 = [resp]\,\Box_{out}\,(req \wedge apr).$$

In formula φ_1, $[resp]$ is an operator from μHML and is an action operator. In general, for action α, $[\alpha]\psi$ means that right after action α happens, property ψ must hold. Operation $\Box_{out}$ is a knowledge operator from Epistemic Modal Logic. For an agent a, $\Box_a\psi$ means that a knows that ψ holds. Therefore, formula φ_1 claims that if component out outputs a response, then it must know that a request has been made and it has been approved. For another example, we consider a system that includes a component h that is responsible for heating an area. We want to specify that if h knows that the temperature is less than 20 degrees Celsius, then it must heat the room. We may formalize this requirement as $\Box_h(temp \leq 20) \rightarrow heat$.

Distributed Monitoring Systems with Unreliable Components

One of the strengths of Epistemic Modal Logic is its ability to formalize and reason about higher-order epistemic statements (for example, 'A knows that B knows that A does not know φ'). Therefore, it can be used for expressing knowledge in monitoring systems that have components that monitor each other. In a distributed setting, another party may provide certain monitoring components and therefore these components may be deemed unreliable. In addition, a monitoring component's behavior may rely on the system that is monitored – especially if that system includes the communication medium of the monitors. In these cases, it makes sense to consider specifications for distributed monitoring systems, the most straightforward being that if a monitoring component is aware that the property it monitors for is not satisfied, then it should report it: $\Box_i\neg\phi_i \rightarrow report_i(\neg\phi_i)$.

We plan to consider distributed monitoring systems with possibly unreliable components. As we mention above, the unreliability of the monitoring components may be, for instance, the result of communication failures of the underlying network, or the result of another party providing certain monitoring components.

A Mechanism for Resolving Conflicts and a New Logical Framework

Another source of unreliability may be unreliable observations of the monitoring components. For example, a monitoring component may receive two contradicting reports that rely on different (and contradicting) measurements of the current temperature. It would then need to choose which of those reports is the most reliable. Therefore, that component

needs to be able to recover in each report the temperature measurement it relies on and then decide which one to keep. To confront such situations, we plan to build on two fields: *Abstract Argumentation Theory and Justification Logic.*

Abstract Argumentation Theory is a new, already successful field, which has its roots in Computer Science [11] and is strongly related to Nonmonotonic Reasoning, Informal Logic and Philosophy, with applications in Artificial Intelligence and Logic Programming, but also in fields like Law and Medicine. It introduces a clear framework for studying arguments and the way they can attack or support each other. Justification Logic can be thought of as an explicit version of Modal Logic, introducing justifications to Epistemology (see [12]), thus allowing one to track in an epistemic statement the reasoning and initial information behind it.

We propose to design and study a logical system that combines Argumentation Theory with Modal and Justification Logic. Such a system would allow us to formalize more accurately situations where the tracking of contradicting evidence is important, such as when a monitoring component receives contradicting reports, but also other situations of interest, as it is often the case both that beliefs are shaped by arguments and that arguments can be based on beliefs. Part of our research will be on determining the cost in computational time and memory of giving answers to questions posed in the language of this system – the Computational Complexity of the system, as it is important that the monitoring system introduces as little overhead as possible.

An argumentation framework is a pair $(A, \rightarrow)$, where A is a set of arguments and $\rightarrow$ is an attack relation between arguments (i.e. $a \rightarrow b$ indicates that argument a attacks argument b). Given such a structure of arguments, we consider particular sets of arguments that interest us. In the context of Argumentation Theory, propositions are not distinguished from arguments and these are often considered to be the same. The result is a very clear and useful presentation of the way arguments interact. Argumentation is based on the observation that we often do not have concrete undisputable proof for statements we believe, but we often have to form beliefs based on what we can support through convincing arguments (ones that can defend against attacking arguments by attacking those in turn by other arguments, etc.). This is often the case in Artificial Intelligence, but we can argue that this is also the case when we are working on foundational issues, where we need to accept certain basic principles on which to establish some theory (of course, without concrete proof).

Justification Logic is a relatively new field (see [12] for an overview), which started from [13] and complements Modal Logic, introducing justifications to modal epistemology. It

includes explicit versions of many epistemic modal logics, such as K, D, T, K4, D4, S4, which correspond to J, JD, JT, J4, JD4, and LP respectively. In Justification Logic, we have formulas of the form $t: \varphi$ instead of the modal $\Box\varphi$, where t is a term representing a justification for the necessity/knowledge/belief of φ. All theorems of one of these justification logics can be easily translated to theorems of the corresponding modal logic by replacing each term by a box (□) and vice-versa (not so easily) by replacing each box by an appropriate term.

An important axiom in Modal and Justification Logic is Negative Introspection. In its modal version it claims that if an agent i does not know (or believe, etc.) a statement, then i knows that they do not known the statement. The corresponding axiom in Justification Logic claims that whenever a justification is not acceptable for a statement, this fact is justified – in other words, the proof that p is not a proof of A is easy to extract: simply check that p is not a valid proof for A (which should be straightforward for a reasonable concept of proof). Negative Introspection is important, both because it is an axiom appropriate in many situations and because of its complexity properties. As Ladner [9] and Halpern and Rêgo [14] have demonstrated, when we add Negative Introspection to a modal logic, the complexity of its satisfiability problem drops to NP. On the other hand, for Justification Logic, the complexity, and even the decidability, of logics with Negative Introspection remains open.

The attack relation between arguments is the building principle in Argumentation Theory. With $a \to b$, what is declared is that a can be used as an argument that b is not a valid argument. Interestingly, this phenomenon is similar – yet not completely the same, of course – to Negative Introspection in Justification Logic. If b is not a valid justification (or argument) for a statement, then there is a justification a of this fact, i.e. $\neg b: \varphi \to a: \neg b: \varphi$ when expressed in the syntax of Justification Logic, which is roughly the form of Negative Introspection and can be thought as claiming that if b fails as an argument for φ, then there is an argument that attacks b.

The similarities become stronger when one considers modal logic $S4.2$ and its justification logic counterpart, $J4.2$, which was introduced and examined by Fitting [15]. This logic uses the axiom $\Diamond \Box \varphi \to \Box \Diamond \varphi$, equivalently $\Box\neg\Box\varphi \vee \Box\neg\Box\neg\varphi$, which can be thought as claiming that for statement φ, we either have an argument that φ cannot be supported by an argument, or that its negation cannot be supported by an argument. Here we call this axiom .2 for convenience. Its justification version, used in $J4.2$, is

$$f(t,u): \neg t: \varphi \vee g(t,u): \neg u: \neg\varphi.$$

Operations f and g seem to directly represent an attack relation between arguments. There is

an infinite family of modal logics, called Geach logics, with justification counterparts that are based on generalizations of axiom .2 [15].

We plan to introduce a framework that introduces Abstract Argumentation to Modal and Justification Logic. This would be a different approach from the ones that have already attempted to link argumentation to modal logic (ex. [16, 17, 18]), which are mostly semantic and their purpose is mainly to introduce the tools of modal logic to argumentation; furthermore, they assume a fixed and given argumentation framework, which is used to draw all conclusions. In the proposed approach, arguments would support propositions of the language; from these and perhaps other data, we can infer further conclusions. Given an argumentation framework, we may want to distinguish between arguments and propositions: there may be different arguments for the same proposition and the same argument might be used to support more than one claim – and it might only be convincing for some. We may also want to have arguments about arguments, which makes sense from human experience and follows the very successful practice of Epistemology (where we often encounter assertions of interest that someone knows that someone else knows – or not – something). Justification Logic offers an appropriate mechanism to accommodate such situations.

In the context of Distributed RV, a monitoring component may receive two (or more) reports, together with their justifications, from other monitoring components. It would need to determine whether the reports are in conflict with each other; if they are, it would have to determine, based on their justifications, which report is more reliable. For a simple example, we consider a distributed monitored system with the specification $(temp \leq 20) \rightarrow heat$ – that is, if the temperature is less than or equal to 20ºC, a heating component should be heating. Monitor i receives a report B that the heating component of the distributed system is not working correctly and another report G that it is working correctly. Quickly, i determines that the two reports contradict each other. By examining the justification of B, i determines that B's justification is based on the observation that meter u has determined that the temperature is 19ºC and that the heating component is not heating. By examining the justification of G, i determines that G's justification is based on the observation that meter r has determined that the temperature is 21ºC and that the heating component is not heating. According to the argumentation framework that the behaviour of i is based on, i knows that r is more reliable than u when the temperature is between 17ºC and 24ºC. Therefore, it determines that G is the more reliable report. An alternative specification could be

$$J(r):_h (temp \leq 20) \rightarrow heat,$$

which says that if the heating component has a reason to believe that the temperature is at most 20ºC and that reason is based on a report from r, and possibly other things, then it should be heating. Thus, u is ignored in this context. Of course, Argumentation Theory provides a framework to model more complex interactions between arguments/justifications.

Complexity Considerations

The following step will be to obtain complexity results about the new system, as well as variations of it, including logics with Negative Introspection and Geach logics. We expect several nontrivial results in this direction, as, based on preliminary investigations, we conjecture that, at least in several cases, adding Negative Introspection to a justification logic will increase its complexity. This would be the opposite phenomenon than the one we encounter in Modal Logic (where Negative Introspection drops the complexity of satisfiability from PSPACE to NP) and it would already demonstrate that it makes a lot of sense to examine this generalized Negative Introspection / attack relation closely.

Since we expect such a jump in complexity and since the reasoning that the monitors are expected to do should not introduce significant overhead, part of our effort will concentrate on identifying suitable fragments of the system with tractable complexity. Fortunately, as Artemov and Kuznets have demonstrated, Justification Logic has such fragments that are both expressive and in P [19, 20]. Therefore, we expect to identify appropriate fragments that we can use for the runtime verification of distributed systems.

The Importance of Time

Timed logics and automata play an important role in practical applications (see [21], for example). This is even more the case for Distributed RV, as it is important for monitoring components to keep track of the timing of events. In a non-distributed setting, a monitor that observes the interactions of the system with an environment can observe the order of events. On the other hand, in a distributed setting, a monitoring component can only infer the order of reported events from their timestamps. These timestamps can be generated by using a timing mechanism. Therefore, the logical systems and monitoring frameworks we develop will need to include such a timing mechanism.

Note: a Specification Language for RV vs. a Reasoning Framework for Monitors

We observe that in the description of the logical framework that we plan to develop, the Justification Logic part seems to be used as a mechanism for the monitors to reason about

their situation, more than as part of the specification that generates monitors. Alternatively, the monitors are expected to judge whether a set of reports is consistent or not and then derive the resulting information. This operation is closer to satisfiability checking and belief revision than to the usual operation of a monitor that mainly verifies a specification property for a certain process/trace. This phenomenon is the result of the syntactic nature of Justification Logic, but also of reasoning under uncertainty and handling unreliable information. We are confident that many interesting observations are bound to result from this research avenue.

Originality and Impact

The proposed research touches on several fields, the focus being on Distributed Runtime Verification. Since the main goal is the development of logics that can be used as specification languages, other subareas of formal verification will benefit as well. The approach we take, following [6], is that the specification language should be oblivious, in general, of the verification technique that is used. Therefore, the specification logics that we will introduce should be significant for the field of Formal Verification in general. To our knowledge, a combination of μHML and Epistemic Modal Logic has not been used before for Distributed RV and although Justification Logic has been combined with LTL and other logics used for formal verification, no RV framework has been developed for such logics.

Our project will also shed light on Abstract Argumentation Theory and Justification Logic, by examining how justifications that are, at the same time, arguments may behave. Abstract Argumentation Theory provides methods for resolving conflicting justifications and Justification Logic provides a framework for examining arguments for statements about arguments. Furthermore, by studying the complexity of the resulting system, we hope to illuminate the effect of the generalized negative introspection/attack relation on the complexity of a logic. The decidability of justification logics with negative introspection is still open, so any result we can achieve on this front will be of interest to the field of Justification Logic. On the other hand, even if we do not completely solve this open problem, for our purposes of constructing a system used for distributed RV, a lightweight version of the system would suffice – and indeed, would be preferable. For such a lightweight system, presumably it would be easier to extract complexity bounds.

We expect to fund an MSc student through this project to work on implementing a tool for distributed monitoring, based on the results we will have produced. This will increase the visibility and usability of the project's research.

b. PRESENT STATE OF KNOWLEDGE IN THE FIELD

Specification properties for RV are usually expressed in a logic such as LTL [22, 23, 24, 3, 4], CTL, CTL* [5], and μHML [25, 26, 27, 6]. Distributed RV frameworks have been proposed and examined, spanning several approaches and assumption (see [28] for a taxonomy and [2] for a review). Francalanza et al. in [28] present a general unifying framework for distributed Monitorability, they introduce a taxonomy of distributed RV approaches, and they identify several important challenges of distributed RV.

In [29], Sen et al. present a logic that combines LTL with epistemic elements, although the epistemic component is limited to describing a component's prediction of the global configuration. Their model of distributed monitoring focuses on multi-threaded systems, instead of distributed in general, and their monitoring components communicate through a shared memory. In [30], Colombo and Falcone present a method to generate a distributed monitoring system from an LTL formula. Their monitor components communicate over a distributed network that is expected to use a global clock – see also [31] for an alternative approach. Graf et al. in [32] use a knowledge-based approach to distributed monitoring that uses information derived from first running a model-checking procedure. Although Graf et al. do not explicitly use Epistemic Modal Logic in [32], their semantics for knowledge assertions correspond to the modal logic $S5$. In [33], Bonakdarpour et al. introduce a multi-valued family of logics based on LTL for distributed monitoring that tolerates monitor failures. The authors demonstrate that the multiple truth-values are necessary and a result of the incomplete picture of the system that each monitor has – see also [34, 35, 36].

Baltag et al. introduce a logic that combines Dynamic Epistemic Logic, Justification Logic, and Argumentation in [37]. Their logic is capable of handling belief revision in the presence of justifications. It is worthwhile to note that their logic includes a temporal operator Y, where $Y\varphi$ means that before the last epistemic operation, φ was true. Letia and Groza present a distributed justification logic for agents that use an argumentation framework in [38] – see also [39]. Their logic brings a good combination of the concepts we require, but it lacks explicit interactions between different agents. Yavorskaya introduced such agent interactions for variations of a two-agent justification logic in [40]. The proposer then expanded Yavorskaya's two-agent logic to a family of multi-agent justification logics with interactions between the agents and studied the complexity of these logics [41, 42, 43] – see [44] for an overview.

Renne in [45, 46, 47] introduces Dynamic Epistemic Logic, multiple agents and public announcement to Justification Logic; Renne's system is able to handle a kind of belief

revision, as certain justifications may not be valid after an update. Studer in [48] uses a multi-agent justification logic to verify non-repudiation properties of communication protocols. Fitting introduced Justification Logic counterparts for Geach logics in [15]. He proved realization, and soundness and completeness with respect to their semantics.

The decidability of the satisfiability/provability problem for justification logics with negative introspection remains open – it is not clear what the model-checking problem for Justification Logic would look like, as it is not clear how one would finitely represent a model for Justification Logic, as these have infinite components. Studer demonstrated in [49] that if only a finite number of axiom instances are justified, then justification logic with negative introspection is decidable. To the best of our knowledge, there are no non-trivial complexity bounds known even for these cases and the complexity of Geach logics and of logics that combine Justification Logic and Argumentation has not been investigated yet.

c. Research plan (time and work plan, methodology, milestones, present status of project, etc.) and deliverables. Explain if consents and/or permits are needed

During the first year of the project, we will focus on constructing a Distributed RV framework that uses Epistemic Modal Logic and μHML. Within this framework, monitoring components will be able to monitor for failures of the monitoring system. This framework is a logical first step and it will serve as a guide for what follows. Then, we will develop a timed version of this framework. We expect at least two publications to result from this research.
Milestone: A framework for distributed RV with monitors that detect faults of the monitoring system and track the time, using Epistemic Modal Logic.

During the second year of the project, we will focus on constructing a variation of the first year's distributed RV, using Justification Logic and a simple form of argumentation to handle inconsistent information. We will examine the corresponding decidability and complexity issues of the specification logics, and prove complexity bounds for these logics and their fragments that are appropriate for our framework. As we expect the complexity of these logics to be higher than what would be practical in RV, this process will help us identify appropriate fragments that we can use in an RV setting. We expect at least three publications to result from this year's research.
Milestone: A framework for distributed RV, based on Epistemic Modal Logic, Justification Logic, and a simple form of Argumentation, with monitors that can handle failures and

inconsistent information; decidability and complexity results for the resulting logics; identification of suitable fragments.

By the third year of the project, we will already have a clearer picture of the computational properties of the specification logic required for our distributed RV framework. During the third year, we will focus on developing a more complete distributed RV framework. Our goal will be to be able to argue that our resulting framework is optimal under reasonable criteria. At the same time, work on the implementation of our framework will begin, with the contribution of an MSc student. We expect at least two publications and a prototype tool to result from the third year's work.

Milestone: A full framework for distributed RV, based on Epistemic Modal Logic, Justification Logic, and Argumentation Theory, optimal under reasonable criteria; development of a prototype tool.

d. MANAGEMENT AND CO-OPERATION (DOMESTIC/FOREIGN)

There is already a running collaboration between the proposer and the following researchers, who will participate in the project:

Prof Luca Aceto at Reykjavik University

Prof Anna Ingólfsdóttir at Reykjavik University

Dr Adrian Francalanza at the University of Malta

Their collaboration started in September 2016, when the proposer was hired as a postdoctoral researcher at Reykjavik University for one year to work on the project "*TheoFoM: Theoretical Foundations of Monitorability*". This collaboration has proven successful, as it has already resulted in one paper published in a conference with formal proceedings with a journal version under submission, one more to be presented in a conference without formal proceedings for now, one paper under submission, and one more working paper.

Luca Aceto and Anna Ingólfsdóttir are experts in Computational Logic and Formal Verification (among other fields), with an ongoing collaboration that spans many years. Adrian Francalanza is an expert in the field of Runtime Verification with contributions to distributed RV. The proposer has acquired significant experience during his PhD studies on Multi-agent Justification Logic and Epistemic Modal Logic, as well as on relevant complexity-theoretic issues. Under the supervision of Prof Sergei Artemov, he developed a Multi-agent Justification Logic framework (based on Yavorskaya's two-agent logic of proofs [40]) and proved several complexity results [50, 41, 42, 51, 43, 44], answering two previously

open questions in the process [50, 43]. During his year as a postdoctoral researcher at Reykjavik University, he has also gained experience in the theory of Runtime Verification.

e. Proposed publication of results and data storage (including adherence to open access policy)

We intend to present our results to relevant conferences and journals of the field. To ensure accessibility to the results, we plan to publish updated versions of our papers on the arXiv, the open access depository. We will also post a webpage for the project on the proposer's website with updates and links to the publications that will result from the research for the project.

f. Contribution of doctoral and master's degree students to the project

An MSc student is expected to work on partly implementing the resulting framework. The student will cooperate with PhD students Duncan Paul Attard and Ian Cassar, who are pursuing a joint PhD degree at Reykjavik University and the University of Malta and who have experience working on the runtime monitoring tool DetectER [52, 53]. This should be an especially good learning opportunity for the student, as the project touches on several different areas of research.

g. Career Development Plan

As part of pursuing his PhD, the proposer has worked on Complexity issues for Multi-agent Justification Logic [50, 41, 43, 42, 44]. At the same time, he published results on the complexity of Modal Logic [51, 54, 55, 56]. This experience has given him significant expertise in these fields. Since the proposer was hired as a postdoctoral researcher at Reykjavik University, to work on the project "*TheoFoM: Theoretical Foundations of Monitorability*", he also gained experience in the theory of Runtime Verification [57, 58, 59, 60], which has expanded his area of research. The proposed project touches on several areas of research. The proposer already has experience with many of these, but he will have to learn about others, as well. Furthermore, working on the interactions of these areas will be an additional gain of the project. In pursuit of an academic career, the publications, knowledge, and experience that the proposer will gain from the project will be invaluable, especially since the proposer plans to keep doing research on related areas.