Using I* For An Early Analysis Of Interoperability Requirements

Roberto de Cerqueira Figueiredo[1] [0009-0002-6516-4754], Daniela Barreiro Claro[1] [0000-0001-8586-1042], Rita Suzana Pitangueira Maciel[2] [0000-0003-3159-6065], and Julio Cesar Sampaio do Prado Leite[2] [0000-0002-0355-0265]

[1]FORMAS Research Center on Data and Natural Language – Institute of Computing – Federal University of Bahia, Salvador, Bahia, Brazil.
[2] Institute of Computing – Federal University of Bahia, Salvador, Bahia, Brazil.
(roberedo,dclaro,rita.suzana,julioleite)@ufba.br

Abstract. Interoperability is a desired quality when there are cooperating software components. Usually, interoperability is classified into three levels: syntactic, semantic, and pragmatic. The pragmatic level is challenging as it requires modeling contexts and intentions. Over the works we reviewed, some tackle intentions, others context, at different levels of abstraction. Our contribution lies in dealing with intentions and context based on distributed intentionality among the different cooperating components. Pragmatic interoperability depends on use, which varies depending on context and intention. As such, the notion of variability is key in dealing with this type of interoperability. This paper explores the early analysis of interoperability from the standpoint of a goal-based modeling strategy. Using a real case, we show how the models in i* allow for an early analysis of pragmatic interoperability, as modeled by intentions and contexts. A University campus surveillance case helped us to show how modeling helps in the derivation of pragmatic interoperability requirements.

Keywords: Pragmatic interoperability · Goal oriented requirement engineering · Intentional modeling.

1 Introduction

Software components need to cooperate to achieve desired goals, which they could not achieve alone. Effective interaction occurs when we have interoperability among components. We achieve interoperability when software components exchange and effectively use shared information.

Usually, interoperability is classified into three levels: syntactic, semantic, and pragmatic. According to Huang [1], the syntactic level is the formal relation between parts, the semantic level is upgraded with the denoted relation, and the pragmatic level addresses the use the parts employ within the relations.

As argued by [2], providing interoperability by considering the format and meaning (i.e., syntax and semantics) in data exchange is insufficient to achieve

Proceedings of the 27th Workshop on Requirements Engineering (WER24), August 07-09, 2024, Buenos Aires, Argentina.
DOI: 10.29327/1407529.27-17

complete, adequate, and meaningful cooperation. In this context, pragmatic interoperability is critical to meet the desired effects during message exchange among cooperating components.

Interoperability on syntactic level [3] and semantic [4] is well known. Our focus is on pragmatic interoperability. Pragmatic interoperability is a challenge as it requires modeling intentions and context [2],[5],[6]. Recently, Ribeiro et al. [7] proposed a conceptual framework for pragmatic interoperability (CAPITAL), which defines and structures pragmatic interoperability elements. CAPITAL addresses the aspects of environment, contexts, and intentions, with particular emphasis on the context part but less emphasis on modeling intentions.

Our main goal is to provide a way to map intentionality to achieve pragmatic interoperability. We delve into the contextual and intentional requirements, modeling them with a goal-oriented language. Asuncion [5] points out that some authors distinguish intention and context, while others combine them. We adopt the latter approach, using the i* language to allow actors (components) to define the context and the intentional elements of the i* to define intention.

Our research takes a practical turn as we extract interoperability requirements from the goal-oriented model. To underscore the real-world relevance of our work, we tackle pragmatic interoperability issues in a university campus surveillance system. By producing models written in i* and deriving the necessary interoperability requirements, we pave the way for pragmatic interoperability among actors.

Using the IEEE Computer Dictionary [8], we combine the definitions of interoperability and requirements to state that interoperability requirements are the essential conditions or capabilities for two or more systems or components to exchange and use information. In our case, we specifically focus on interoperability requirements for pragmatic interoperability, providing a clear understanding of our research scope.

We approach interoperability from the standpoint of requirements engineering. Our choice was to anchor our research from the standpoint of GORE (Goal Oriented Requirements Engineering), which aims to focus on the intentionality of software before committing to the software functions, and in particular from the idea of distributed intentionality as proposed by Yu [9] in the goal-oriented language i*. As such, we aim to bring the interoperability concern before architectural choices in the software design by focusing on early requirements, particularly the interoperability requirements in the context of software components, modeled as i* actors.

We organized this paper as follows: Section 2 describes related works, Section 3 describes our research design choices, and Section 4 details i* models built for the university campus case. Section 5 evaluates the results, and Section 6 summarizes the contributions and points out for future work.

2 Related Works

In user-centered services, the user's goals and context are essential for helping achieve pragmatic interoperability between users with different intentions and enabling service-based applications [2].

We performed a literature survey to learn intention and context representations in the user-centered services approach. We discovered few works that deal with the issues of context and intention representation in user-centric services.

Najar et al. [10] proposed user-centric service discovery mechanisms based on user context and intention. The authors represented the intention as a tuple of (verb, target) extracted from the user request text. The user's intention is then compared with the service's intention, and a matching mechanism calculates the similarity. It uses an ontology to reason through the words in the text.

Daosabah et al. [11] proposed a dynamic web service composition mechanism. It transformed the web service composition problem into an AI planning problem. The author aims to find an optimal composition plan that achieves a goal (intention) according to the problem domain description (context). The user requests services through a textual declaration. Intentional data is extracted from the text and categorized into verbs, goals (target), and functional and non-functional constraints. It uses an ontology to reason through the words in the text, and the AI planning method combines service inputs, outputs, preconditions, and effects to find a composition that satisfies the user's intention in the context.

These works do not use a requirements modeling language to represent intention and context. GORE modeling is important because it emphasizes understanding and modeling stakeholders' goals. As such, there is a need for studies on requirements engineering because the absence of a modeling language can lead to the definition of inconsistent requirements, such as producing contradictory requirements.

Secondly, motivated by the lack of approaches that use GORE in user-centered services, we looked for work [2] that proposed such modeling that contributes to this aim.

Qureshi et al. [2] look at interoperability at various levels of abstraction: users, services, and resources. The focus is variability and its effects on the requirements for interoperability. The authors also focused on the human dimension, identifying the high-level goals of the actors. Using goal-oriented, they analyzed intentional elements (goal, softgoal, resources). During this analysis, the notion of variability that exists between actors' goals is made more explicit. The proposed guide helps the analyst to discover and specify interoperability requirements. However, using different representation schemes (a goal model, similar to I* and BPMN, a process model) leads to more work as requirements engineers must use two languages. It also does not deal with the concept of means-end, so there is no representation of tasks. Similar to our work, the concept of variability is central.

As such, we aim to bring the interoperability concern before architectural choices in the software design, that is we need to be aware of possible requirements variants, thus dealing with variability. We do so, by focusing on early requirements, particularly the interoperability requirements in the context of software components, modeled as i* actors.

Our contribution lies in dealing with intentions and context based on a distributed responsibility among the different actors that will interoperate to provide an early analysis of the desired pragmatic interoperability requirements. Our proposal uses a more holistic approach by focusing on the taxonomy of actors, identifying the significant interactions, and describing the rationale of each component (actor) as to how they deal with different situations to maintain interoperability with cooperation actors. By focusing on the means-end relationship, the i* SR model can identify requirements early, describing tasks and their impact on the desired qualities (softgoals).

3 Research Design

Our work was based on a real case from the university campus surveillance, examining the implementation of an autonomous surveillance system and pursuing a better understanding of pragmatic interoperability.

We sought a modeling language focused on intentionality to comprehend the surveillance case and the significance of context and intentions in understanding pragmatic interoperability. After careful consideration, we opted for the original version of i*, recognizing the crucial role of the means-end relationship in gaining a deeper understanding of goal attainment variability. On top of that, the relations among actors (*plays*, *occupies*, and *is-a*) allow for variability as well. Consequently, we chose not to utilize the 2.0 version of i*. The i* version chosen for this research is the original [12]. The reason for avoiding the i* 2.0 [13] is that i* 2.0 does not allow us to indicate the means to reach a resource, and the relationship refinement, which allows the refinement of goals into tasks and vice versa, has semantics different from the means-end relationship. It is also the case that the actor relationships *plays* and *occupies* are not in the i* 2.0.

We have decided to focus on the actual operating mode of the campus surveillance systems, the AS-IS system, to understand the challenges the learning models would have when designing the future surveillance system. To better understand the softgoals, we also had to recur to the non-functional requirements (NFR), or quality requirements. For representing these qualities requirements we have used the NFR Framework [14], a NFR language.

4 Case Study On Surveillance Camera Operation

Our case study tackles a surveillance camera monitoring system to provide pragmatic interoperability. Such scenario aims to detect risk situations through video cameras(CCTV) on a university campus.

Our interaction with the university campus security sector identified the need to monitor the following risk situations: flooding, fire, smoke sources, detection of vehicle license plates, firearm possession, knife possession, and people with suspicious attitudes near vehicles.

In some meetings we had with CCTV sector managers at the university, we discovered that there are factors influencing CCTV monitors' decision-making. It was considered a significant obstacle to effectively classifying risk situations. Piza and Moton [15] realized a study about such factors. They performed a systematic social observation of CCTV monitor's activity and produced a taxonomy of factors. Table 1 shows these factors, and we selected two (rank and visible obstructions) for modeling. Rank means monitors' experience related to the ability to evaluate an individual's intention to harm. Visible obstructions refer to the obstacles in CCTV images that impact the monitor's decision-making. We chose these two situations because they represent cases with variability between actors.

Table 1. Factors that influencing CCTV monitors' decision-making

Number	Factors	Chosen for modeling
1	Surveillance targets	
2	CCTV site	
3	Visible obstructions	X
4	Operator rank	X
5	Operator gender	

We model these situations using the i* language. We simplified de Souza Cunha's [16] modeling process and modeled the actor's intentionality in the scenario as illustrated in Figure 1.

Fig. 1. Process to modeling the campus scenario.

In our case, although we have used the information provided by experience with human agents [15], we understand that intelligent services can be a proxy for these agents. So, these agents in the i* model are components. Our future goal is to evolve these components into intelligent services with more different experience levels using neural network methods and different datasets because artificial intelligence through deep learning allows designing the monitor's behavior.

The following subsections detail the resulting models, which have been improved over four meetings among the co-authors.

4.1 Identify And Refine Actors

The first step is the actor's identification and refinement. We use the SA (Strategic Actor) model[17]. The SA model aims to model actors and the relationships between them. The elements of actor modeling that we used, based on [9], were:

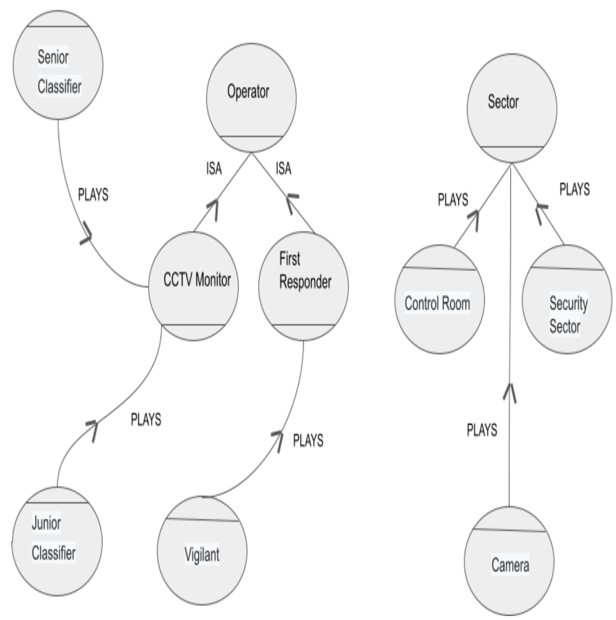

Fig. 2. A SA model of campus scenario. The system has six actors: the control room, cameras, senior classifier, junior classifier, security sector, and vigilant.

- Actor: "An actor is an active entity that performs actions to achieve their goals through their knowledge".
- Role: "A role is an abstract characterization of the behavior of a social actor in some specialized context or domain".
- Agent: "An agent is an actor with physical, concrete manifestations". The term agent is used in place of a person to generalize and can refer to both human agents and artificial (hardware/software).

The relationships we used, based on [16] were:

- Plays: "agent-to-role relationship : an agent can play more than one role, and a role can be played by more of an agent".
- Is a: "actor-to-actor relationship : of role for the role; from agent to agent; actor, role, and agent can be specialized by more than one actor, role, and agent respectively".

We created a SA model, as illustrated in Figure 2, with the actors that seem relevant in the case of the University surveillance. Senior and Junior classifiers carry out the main action of the scenario, showing a variability on playing the role of CCTV Monitor. They monitor images from surveillance cameras to detect risk situations.

The SA model (Fig 2) may be described by: The CCTV Monitor role is an Operator role; the First Responder role is an Operator role; a junior classifier plays the role of CCTV monitor; a senior classifier plays the role of CCTV monitor; a vigilant plays the role of First Responder; a control room plays the role of a Sector; a security sector plays the role of a Sector; and a camera plays the role of a Sector.

4.2 Identify Dependencies Between Actors

The next step is to identify the dependencies between actors. A dependency represents an agreement between two actors, where one actor (*depender*) depends on another actor (*dependee*) so that actors can achieve a goal, perform a task, make available a resource, and reasonably satisfy a flexible goal (softgoal).

We used resource, goal and task types of dependencies based on [16]:

- Resource dependency: "occurs when an actor (*depender*) depends on another the (*dependee*) so that an entity (physical or logical) is made available" .
- Goal dependency: "occurs when *depender* depends on *dependee* for the achievement of a specific state of the world".
- Task dependency: "occurs when an actor the (*depender*) depends on another the (*dependee*) so that this other can perform a task".
- softgoal dependency: "occurs when an actor (or *depender*) depends on another (*depender*) to perform some task for a flexible goal to be "satisfied [1] satisfactorily" or "reasonably satisfied", that is, satisfied at a level considered acceptable ".

Four main dependencies exist between actors: the control room and cameras, the control room and classifiers, the security sector and control room, security sector and vigilant, as illustrated in Fig 3.

We described the SD model (Fig 3) as: The Control Room agent depends on the Junior and Senior classifiers to achieve "incident be classified" goal; The Control Room agent depends on the Junior and Senior classifiers to receive "Labeled image" resource; The Senior and junior classifiers depends on Control Room to receive "Video stream" resource; The Security Sector agent depends on the Vigilant agent to carry on "act on the incident" task; The Control Room agent depends on the Camera agent to receive "Image" resource; The Security Sector agent depends on the Control Room agent to carry on "send alert" task.

[1] "Simon[**?**] uses the term "satisfice," which is central to his theory of behavioral decision, to mean degrees of how a nonoptimal solution may be accepted."

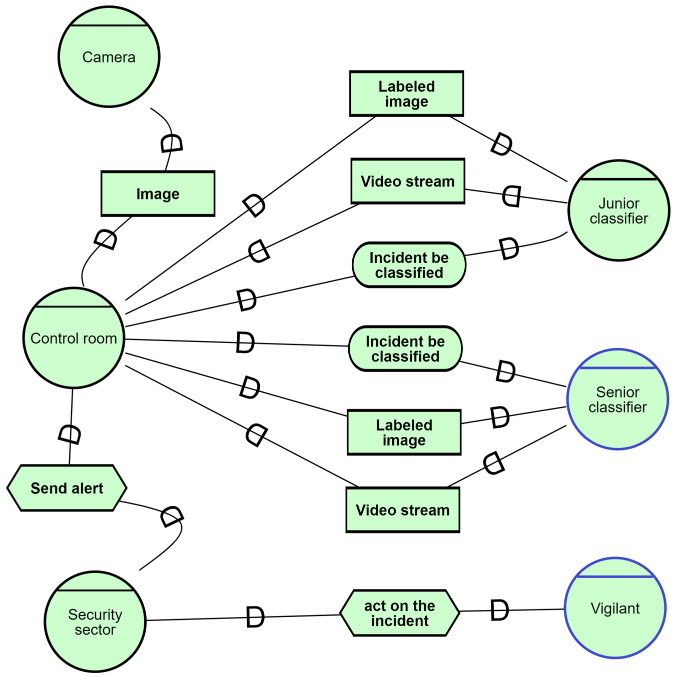

Fig. 3. A SD model of campus scenario.

4.3 Identify Goals, Softgoals, Tasks, And Resources

In the previous section, we represented the dependencies between actors in the campus surveillance system. The dependencies between the control room and the classifiers represent variability. This section will produce the SR model to describe such variability. We incorporate intentional elements, as illustrated in Fig 4, in the SR model to characterize such variability.

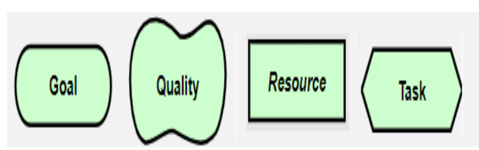

Fig. 4. Intentional elements.

Building an SR model involves defining softgoals and other elements. Souza Cunha [16] recommends finding terminology for expressing softgoals (quality)

[1] Note that in Fig 3 we have the agents that play the role of CCTV as depicted in Fig. 2.

through catalogs proposed by Chung [14]. We expressed softgoals like specific subtypes, such as response time, and the accuracy of abstract types, such as performance and security. Besides, we expressed the senior and junior classifiers' experience through workability [9] softgoal.

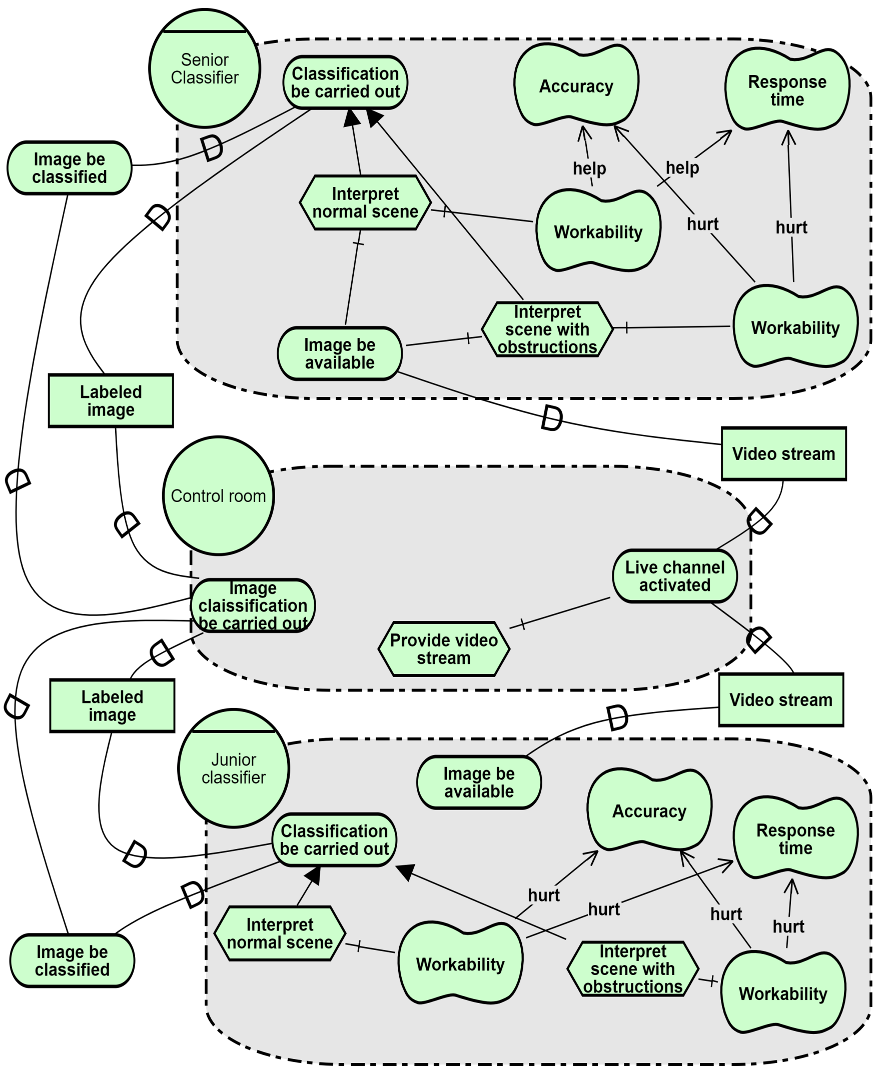

Fig. 5. A SR model of relation between image surveillance operators and control room.

Based on Table 1, we characterize Factors that influence CCTV operators' decision-making: Operator rank through variability between agents, considering Workability, the softgoal that represents the experience of each operator and impacts the other softgoals: accuracy and response time; Visible obstruction factor represent variability between tasks given that we have a possibility of execute a task of interpreting usual scene and a task of interpret scene with obstruction, both tasks impact differently in accuracy and response time softgoals. We model the SR model of these two situations, as illustrated in Fig 5.

The following descriptions represent SR elements (Fig 5): The Senior Classifier agent has the “Response time” and “Accuracy” softgoals; The means to achieve “classification be carried out” goal by the Senior Classifier agent are the “interpret normal scene” and “interpret scene with obstructions” tasks; The “interpret normal scene” task of the Senior Classifier agent contributes to (help) The “response time” softgoal, and contributes to (help) the “Accuracy” softgoal; The “interpret scene with obstructions” task of the Senior Classifier agent contributes to (hurt) The “response time” softgoal, and contributes to (hurt) the “Accuracy” softgoal; The Junior Classifier agent has the “Response time” and “Accuracy” softgoals; The means to achieve “classification be carried out” goal by the Junior Classifier agent are the “interpret normal scene” and ”interpret scene with obstructions” tasks; The ”interpret normal scene” task of the Junior Classifier agent contributes to (hurt) The ”response time” softgoal, and contributes to (hurt) the ”Accuracy” softgoal; The ”interpret scene with obstructions” task of the Junior Classifier agent contributes to (hurt) The ”response time” softgoal, and contributes to (hurt) the ”Accuracy” softgoal.

4.4 Extracting Interoperability Requirements From I*

To validate whether the system is pragmatically interoperable, we need to define and analyze the intentions and context presented by the agents. By modeling intention and context, we can better cover pragmatic interoperability. We can extract pragmatic interoperability requirements from this analysis through the dependencies observed between actors as shown in the table 2.

Table 2. Intentional interoperability requirements.

Interoperability requirements				
id	Depender	Dependee	Dependum	Interoperability
1	Control Room	Senior Operator	image be classified	I1= {wrk,cco, rt, acc}
2	Control Room	Junior Operator	image be classified	I2= {wrk,cco, rt, acc}
3	Senior Operator	Control Room	video stream	I3 = {vs}
4	Junior Operator	Control Room	video stream	I4= {vs}
5	Control Room	Junior Operator	labeled image	I5= {li}
6	Control Room	Senior Operator	labeled image	I6= {li}

This table 2 shows that the dependencies between actors produce elements necessary to achieve interoperability. Dependency 1 and 2 produced the goal cco (classification be carried out) and the softgoals web (workability), rt (response time), and acc (accuracy), which are the basis for the pragmatic interoperability. Dependencies 3 and 4 produced the vs resource (video stream), and dependencies 5 and 7 produced the vs resource (image labeled); these are the basis for the syntactic interoperability produced by tasks of both actors (semantic interoperability). As such, the IDs I1 and I2 are the sources for the derivation of the pragmatic interoperability requirements, as can be seen below:

- RF1 The system must produce the classification of the video stream.
- RNF1 The junior classifier must classify the video stream with accuracy according to his workability
- RNF2 The senior classifier must classify the video stream with accuracy according to his workability
- RNF3 The junior classifier must classify the video stream with response time according to his workability
- RNF4 The senior classifier must classify the video stream with response time according to his workability

The IDs I3, I4, I5, and I6 are the source for deriving the syntactic and semantic interoperability requirements. As an example, we list one of each type below:

- RF2 The control room component must send the video stream to the junior classifier component.
- RF3 The junior classifier must interpret the normal scene to produce the labeled image.

Our work, which uses the i* elements with the interoperability perspective, has significant practical implications. By interpreting resources as syntactic elements for syntactic interoperability, tasks as semantic elements for semantic interoperability, and goals and softgoals as pragmatic elements for pragmatic interoperability, we pave the way for enhanced system integration and communication.

5 Evaluating The Results

5.1 The Requirements

The requirements outlined above underscore the pivotal role of i* models in pinpointing requirements crucial to pragmatic interoperability. In the case of RF1, it becomes evident that this requirement encapsulates the entire proposal for the interoperability of the three components (Control Room, Senior Classifier, Junior Classifier). The other requirements, RNF1, RNF2, RNF3, and RNF4 are requirements that take into account the variations in the workability of the two classifier components, thereby illustrating that the use of different components can lead to diverse implementations of the RF1. This achievement was possible because we factored in the overarching intent of the cooperation and the distinct uses of the video stream in two different contexts.

We also considered the agent's experience or workability in both junior and senior classifiers. The level of workability has a direct bearing on the attainment of accuracy and response time. Therefore, by preparing the overall system to handle different uses of the video stream, we are addressing the interoperability requirements that would vary depending on the context and the intentions at hand, i.e., the pragmatic interoperability.

5.2 The Model

Our research reused a literature result [15] related to surveillance scenarios in CCTV monitors, which led to the revelation of variability between operators, a crucial aspect of service quality. We were able to model this variability as demonstrated in Fig 2, using agents that simulate the role of CCTV monitors. Moreover, we modeled the overall intention of cooperation between the Control Room and the CCTV monitor, as well as the inherent qualities (intentions) of the qualifiers' context, selecting three qualities for this case. One quality reflects the experience, while the other two represent well-known qualities related to video classification.

The series of models that we designed showed at different levels their capability of dealing with situations that may arise in a set of interacting components. The fact that i* models interactions between actors and their refinements as agents and roles showed that understanding the components as actors allows the representation of the flexibility and variability in the campus surveillance example. As actors, understood as components, interact among themselves, i* represents the interaction as dependencies. Each of these dependencies allowed us to model different situations that we found in the campus case, so that an SD diagram could capture the main interactions devised in the campus case regarding the cameras.

The need to describe the rationale of each actor, that is, the context where the actor would perform to attain the intention of the depender, led to the SR diagram. In this diagram, the use of the means-end relationship model the impacts the quality goals (softgoals) suffer from the different tasks carried out by different agents (senior and junior classifiers). It is a way of demonstrating the power of the representation, which guides the modelers to identify and model different situations and their impact on the component (actor) and the interaction with other components (actors), allowing for an early analysis of the interoperability of the components ensemble.

5.3 The Result

Considering the above, we grappled with the challenges posed by different authors [6] ,[5],[7] regarding the aim of pragmatic interoperability. These authors rightly recognized that dealing with intentions and contexts is fundamental to handling potential different uses of information when distinct components interoperate (pragmatic interoperability). As such, having an SLA (Service Level Agreement) between components that deals only with syntactic and semantic information is not sufficient. There is a need to consider possible variations in the use of the shared information. However, when focusing on intentions and contexts, it is possible to cover more ground in identifying the possible uses of shared information. Therefore, our paper not only presents the use of i* models to derive requirements for pragmatic interoperability but also highlights the complexity and depth of the challenges we scratched in our research.

6 Conclusions

Our research aimed to improve the analysis of pragmatic interoperability among software components early on at the requirements level. Our exploration of the capabilities of i* as a modeling language allows us to reason about different situations of a pragmatic nature related to how other components may use them.

Using a real case in the surveillance domain, we explored the particular context of the University surveillance, which aims to automate their system by focusing on the AS-IS context.

Our paper shows that intentional modeling, through i* models, helps attain the challenges for deriving pragmatic interoperability requirements.

Further work will aim for an autonomous version (TO-BE) of the University case, concentrating on modeling, in i*, the agents that will compose the corpus to be used by the intelligent model for image classification.

Acknowledgements

Leite acknowledges the partial support from CNPq. This material is partially based upon work supported by the FAPESB under grant TIC 0002/2015. This material is partially based upon work supported by CAPES Financial code 001.

References

1. Huang, Y. (2007). Pragmatics. Oxford University Press.
2. Qureshi, N. A., Nguyen, C. D. & Perini, A. (2010). Analyzing interoperability requirements for adaptive service-based applications: A goal-oriented approach. 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, 239-244. https://doi.org/10.1109/COMPSACW.2010.49
3. Nasfi, R., Bronselaer, A., & De Tré, G. (2023). A novel approach to assess and improve syntactic interoperability in data integration. Information Processing & Management, 60, 103522. https://doi.org/10.1016/j.ipm.2023.103522
4. Davies, J., Welch, J., Milward, D., & Harris, S. (2020). A formal, scalable approach to semantic interoperability. Science of Computer Programming, 192, 102426. https://doi.org/10.1016/j.scico.2020.102426

5. Asuncion, C. H., & Van Sinderen, M. J. (2010). Pragmatic interoperability: A systematic review of published definitions. EAI2N 2010, Held as Part of WCC, 164-175.
6. Neiva, F. W., David, J. M. N., Braga, R., & Campos, F. (2016). Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. Information and Software Technology, 72, 137-150. https://doi.org/10.1016/j.infsof.2015.12.013
7. Ribeiro, E., Claro, D., & Maciel, R. (2021). Defining and providing pragmatic interoperability: The MIDAS middleware case. Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação, 122-137. https://doi.org/10.5753/sbsi.2021.15367
8. Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., & Springsteel, F. (1991). IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries. IEEE Press.
9. Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering, 226-235. https://doi.org/10.1109/ISRE.1997.566873
10. Najar, S., Pinheiro, M. K., & Souveyet, C. (2014). A new approach for service discovery and prediction on pervasive information system. The 5th International Conference on Ambient Systems, Networks and Technologies, 32, 421-428.
11. Daosabah, A., Guermah, H., & Nassar, M. (2023). User's intention and context as pertinent factors for optimal web service composition. Service Oriented Computing and Applications, 1-34. https://doi.org/10.1007/s11761-023-00380-w
12. Yu, E. (1995). Modelling Strategic Relationships for Process Reengineering (PhD thesis). Graduate Department of Computer Science, University of Toronto, Toronto, Canada.
13. Dalpiaz, F., Franch, X., & Horkoff, J. (2016). iStar 2.0 language guide. arXiv preprint arXiv:1605.07767.
14. Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers.
15. Piza, E. L., & Moton, L. N. (2023). Proactive monitoring and operator discretion: A systematic social observation of CCTV control room operations. Journal of Criminal Justice, 86, 102071. https://doi.org/10.1016/j.jcrimjus.2023.102071
16. de Souza Cunha, H. (2007). Uso de estratégias orientadas a metas para modelagem de requisitos de segurança (Master's thesis). PUC-Rio.
17. do Prado Leite, J. C. S., Werneck, V., Oliveira, A. P. A., Cappelli, C., Cerqueira, A. L. A., de Souza Cunha, H., & Gonzalez-Baixauli, B. (2007). Understanding the strategic actor diagram: An exercise of meta modeling. WER, 2-12.