A Catalogue of KAOS Extensions

Pedro Rodrigues de Carvalho Junior[1], Enyo José Tavares Gonçalves[1], Rainara Maia Carvalho[1], and Marcos Antonio de Oliveira[1]

Universidade Federal do Ceará, Campus de Quixadá (UFC)
pedrojunior@alu.ufc.br, {enyo,rainara,marcos.oliveira}@ufc.br

Abstract. KAOS, a goal-based modeling language, has been extended since its creation. Searching for existing KAOS extensions and their constructs is a task that can be the start point for requirements modeling or for the creation of new KAOS extensions. This search can be performed by using strings searches in databases or through a catalogue that supports them. This exploratory task about extensions can take a great deal of time and be prone to failure when performed without specific and adequate support. catalogues have been used successfully to bring together a body of knowledge, including knowledge of modeling language extensions. Motivated by this scenario, this work proposes a catalogue of extensions to the KAOS modeling language. The results suggest that the proposed catalogue can be used to retrieve information about extensions and their constructs correctly and that it is easy to use.

Keywords: Requirements Engineering · KAOS · Extensions · Catalogue

1 Introduction

Modeling languages play an important role in Model-Based Engineering (MBE) and in Model-Driven Engineering MDE) [1].

KAOS (Knowledge Acquisition in autOmated Specification) [16] is a requirement modeling language that represents the system through its objectives, conflicts, obstacles, objects and the interaction between them [5]. KAOS uses a combination of four models: objective model, responsibility model, object model, and operating model.

Extending an ML is to add new constructs, modify or remove the existing ones [1]. An extension is generally used to i) model applications in the domain for which it was proposed and ii) be used as the basis for a new extension. Previous results [12] [15] identified 42 KAOS extensions.

Identifying a KAOS extension or its constructs based on some specific characteristic can be a complex task that requires a lot of time and is prone to fail when performed without adequate support. Two possible ways to perform this kind of task would be: i) create a search string, search in databases, select extensions, and analyze the results, or ii) analyze the papers made available by Systematic Literature Review (SLR) and try to identify the information you want.

Wiki, spreadsheet shared, and catalogue can be used to systematize the data presentation and easier the search and recovery tasks. We can highlight catalogues since they have often been used successfully to gather accumulated knowledge, including extensions of modeling languages. A catalogue allows search information about the papers and constructs, as well as the suggestion of new KAOS extensions by users.

This work proposes a web catalogue of the KAOS modeling language extensions to support the requirement engineers' work. We also evaluated the catalogue of KAOS extensions with users.

This work is part of a set of studies to analyze and systematize the KAOS extensions similarly as performed by Gonçalves et al. [3] with iStar extensions

This paper is structured as follows: Section 2 presents the background about modelling languages and their extensions, KAOS and catalogues. Section 3 presents related works. The methodology is presented in the Section 4. Section 5 presents the catalogue of KAOS extensions. The catalogue evaluation is presented in Section 6. Conclusions and future work are described in Section 7.

2 Background

This section presents the essential concepts that support this work.

2.1 KAOS and its Extensions

KAOS is a modeling language that uses a combination of four models: objective model, responsibility model, object model, and operation model. These models will be detailed below based on [5].

- **Objective Model:** This model represents the system requirements as goals and objectives and thus focuses on achieving those goals. Goals are typically all functional and non-functional requirements that must be incorporated into the system being developed, often through the assistance of some agents.
- **Responsibility Model:** This model involves entities called agents that represent humans or automated components and that are concerned with achieving objectives. The responsibility diagram describes the requirements and expectations assigned for each agent.
- **Object Model:** This model focuses on modeling the objects, entities, agents, and associations between them. Entities describe and translate the state of the object but do not perform operations. Agents are responsible for executing operations. Associations are entities that depend on the object and cannot perform operations.
- **Operation Model:** The operation model represents all the behaviors that agents must have to reach their needs. Behaviors are operations performed by agents.

KAOS has been extended to the modeling of several application areas and domains, such as aspects, adaptive systems, and security [12] [15].

Figure 1 shows an example of using a KAOS extension for modeling custom adaptive systems [11]. We can identify elements that are not part of the standard KAOS syntax, such as *domain assumption* and *human actor*.

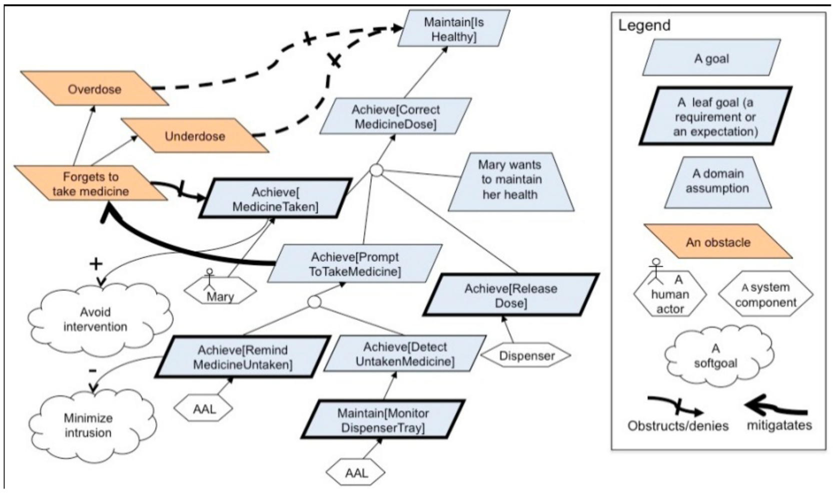


Fig. 1. Example of a KAOS Extension to Model Adaptive Systems [11]

2.2 Catalogues in Software Engineering Area

Catalogues are a usual solution to help software engineers to reach quality characteristics [6]. According to [7], a catalogue is a set of joined knowledge about previous experience.

Software requirements catalogue is defined by [8] as a set of requirements patterns that are related in a way that suggests additional requirements. They are generally proposed to allow reuse.

Requirements reuse consists of using requirements developed for a given system to model a different system. It is possible to save time and effort by reusing requirements since the reused requirements have already been analyzed in other systems [9]. Furthermore, the use of catalogues prevents engineers from spending time researching diverse sources or relying on experts in the field to make decisions on how to obtain [6] requirements.

3 Related Work

We did not find any catalogue of KAOS extensions. However, catalogues have been proposed to gather knowledge in requirements engineering, as presented below.

The work of [2] consists of identifying and analyzing the existing extensions of the iStar modeling language based on an SLR that resulted in 96 papers and 307 constructs identified. This research is related to the proposed work because the authors of [2] present a list of extensions of a modeling language as a result of an SLR.

A catalogue of iStar extensions is presented at [10]. The extensions and their constructs were identified from an SLR, which identified 96 proposed extensions up to 2016. This work is related to the proposed work because it results in an online tool (catalogue) with the list of works that define new language extensions of iStar modeling obtained from an SLR.

The work of [12] aimed to identify and analyze KAOS extensions. The SLR search identified 955 articles in 7 electronic databases. This SLR considered articles up to the year 2019. This SLR was recently updated in [15] so that 42 KAOS extensions were identified in total. This work is related to this SLR because, as a result of the SLR, a list of works that define new extensions of the KAOS modeling language is obtained.

4 Methodology

This section describes the steps followed to propose the catalogue of KAOS extensions. Figure 2 illustrates this sequence.

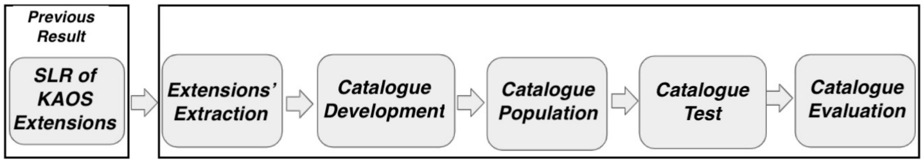


Fig. 2. The method used to create the catalogue of KAOS extensions

The results of the SLR of KAOS extensions presented in [12] [15] are used as a starting point for the steps performed in this work. Thus, it was not represented in Figure 2 as a step in the methodology but as a previous result.

Each step of the methodology is described below.

Extensions' Extraction: To extract extension and construct information, templates were built that served as a basis for extracting extension data, creating and populating the catalogue. The templates are presented in Section 5.1.

Catalogue development: The KAOS extensions catalogue was developed as a website. It is based on the extraction form fields described in the previous step. An incremental approach was followed, starting with the features related to the extensions, and then the features related to the constructs were implemented.

Catalogue Population: From the spreadsheets[1] obtained in the Information Extraction step and with the *deploy* of the catalogue made at *Firebase*

[1] Link to the spreadsheets: http://bit.do/fTpM8

Hosting [2], the extensions and their constructs were registered in the online catalogue.

Catalogue Test: Functional testing is a quality assurance process. Functions are tested by feeding them input and examining the output.[13]. catalogue functionalities were tested based on a test sheet. The tests were carried out by an undergraduate student, a master's student in computer science, and a professor. The identified *bugs* have been fixed, and the identified improvements have been implemented.

Catalogue Evaluation: An empirical study was performed to evaluate the catalogue of KAOS extensions. We analyzed the correctness of the results when using the catalogue and the effort involved.

5 Catalogue of KAOS Extensions

The catalogue of KAOS extensions will be presented in this section. It is structured as follows: Initially, we will describe the extraction of extensions through templates, the technologies involved in the development, and then an overview.

5.1 Extensions'Extraction

Two templates were created to extract information about extensions and constructs from the SLR of KAOS extensions. The list of extension's template fields is shown in Table 1.

Table 1. Fields of the extension's template.

ID - Code to identify the extension
Title - Title of the paper related to the KAOS extension
Link - Link to access the paper of the KAOS extension
Authors - Authors of the paper of the KAOS extension
Field - Journal/Conference/Book
Extension base - If the extension is based on other extension
Application Area - It can be one or more of the following values: Aspects, Adaptive Systems, Autonomic Systems, Business process, Enterprise, Web services, Risks, Safety, Organizational , Security-privacy-vulnerability and Other
Level of extension - The level of the representation involved in the proposal of the extension (Abstract syntax, Concrete Syntax or Both)
Compatibility between metamodel and concrete syntax - To identify if there is compatibility between the representation in the metamodel and concrete syntax

The list of constructs' template fields is shown in Table 2.

The information was manually extracted from the extensions papers. The catalogue was developed based on this structure, and the extracted data were used to populate it.

[2] https://firebase.google.com

Table 2. Fields of the constructs' template.

ID - Code to identify the construct
Name - Name of the concepts which the construct represents
Description - meaning of the concept the construct represents
Type - Node or Link
Example - A model with the usage of the construct

5.2 Presenting the KAOS Extensions Catalogue

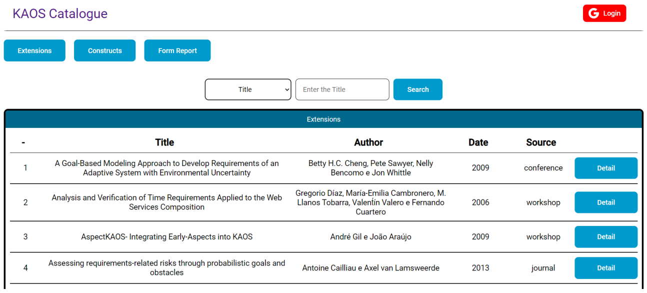
The catalogue was developed based on the following technologies *Firebase* (Development Platform), *Firebase Authentication* (Authentication provider), *Firebase Hosting* (deployment), *Firebase Realtime Database* (Database), *Firebase Cloud Storage* (File storage), *React* (Library for creating interfaces and *Typescript* (Programming language).

The KAOS extensions catalogue is available at https://kaos-catalogue-d9e49.web.app.

It is possible to see the list of all cataloged extensions and select one to detail showing all the above information and new information

The tool allows you to search for extensions by author or title and filter by application area, year of publication, place of publication, or if it has a support tool.

Figure 3 shows the list of registered extensions with previous information and a link to access detailed information for each extension. On this page, you can search for extensions by author or title or filter by application area, year of publication, place of publication, or if it has a support tool.



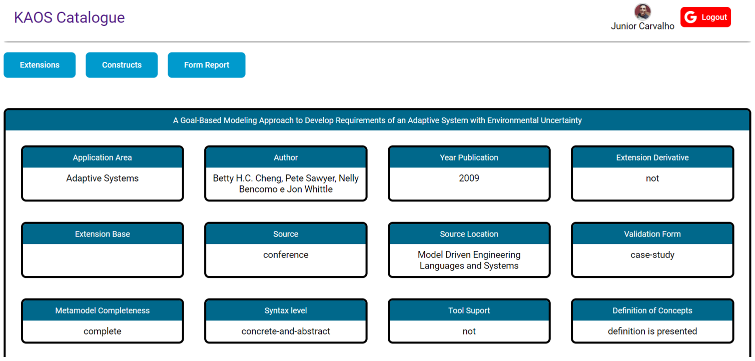
Extensions

-	Title	Author	Date	Source	
1	A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty	Betty H.C. Cheng, Pete Sawyer, Nelly Bencomo e Jon Whittle	2009	conference	Detail
2	Analysis and Verification of Time Requirements Applied to the Web Services Composition	Gregorio Díaz, María-Emilia Cambronero, M. Llanos Tobarra, Valentín Valero e Fernando Cuartero	2006	workshop	Detail
3	AspectKAOS- Integrating Early-Aspects into KAOS	André Gil e João Araújo	2009	workshop	Detail
4	Assessing requirements-related risks through probabilistic goals and obstacles	Antoine Cailliau e Axel van Lamsweerde	2013	journal	Detail

Fig. 3. Extensions list

The catalogue stores the following information about extensions: title, application area, authors, year of publication, name of the journal/event, if it is an extension based on another existing extension, compatibility between metamodel and concrete syntax, type of validation, if it presents the definition of concepts, if there is tool support, level of extension, completeness of the metamodel, list of constructs and a link that leads to the site from which the work was published. The list of constructs related to the selected extension is also displayed—additionally, there is a button to access the details for each construct.

Figure 4 shows an extension detail page. It shows the information about the selected extension and a link to the extension paper page. In addition, the list of constructs of the selected extension is presented. In the list of constructs, it is also possible to access the details of each one.



A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty

Application Area	Author	Year Publication	Extension Derivative
Adaptive Systems	Betty H.C. Cheng, Pete Sawyer, Nelly Bencomo e Jon Whittle	2009	not

Extension Base	Source	Source Location	Validation Form
	conference	Model Driven Engineering Languages and Systems	case-study

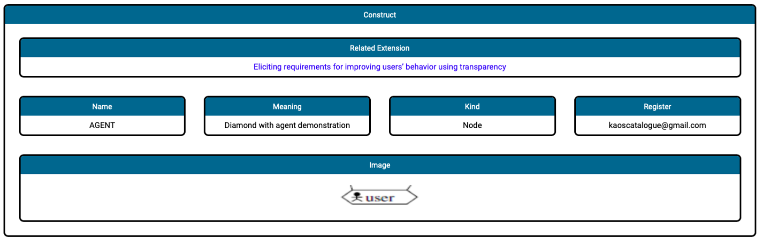
Metamodel Completeness	Syntax level	Tool Suport	Definition of Concepts
complete	concrete-and-abstract	not	definition is presented

Fig. 4. The information of a KAOS extension

The catalogue has the following information about the constructs: description, meaning, format, and visual example.

Similar to the extensions list, it is also possible to access the list of all cataloged constructs and search for constructs by name or kind (Node or Link). Information about a construct is presented when the option detail is selected. Related extension presents the link to details of the extension that proposed the construct. Additionally, the construct's name, meaning, kind, and figure are presented.

Figure 5 shows the functionality of searching constructs and Figure 6 shows the detailing of a construct.



Construct

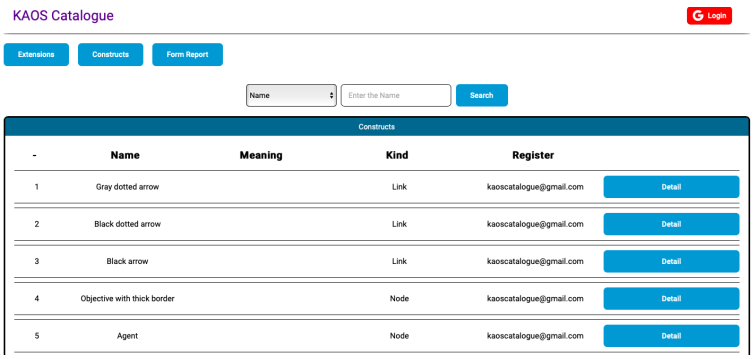
Related Extension

Eliciting requirements for improving users' behavior using transparency

Name	Meaning	Kind	Register
AGENT	Diamond with agent demonstration	Node	kaoscatalogue@gmail.com

Image

Fig. 6. Detailing information of a construct



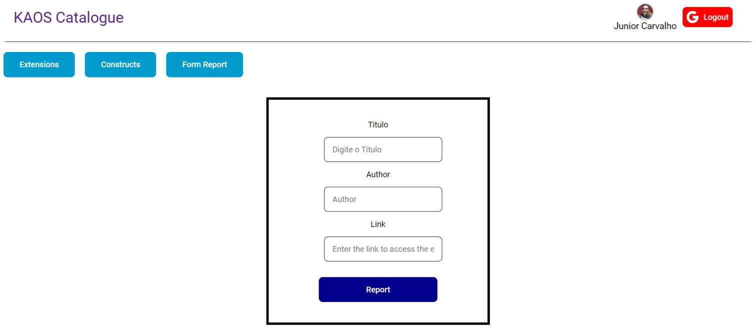
Constructs

-	Name	Meaning	Kind	Register	
1	Gray dotted arrow		Link	kaoscatalogue@gmail.com	Detail
2	Black dotted arrow		Link	kaoscatalogue@gmail.com	Detail
3	Black arrow		Link	kaoscatalogue@gmail.com	Detail
4	Objective with thick border		Node	kaoscatalogue@gmail.com	Detail
5	Agent		Node	kaoscatalogue@gmail.com	Detail

Fig. 5. List and Search Constructs

Finally, it is possible to report a new KAOS extension by informing the title, link to access it, and email. And then, Experts in KAOS extensions can endorse the reported extensions.

Figure 7 shows the form to report a new extension. After logging in, one can report the extension, and the platform administrator has the option to accept it or not.



Titulo

Digite o Titulo

Author

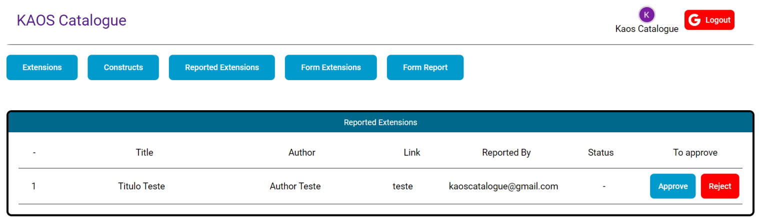
Author

Link

Enter the link to access the e

Fig. 7. Inform a KAOS extension

Figure 8 shows the list of extensions to be endorsed by the administrator. It lists all extensions that someone has reported and the link to access them. After the analysis and deciding whether or not it is a new KAOS extension, the administrator can approve or reject each one by selecting the related option.

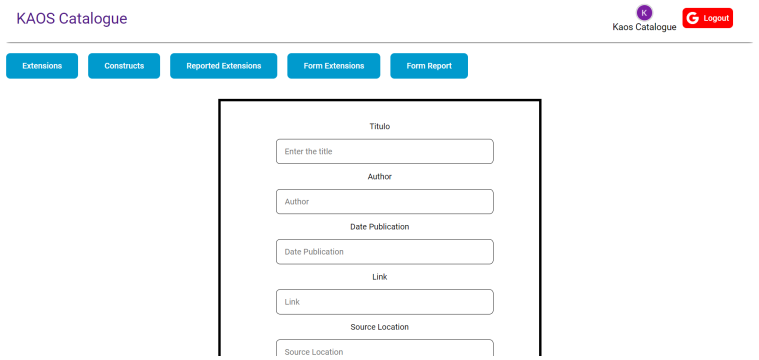


Reported Extensions

-	Title	Author	Link	Reported By	Status	To approve
1	Titulo Teste	Author Teste	teste	kaoscatalogue@gmail.com	-	Approve Reject

Fig. 8. A Reported extension

Figure 9 shows the final registration of a new KAOS extension. This form is only visible to the administrator user.



Titulo

Enter the title

Author

Author

Date Publication

Date Publication

Link

Link

Source Location

Source Location

Fig. 9. Registration of a new KAOS extension

6 Evaluation

This section presents the catalogue evaluation. This study analyzes the correctness of identifying extension information through the KAOS catalogue and the effort involved. We carried out a study with data collection based on a quantitative-qualitative questionnaire (survey) following the principles of [14]. We believe that experienced researchers can benefit from using the catalogue. However, we think inexperienced (naive) users are more prone to errors and have difficulties finding extensions and their constructs. Because of this, we defined participants with no experience with KAOS extensions as the study's target population. We chose students from a class of Software Engineering at the Federal University of Ceará in Quixadá as a sample. Twenty-two (22) students were invited, with seventeen (17) participants. We created a questionnaire using *Google Forms*. The list of questions is available at https://bit.ly/3wmJn8A.

The questionnaire is structured in four parts: Part1 - questions to characterize the participant's profile; Part2 - Questions about extensions; Part 3 - Questions about constructs; Part 4 - Questions about general opinion about the catalogue.

In part 1, we asked about the undergraduate course, knowledge about requirements engineering, and knowledge about modeling.

Parts 2 and 3 involve performing a task with the catalogue and an associated pair of quantitative questions. These questions analyze the result of the requested task (to identify the correctness) and the difficulty level of performing it.

An example of these questions: Is the paper "A Comparison of goal-oriented approaches to model software product lines Variability" a KAOS extension? Yes/No. What is the difficulty level of using the catalogue to get this information? Answers on Likert scale are in five levels.

The tasks in part 2 are related to identifying a KAOS extension, the number of KAOS extensions in a specific area, extensions of an author, extensions by year, and questions about the classification of extensions(definition of the concepts' meaning, level, venue, tool support and kind of validation/evaluation). The tasks in part 3 are related to identifying the number of constructs of a specific extension, the symbol of a construct, the kind of construct (Node or Link), the number of constructs per kind (Node or Link), and the extension proposed a specific construct.

Part 4 has questions about a general description of the catalogue, the catalogue layout, the perception of the usefulness of the catalogue to identify the correct information about extensions and constructs, and suggestions for improvements.

Before starting the evaluation, we carried out a pilot with 4 (four) participants. We applied the identified corrections, and then the pilot answers were discarded.

Due to the COVID-19 pandemic, the evaluation was carried out through a google meet in January 2022. Initially, we gave training on the KAOS extensions catalogue. Then the link to the form was made available to the participants.

6.1 Evaluation Results

We present a compilation of the main findings of this study in this section.

We describe the participants' profile (Part 1) as follows: participants were students of the Software Engineering undergraduate course in almost all terms of the course (except the second, third and fifth semester). Participants mentioned mainly a medium knowledge level about requirements. We got 1 answer in level 2, 7 answers in level 3, 5 answers in level 3, 2 answers in level 2, and 4 answers in level 5. Participants also mentioned mainly a medium knowledge level about modeling, we got 1 answer in level 2, 4 answers in level 3, 8 answers in level 3, 3 answers in level 2, and 1 answer in level 5.

Figure 10 presents data about the correctness of the main questions related to KAOS extensions.

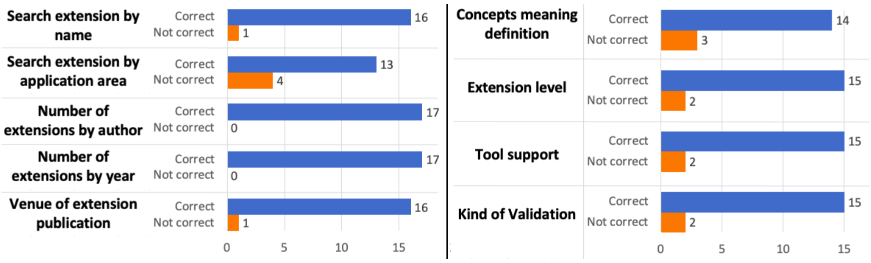


Fig. 10. Results of main questions related to KAOS extensions

Figure 11 presents data about the correctness of the main questions related to KAOS extensions' constructs.

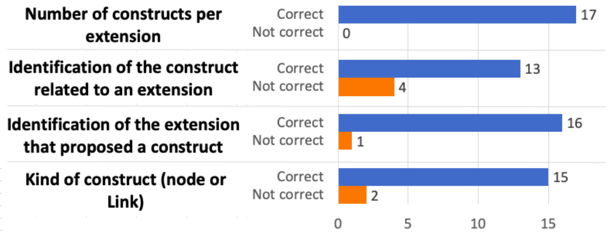


Fig. 11. Results of main questions related to KAOS extensions' constructs

The number of correct answers to the questions presented in figures 10 and 11 is greater than the not correct answers.

Analyzing the data obtained from ALL questions of parts 2 and 3, we can identify that: 47.06% (8/17) of the participants had a 100% hit rate; 23.53% (4/17) of the participants had a hit rate < 100% and >= 90%; 23.53% (4/17) of the participants had a hit rate < 90% and >= 80%. Only 1 of the participants reached a percentage of correct answers below 80%.

Four questions (4/15) had a hit rate of 100%; another four questions (4/15) had a hit rate < 100% and >= 90%; Another five questions (5/15) had a hit rate < 90% and >= 80%; Only two questions (2/15) reached a percentage of correct answers below 80%.

Among the questions with a lower percentage are questions 2.2 (How many KAOS extensions for "ADAPTIVE SYSTEMS" have been proposed?) and 3.2 (Identification of constructs that are not part of the "Adaptation Goals for Adaptive Service-oriented Architectures"). Taking into account the suggestions for improvements proposed by the participants in the questions at the end of the questionnaire, one can see that among them is the suggestion of showing the number of results obtained by applying a filter on the list of extensions which would help in the proposed task to answer question 2.2.

Regarding the participants' perception of the difficulty of performing the actions, the results show that (11/15) participants judged it as very easy or easy to obtain the answer using the catalogue in 73.33% of the questions.

In general, participants in the evaluation study described the experience of using the catalogue as positive. Among the main features pointed out by them are: simple, intuitive, fast, and efficient.

Less than 50% respondents agreed with the statement that the layout is user-friendly. Thus, usability needs to be analyzed and improved.

Regarding the complexity of performing some tasks in the catalogue, we identified at least 80% of the participants agreeing with the statement that the functionalities are intuitive.

At least 90% of the participants agreed with the statement that the catalogue was useful in quickly and correctly identifying extensions.

All questionnaire results are in a spreadsheet available at the link bit.do/fTnHA.

6.2 Threats to Validity

This section discusses threats to the validity of the survey instrument. For [14] 4 (four) aspects must be considered: Face, Content, Conclusion and Construction.

[Face validity] One can understand as inexperienced people's superficial review of the instrument items. We presented the questionnaire initially to two people who did not know the subject evaluated to review the structure, design, and objectivity of the questionnaire tasks. Even without previous knowledge of the subject, the participants achieved a high percentage of correct answers to the questions. There were some suggestions for improvement that we took. [Content validity] This is a subjective assessment of how suitable the instrument seems to a group of people with knowledge of the subject. We carried out another pilot involving two people, one being a requirements engineering specialist and the other a student of the Academic Master's Program in Computing at UFC Quixadá. The objective was to ensure that it included all the necessary demands for validation. The feedback received in this pilot showed the need to have open questions regarding the participants' opinions.

[Conclusion validity] It concerns the ability to reach the correct objective on the data collected, using statistical tests, and how reliable the measurements and these data are. The number of participants participating in this study makes it impossible to make statistical inferences about the data, which threatened the validity. [Construction validity] It is the observation of how the research instrument behaves when used. To try to mitigate threats of this type, we sought to develop objective questions, which were validated in the pilot tests that were carried out. In the results of this study, which are better detailed in the subsection 6.1, we saw that the data converged towards a high percentage of success when using the catalogue to answer the questions of our research instrument.

7 Conclusions and Future Work

Several KAOS extensions have been proposed since its creation. Consequently, finding a KAOS extension based on its characteristics or a construct for its extensions are complex tasks that can require time and effort.

In this work, we present a catalogue of KAOS extensions to ease the identification and visualization of KAOS extensions and their constructs. The catalogue is available online, and it is possible to list all extensions found in an SLR [12] [15] and access detailed information for each one. The catalogue enables to search KAOS extensions by author and title, to filter them by application area, year of publication, place of publication, or if it has a support tool.

This study is part of the results to analyze and systematize KAOS extensions. Thus, we intend to analyze the extenders' point of view about how one performs KAOS extensions and what they can do to improve the quality of the next ones in a similar way followed by Gonçalves et al. [4].

The catalogue evaluation showed that it is possible to search extensions and their constructs and obtain correct results with little effort.

In future work, we intend to analyze its usability, such as a heuristic evaluation. We also believe that it would be essential to carry out an experiment based on the principles presented by [17] to analyze the impact of using the catalogue in identifying and visualizing KAOS extensions. Finally, we intend to support the automatic extension recommendation based on the user feedback, gathering of user feedback, and user feedback classification.

References

1. Brambilla, M., Cabot, J., Wimmer, M. Model-Driven Software Engineering in Practice. Morgan Claypool Publishers. (2012)
2. Gonçalves, E., Castro, J., Araujo, J., Heineck, T. A Systematic Literature Review of iStar Extensions. The Journal of Systems and Software, v. 137, pp. 1-33 (2018)
3. Gonçalves, E. PRISE: A Process to Support iStar Extensions. Thesis on Computer Science, Centro de Informática, Universidade Federal de Pernambuco (2019)
4. Gonçalves, E., De Oliveira, M. A., Monteiro, I., Castro, J., Araujo, J. Understanding what is important in iStar extension proposals: the viewpoint of researchers. Requirements Engineering, v. 24, p. 55-84, (2019).
5. Fatima, M. Kaos: A goal oriented requirement engineering approach. International Journal for Innovative Research in Science Technology, v. 1, n. 10, p.133–135, (2015)
6. Chung, L. and Nixon, B. A and Yu, E. and Mylopoulos, J., Non-functional requirements in software engineering, Springer Science Business Media (2012)
7. Cysneiros, L. M., Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements, Workshop on Requirements Engineering (2007)
8. Silva, R. C., Uma abordagem para o reuso de requisitos baseada em padrões e rastreabilidade, Dissertação de mestrado, Univali (2011)
9. Kotonya, G. and Sommerville, I., Requirements Engineering: Processes and Techniques, Willey. (1998)
10. Gonçalves, E. and Heineck, T. and Araújo, J. and Castro, J., A Catalogue of iStar Extensions, Workshop on Requirements Engineering (2018)

11. Sutcliffe, A and Sawyer, P, Modeling Personalized Adaptive Systems, International Conference on Advanced Information Systems Engineering (2013).
12. Souza, S., Uma revisão sistemática de literatura das extensões de kaos (knowledge acquisition in automated specification, Trabalho de Conclusão de Curso em Sistemas de Informação, Universidade Federal do Ceará. (2019)
13. Costen, F., Development of Three-dimensional Model of Avocado, Doctoral dissertation, The University of Manchester (2019)
14. Kitchenham, B. and Pfleeger, S., Principles of Survey Research, Software Engineering Notes, v. 27, n. 5 (2002)
15. Monte, L. and Gonçalves, E. and Cunha, S., A systematic literature review of KAOS extensions, Under review for a copy contact us enyo@ufc.br.
16. Dardene, A and Van Lamsweerde, A. and Fickas, S., Goal-directed requirements acquisition, Science of computer programming. (1993)
17. Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., Wesslen, A., Experimentation in software engineering, Springer Science & Business Media (2012),