JCSE Journal of Computing Science and Engineering

Regular Paper
Journal of Computing Science and Engineering,
Vol. 19, No. 2, June 2025, pp. 46-60

BKAOS: An Extension of KAOS for Big Data Projects

Chabane Djeddi*
National School of Artificial Intelligence, Algiers, Algeria
chabane.djeddi@univ-constantine2.dz

Oussama Arki and Nacereddine Zarour
LIRE Laboratory, University of Constantine 2, Constantine, Algeria
oussama.arki@univ-constantine2.dz, nasro.zarour@univ-constantine2.dz

Abstract
The success of software projects is greatly impacted by requirements that are corrected on time. In the development process, fixing requirements errors early in the requirements engineering (RE) phase saves a significant amount of effort. Requirements, however, must be carefully manufactured; they are not just found. This holds true for systems in general and is particularly crucial for big data projects due to their specific characteristics, such as volume, variety, volatility, and the execution time, etc. Unfortunately, there is very little research on eliciting requirements specifically tailored to big data. In this study, we analyze existing literature to identify big data characteristics that are not adequately supported by traditional RE methods. We propose the BKAOS (big data KAOS) method, which is an improved form of the famous KAOS, to close this gap. Through 15 illustrative scenarios, we show the usefulness of BKAOS. We further verify the integrity of the method by generating a Bigraphs-based description for both BKAOS and KAOS. Our results show that BKAOS is more appropriate for eliciting requirements in big data projects than KAOS, which results in more precise requirement elicitation, lower effort levels overall, and easier data analysis.

Category: Embedded Systems and Real-time Systems

Keywords: Requirements engineering; Big data; KAOS; KAOS extension; Formal checking; Bigraphs

I. INTRODUCTION

The scientific community and businesses are realizing big data's enormous potential more and more [1, 2]. Major digital companies like Google, Baidu, and Facebook handle massive data volumes ranging from hundreds of petabytes (PB) to tens of PB driving the demand for effective big data initiatives. Government agencies have declared considerable ambitions to improve big data research and application [3]. To fully utilize data analytics and insights, these projects need to address specific needs.

Because big data projects deal with vast amounts of data, they need explicit requirements [4]. Thorough requirements engineering (RE) is crucial, and previous research has shown the importance of taking large data characteristics into account when doing so [4-11]. Precise information capture in requirements is essential to software development success [11]. However, the specific characteristics of big data make its processing difficult [4, 8-10], which makes their explicit representation in requirement specifications necessary [5, 6, 9]. Specific characteristics such as volume, variety, volatility, and the

Open Access http://dx.doi.org/10.5626/JCSE.2025.19.2.46 http://jcse.kiise.org

Received 30 August 2024; **Accepted** 22 September 2025
*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

execution time, etc., must be modelled to create effective solution designs [4, 6, 7, 11]. Research in this area remains new and scarce [4], with some exceptions like the one proposing goal-oriented methods and extension of iStar for big data projects [12].

In our pursuit of representing and modelling requirements for big data, we extend the famous KAOS [13]. KAOS serves as a goal-oriented requirements engineering (GORE) approach commonly employed for requirements elicitation. However, since KAOS is classified as a general purpose modelling language (GPML), we recognize the need for a more specialized domain-specific modelling language (DSML) in our context [14, 13]. This extension aligns with the broader trend of adapting existing RE methods to specific domains [4, 15-41].

Our research addresses a number of significant RE and big data issues, including:

1) We investigate the degree to which existing RE methods facilitate the elicitation of needs for massive data projects.
2) We look at the unique characteristics of big data that should be considered while creating requirements.
3) We propose an extension to an existing RE method, which specifically addresses the characteristics of massive data.
4) We verify that the new extension maintains the integrity features of the original method. By offering a specific, tailored method that facilitates the elicitation of precise requirements, our work enhances both RE and the big data industry.
5) We perform a quantitative empirical validation to show empirically the difference between KAOS and BKAOS (big data KAOS).

This paper is an extension of the publication [17] which addresses the limitations of traditional RE elicitation methods in the context of big data projects and highlights their unique characteristics. The contributions of that paper are here extended by including:

- Synthesis of existing research on the convergence of RE and big data (Section II);
- Application of BKAOS on more than 15 case studies to show its utility and scalability (Section III-B);
- Deep investigation on the importance of taking into account the characteristics of big data while handling requirements (Section IV-A);
- Analysis of the big data characteristics and its applicability reasons on BKAOS (Section IV-B);
- Formal semantics for the KAOS description using Bigraphs (Section V-A);
- Application of Bigraphs on BKAOS for integrity validation (Section V-B);
- Application of an empirical validation (Section VI).

Below is the format of this paper: the phases and different approaches of RE and big data are examined in Section II's literature studies. Then, we gather and analyZe the existing works on the convergence of RE and big data. After we introduce the KAOS method in Section III, we apply it to 15 use cases and demonstrate its use in a healthcare-related example. BKAOS, a KAOS extension designed specifically for big data projects is introduced in Section IV, the purpose of this extension is explained, and the extra ideas required to manage massive data requirements are outlined. In addition, we showcase the advantages of BKAOS by applying it to the same 15 illustrative scenarios. Then, we implement formal semantics for KAOS descriptions and apply formal checks using Bigraphs to confirm BKAOS's integrity in Section V. Lastly, in Section VI, we perform an empirically quantitative validation for BKAOS.

II. LITERATURE REVIEWS

This section briefly describes the fields of RE and big data. We examine present research on the convergence of RE and big data, emphasizing the importance of specific methods for addressing the specific challenges of large-scale data processing.

A. Requirements Engineering

The extent to which the stakeholders' goals are met is one of the most crucial criteria for any software's success [42]. The goal of RE is to precisely specify software behavior and track its progress over time, It also helps to identify system constraints and gather stakeholder requirements, which must be verified in order for the requirements to be agreed upon [43]. We take four actions to fulfil this [44].

1) The goal of requirements elicitation is to identify the needs. It is typically broken down into five smaller parts [45]: recognizing the application area, finding the needs' sources, evaluating the stakeholders, deciding the methods, strategies, and instruments to employ, and removing the demands made by other sources and parties.
2) Requirements analysis and negotiation: this section focuses on examining and confirming elicited requirements for accuracy, consistency, clarity, and completeness.
3) Recording requirements can help manage revisions and analyze future items and procedures, including system design, test cases, and validation.
4) Requirements validation ensures that requirements are complete, well-written, and meet customer expectations to maintain quality. This stage may be performed multiple times during the requirements development process to address defects, gaps, supplementary information, and other issues. In light of this, the software product is validated based on its requirements during the software life cycle test phase. The requirements elicitation stage is the first step that we are interested in in this work.

B. Big Data

Big data has emerged in academic and professional literature in an exuberant, disorganized, and unstructured way. Despite the appearance of multiple definitions, there is no precise definition for big data. Large datasets that are inaccessible to conventional methods are referred to as "big data" [3]. From an infrastructure standpoint, big data is defined as a substantial amount of data that satisfies the following criteria: volume, velocity, variety, veracity, and value. From an analysis perspective, big data is viewed as events. From a business perspective, big data can be viewed as an output that can be utilized immediately to improve the work [10, 11]. Not how to store data, but how to quickly analyze diverse data is the most important issue [46].

The following are the key characteristics of big data:

1) Today's manipulated data is not limited to a single representation; in addition to structured data, we also have semi-structured and even non-structured data, like social network content and web pages, which makes it challenging to manipulate this data with traditional systems [3, 47].
2) The name "big data" refers to volume, which indicates that volume had a significant role in the concept's development because most large firms manage zettabytes of data nowadays, which is obviously one of the drawbacks of old systems [3, 47, 48].
3) The traditional systems find it challenging to handle the scenario due to the crucial velocity of incoming data from multiple sources [1, 47].
4) The significance of the recorded data and its value. Decision makers may receive inaccurate results if a user runs certain queries against stored data or misuses already-existing data [47].
5) The complexity lies in how to guarantee the linkages and correlations between the data? Since these are crucial for ensuring the integrity of the data and preventing it from being found in unmanageable scenarios because they are collected from multiple heterogeneous sources in big data [47, 49].

C. Existing Works on the Convergence of RE and Big Data

Despite conducting a thorough search via electronic databases like ACM Digital Library, Science Direct, IEEE Xplore, and others that index a sizable number of articles, journals, and workshop proceedings, no research has been done specifically on the topic of extending KAOS for big data applications. We decided to provide a comprehensive assessment of the current literature on the convergence of big data and RE. By referencing the previous research, we want to shed light on the specific domains in which the RE can benefit big data in this part. We rely on the systematic literature review (SLR) conducted by [6] and [7].

1) Research Works that Posed the RE Challenges in the Context of Big Data Applications

The challenges of RE in the context of big data projects [7] are (1) the big data characteristics; they must definitely be addressed in the RE elicitation process, and it's critical to specify the characteristics in addition to the system quality criteria [4, 8, 9], (2) writing verifiable requirements, defining a testable and verifiable requirement is very important since we must achieve an agreement on the project characteristics [10].

2) Research Works that Proposed Solutions in the Convergence of Big Data and RE

A software verification tool named DICE Verification Tool (D-VerT) is presented in [50]. Its purpose is to enable designers to assess the design system against safety properties, such as reachability of undesirable configurations of the system.

Additionally, to enhance performance in big data initiatives, the work suggested in [51] is employed to gather data in relation to a clearly defined target. The scenario-based method is used to gather data for more informed decision-making. This work provides two main benefits: (1) accurate decision-making based on relevant data, (2) reduced storage space and analysis time. The work that was suggested in [51] in response to [47]'s analysis challenge. However, there is still room for improvement. Two such areas are the weighting of the chosen data and the verification process, which is necessary to confirm that the data collected match our particular needs. We presented an adaptation of KAOS for big data projects in [12]. An RE artifact model is put forth by the authors of [52] in relation to big data software development initiatives. The model illustrates the interactions and RE artifacts that are involved in creating big data software applications. In order to aid software engineers in visualizing privacy needs and designing privacy into big data systems, [53] presents a privacy extension to UML use case diagrams.

The authors of [54] suggest a conceptual descriptive architecture to aid in comprehending the system characteristics and user requirements of big data analytics software.

A goal-oriented approach is used in the work described in [55] to create value. The goal oriented modelling approach (GOMA) is a technique that involves identifying goals and providing guidance for decision-making. It presents hypotheses and uses analysis to validate them in order to determine whether or not they are true.

The authors of [56] suggest a method consisting of two procedures to address the privacy and performance needs of big data and IoT projects in a scrum manner. In the context of developing large data systems, the work suggested in [57] methodically aims to combine architecture design with data modelling techniques. [9] presents a method for analyzing and defining quality standards for big data applications.

Using data characteristics as a basis, the work presented in [4] attempted to extract general requirements for big data (e.g., volume demands greater storage capacity, velocity wants database tools with high performance, etc.).

In summary, based on an analysis of previous studies and as stated explicitly in the SLR [7], there is a dearth of research on RE methods in the context of big data projects and no KAOS extension, which is the driving force behind this study.

III. KAOS AND ITS MODELS APPLICATION

This section explains the KAOS method and its models, provides an example situation, and then applies KAOS to the example case to demonstrate its use.

A. KAOS Method

Obtaining requirements is crucial and challenging. According to surveys, software engineering research and practice are increasingly realizing how important RE is [4]. GORE is a RE method that addresses intentionality based on the relationships between various actors. Knowledge acquisition automated system, also known as Keep All Object Satisfied (KAOS) [13] has been mentioned frequently as one of the key GORE proposals [58]. This technique, known GORE, is frequently applied to requirements elicitation. The four models that make up KAOS are the Operational, Responsibility, Object, and Goal models.

The goal model is a set of objectives arranged in a top-down hierarchy. For each objective, we carry out a top-down and/or deconstruction until we identify sub-goals that may be allocated to an agent. The Object Model defines the interest objects. The Responsibility model specifies each agent's responsibilities, including expectations and tasks. The Operation model defines the required actions of agents based on their duties and operations.

Numerous KAOS extensions have been proposed across various domains, but not for big data, such as [4, 15-41]. This fact shows also that making an extension for a specific domain is so important.

B. The Illustrative Scenario

In order to show how KAOS and BKAOS are applied, we have taken 15 case studies, as it will be shown in the Validation section. One of them is healthcare.

Big data analytics revolutionizes healthcare by enabling data-driven decisions, enhancing patient care, reducing costs, and accelerating research. Through vast data from wearables, electronic health records (EHRs), and imaging, it uncovers patterns and supports personalized treatments. As healthcare faces growing complexity, leveraging big data is vital for innovation, efficiency, and global public health improvement [59].

The data is methodically gathered from a variety of sources, including hospitals, clinics, and wearable devices that constantly monitor patient health measurements. By combining these diverse data streams, the study hopes to create powerful machine learning models capable of accurately forecasting patient health decline. These prediction models are critical in allowing healthcare practitioners to take preventive actions, resulting in considerable reductions in hospital readmission rates.

This is a typical big data project because it requires processing and handling large amounts of data (formatted, semi-structured, and non-structured). The project's requirement to manage such large and diverse data sets in such short timeframes emphasizes the importance of modern data processing tools. Traditional data processing methods are unsuitable for the scope and complexity of this project, emphasizing the need for novel solutions in the field of big data analysis.

This healthcare project marks a significant milestone in the use of big data analytics to improve healthcare delivery. The project intends to transform patient care and resource management by combining and analyzing large amounts of healthcare data from diverse sources, generating predictive models, and presenting outcomes via an easy dashboard. The project's success will not only improve patient outcomes but will also highlight the importance of big data in revolutionizing healthcare practices and decision-making processes [59].

C. The Application of KAOS on the Illustrative Scenario

The following activities help to develop the goal-oriented process of KAOS: goal identification, goal formalization, object modelling and state variable identification, goal

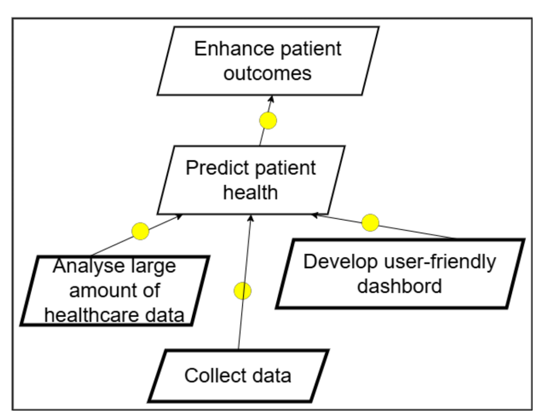


Fig. 1. An extraction from the application of KAOS's goal model on the healthcare case study.

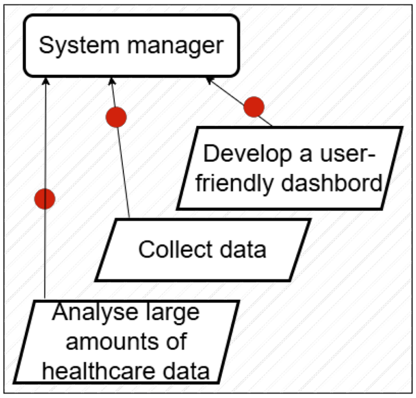


Fig. 2. An extraction from the application of KAOS's responsibility model on the healthcare case study.

conflict level detection and resolution, goal refining and agent responsibility identification, creation of obstacles and goal fulfilment resolution, and derivation of operation requirements from system goals [58].

Fig. 1 shows a small extraction of the application of KAOS's goal model on the healthcare example. Fig. 2 shows a small extraction of the application of KAOS's responsibility model on the healthcare example.

IV. BKAOS: AN EXTENSION OF KAOS FOR BIG DATA PROJECTS

We introduce BKAOS (big data KAOS) in this section, which is a KAOS method extension tailored for large-scale data projects. We first describe why this KAOS modification is necessary to facilitate the elicitation of requirements for big data projects. Next, we go over the concepts that need to be added. Finally, we run the BKAOS on the hypothetical healthcare institution.

A. The Needs for an Extension of KAOS

Big data applications using RE are still in their infancy. More clarification is required [8]. Inaccuracies and shortcomings can have a catastrophic impact on the quality of the final software product as well as the ensuing development stages. As a result, RE must be carried out carefully and precisely. The most important phase in RE is elicitation [13]; a poorly executed elicitation process can result in projects that are not adequately responsive to the demands of stakeholders. Projects involving large amounts of data are becoming increasingly complex. In addition to satisfying a demand, a big data project has to process a large quantity of data of a certain kind quickly (structured,

Table 1. Some of the citations on the importance of taking into account the features of big data while handling requirements

Citations	Reference
It's critical to specify the requirements in depth for the software to be constructed successfully.	[11]
However, each big data application has specific needs. These specifications derived from its attributes.	[4]
We see that the V-characteristics of big data are perpendicular to these conventional quality parameters. Being able to incorporate this complementary collection of qualities into the formulation of system requirements is therefore one of the research problems.	[7]
When defining, analyzing, and specifying the requirements for big data systems, it is necessary to take the features of big data into consideration.	[6]
Big data is one of the challenges in RE.	[8]
The four Vs of big data, which stand for volume, velocity, variety, and veracity, are presenting significant obstacles.	[9]
Researchers have found that the RE process requires a thorough grasp of the properties of big data.	[7]
It is necessary to gather unique needs tailored to big data in order to develop big data applications.	[4]
Since each type of software should have unique qualities, different software types may have distinct requirements.	[4]
The difficulties this assessment identifies originate from the need to comprehend and account for big data-specific features (volume, velocity, diversity, etc.) when addressing the system requirements.	[7]
The necessity of accurately taking into account the big data V-characteristics while defining, analyzing, and defining functional and quality requirements.	[7]
However, all of the models merely define the requirements in terms of functional aspects; they do not take into account bounded or verified time.	[11]
Developing big data applications has several issues, one of which is how to handle the features of big data in terms of quality standards.	[9]
The necessity to appropriately align the features of the distributed systems methods available for use in creating big data systems with the needs of the system being developed is a fourth difficulty in designing data-intensive systems.	[5]

Table 2. Some of the citations on the importance of adding new concepts to consider the features of big data projects

Citations	Reference
The big data needs must take into account the big data characteristics.	[10]
If a model for the formulation and verification of functional requirements can be created, then the requirement problem can be comprehended. It is necessary to create a model that can incorporate characteristics like time and functional elements in order to demonstrate this big data software capability.	[11]
To fully specify the requirements specification for such software, the big data software must contain all three criteria (requirement, time, and verifiable over a period of time).	[11]
The ability to incorporate this additional collection of qualities (known as big data characteristics) into the formulation of system requirements is one of the research problems.	[6]
In order to construct a solution design that satisfies the criteria, the properties of big data must be described in requirements notations.	[7]
In order to construct a solution design that satisfies the requirements, the attributes of big data elements (such as volume, velocity, variety, etc.) must be representable in requirements notations.	[8]

semi-structured, non-structured) [46].

The big data characteristics are viewed by the authors of [4-11] as a challenge in the RE. Additionally, studies [7, 10, 11] have confirmed that the big data software must include all three parameters (functional feature, time constraint, and verifiable during some period) in order to fully define the requirements specification for big data projects. Tables 1 and 2 illustrate in detail the citations from those papers.

B. The Added Concepts of BKAOS

We constantly strive to bring the model closer to reality by adding new concepts in order to improve the accuracy of the big data project that poses its unique challenges, which explains why there is a great deal of extension [12, 17, 60-65].

For each big data characteristic, we conducted discussions with domain experts to assess the feasibility of predicting their values during the requirements elicitation phase. Based on these discussions, we concluded that it is currently feasible to predict the values of volume, variety, volatility, and execution time at this early stage. However, the prediction of the remaining characteristics is not yet achievable now.

Also, based on the needs of requirements for big data in the literature [3, 6, 7, 8, 10, 11, 46, 47]. We decide to include the ideas of objective volatility, execution time, volume and variety of data to process.

Table 3 shows the big data characteristics and their applicability reasons. We highlight that there is a significant difference between Tables 1, 2, and 3. All three tables are essential in the paper. Tables 1 and 2 compile key citations from the literature that provide strong motivation for our work, whereas Table 3 presents the selected characteristics of big data and their relevance to our approach.

It is evident that in order to assist large data projects using the KAOS method, we need to make sure that our

Table 3. Big data characteristics and their applicability

Characteristic	Applicability	Reason
Variety	Applicable	Many papers emphasize taking charge of this property
Volume	Applicable	Many papers emphasize taking charge of this property
Velocity	Not applicable	Impossible to predict now
Value	Not applicable	Impossible to predict now
Veracity	Not applicable	Impossible to predict now
Variability	Not applicable	Impossible to predict now
Validity	Not applicable	Impossible to predict now
Vulnerability	Not applicable	Impossible to predict now
Volatility	Applicable	Many papers emphasize taking charge of this property
Visualization	Not applicable	Impossible to predict now
Execution time	Applicable	Many papers emphasize taking charge of this property

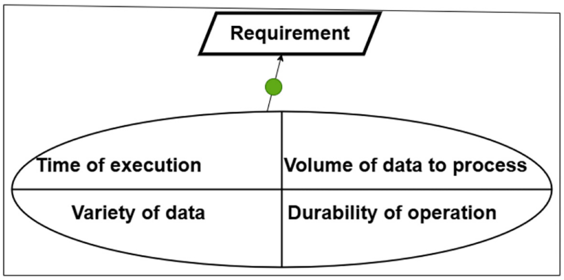


Fig. 3. The added concepts to the goal and the responsibility models of KAOS.

goals—which include execution speed, volume of data to be processed, variety of data, and goal durability are linked to theirs. Fig. 3 shows graphically the concepts added to the Goal model and the Responsibility model.

In the rest of this subsection, we explain in detail each concept and clarify why adding these concepts is necessary.

1) Volume of the Data

Big data projects commonly entail the processing of hundreds of petabytes of data. Despite significant advancements in technologies such as Hadoop and NoSQL, data volume remains a critical factor in big data initiatives. The existing KAOS methodology does not explicitly address data volume as an independent concept for requirements elicitation in big data projects. For example, the stakeholder in Section III illustrative scenario of the healthcare institution has to "analyze healthcare data" but does not indicate the volume of data that needs to be processed.

Upon project completion, the objective was to analyze 100 zettabytes of data; however, this target was not achieved, resulting in the project's failure. This outcome underscores the necessity of establishing the required data volume for each goal at the outset of the project.

2) Execution Time

Execution time is critical, especially for big data applications, because a delayed result can lead to failure. In this context, execution time should be precisely defined as the duration between the initiation and completion of a task. For example, while a stakeholder may request to analyze healthcare data," the absence of a specified timeframe can lead to project failure, as seen in our case where the analysis was completed but exceeded the 15-day target, making the outcome irrelevant to stakeholder needs. Therefore, in the KAOS method, it is essential to model execution time as a quantifiable constraint—such as "analyze 10 TB of patient data within 8 hours"—and integrate it into goal refinements and feasibility analyses. Clearly defining and incorporating such time metrics from the outset ensures that solutions remain aligned with stakeholder expectations and operational requirements, thereby preventing wasted effort and unmet goals.

3) Variety of the Data

One of the most crucial aspects of big data projects is their diversity. Different types of data exist, including non-structured (such as photos and emails) and semi-structured (such as JSON and XML) databases. Additionally, the processing of these data varies widely. The nature of the data to be processed in the elicitation step must be taken into account by KAOS. We examine the healthcare institution example given in Section III, where the stakeholder specifies that they need to "analyze healthcare data" but does not elaborate on the type of data that has to be collected. The project will be completed, but the analysis of semi-structured and non-structured data is still necessary. We get to the conclusion that each goal's type of data needs to be stated at the outset of the project.

4) Durability of the Goal

In the context of big data initiatives, engineers typically design systems to function for a set amount of time, beyond which they may no longer be usable. Therefore, the persistence of a goal needs to be specified right from the start in the RE method for big data. We use the hypothetical healthcare institution scenario from Section III as an example. The stakeholder needs to "launch targeted advertisement," but they don't say how long they think it will last. The project will be completed without a hitch, but when we present it to a stakeholder, he objects, saying that the target must be met during the whole pandemic. Therefore, the project has not met the stakeholder's needs. The durability of a goal must be specified at the beginning of the project.

KAOS does not support the characteristics presented above, which do not allow a complete and refined elicitation of the requirements for big data. BKAOS came to overcome these issues and allow a better elicitation.

C. The Application of BKAOS on the Illustrative Scenario

We keep the same meaning explained in Section III. New concepts, however, are connected to the goal of "analysis of healthcare data" in BKAOS. This means that 100 Petabytes of non-structured and semi-structured data must be analyzed in 15 days to meet this goal, and the system must be operational during the pandemic.

The goal "collect data" must be done within 5 days by analyzing 100 GB of structured data, and it must be functional during the pandemic.

Fig. 4 shows a small extraction of the application of the BKAOS goal model on our healthcare case study. In the graphical representation, NS and SS are the data sources non-structured and semi-structured.

Fig. 5 shows a small extraction of the application of the BKAOS responsibility model on the healthcare example.

The use of BKAOS allows catching all the right specific requirements for big data projects.

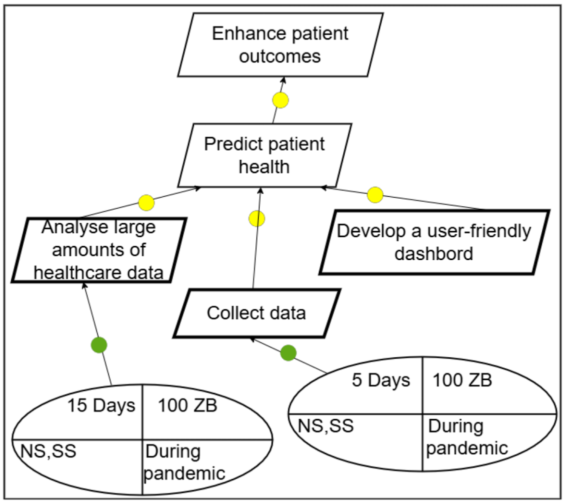


Fig. 4. An extraction from the application of the BKAOS goal model on the healthcare case study.

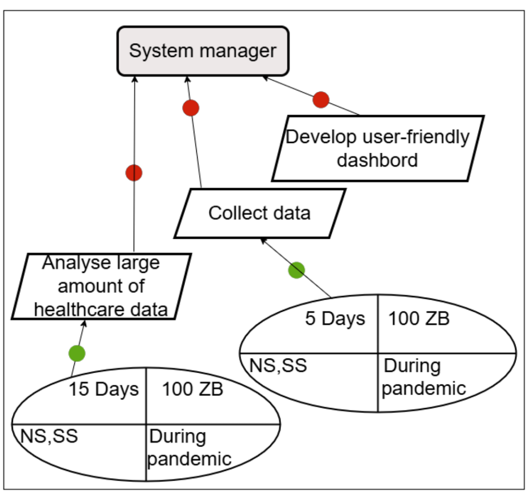


Fig. 5. An extraction from the application of the BKAOS responsibility model on the healthcare case study.

V. THE FORMAL CHECKING OF BKAOS

In this section, we show how to ensure the integrity property of BKAOS; we define a formal semantics based on Bigraphs theory for KAOS and then deduce an extended well-formed model for BKAOS. Bigraphs [66] are an emerging graphical formalism for designing, simulating and analyzing ubiquitous computing systems. Structurally, Bigraphs are a graphical meta-model which emphasizes both locality and connectivity of mobile systems.

We support the use of the Bigraphs model to ensure the integrity property of our new extended method for the following reasons:

1) it is clear and uses a graphical representation for better comprehension;
2) it forms the mathematical foundation for the systems specified, allowing for mathematical operations to be used to enrich or extend the systems;
3) its two underlined structures (the place graph and link graph) are orthogonal and specify the places and links of system agents independently; and
4) it also allows for the specification of system behavior thanks to reaction rules. We designate Bigraphs concept equivalent for every KAOS and BKAOS concept.

KAOS and BKAOS	Graphical representation in KAOS and BKAOS	Bigraphs concepts	Graphical representation in Bigraphs
Goal	Goal	Node (Goal)	Goal
Requirement	Requirement	Node (Requirement	Requirement
Agent	Agent	Node (Agent)	Agent
Big data requirements		Node	BigdataRequirement
Link of the Bigdata Requirements		Link	EBigdataquirements
Link of goal refinement		Link (Goal refinement link)	E_{GRL}
Link of responsibility		Link (Responsibility link)	E_{RL}

Fig. 6. Mapping rules between BKAOS elements and Bigraphs concepts.

Fig. 6 shows the mapping rules between BKAOS elements and Bigraphs concepts for the rest of this section.

A. A Formal Semantics for KAOS Description

In this subsection, we give Bigraphs-based definitions for KAOS. Each element in these models has a formal semantics in terms of Bigraphs, allowing a clear definition of KAOS extensions.

Through the following formal definitions, we give more details.

Definition 1. The KAOS model semantics is defined by a Bigraph:

$$Big_{kaos} = \{N_{kaos}, E_{kaos}, Ctrl_{kaos}, G^{p}_{kaos}, G^{l}_{kaos}, I_{kaos} \rightarrow J_{kaos}$$

where:

- N_{kaos} is the set of nodes, and constitute three types: Goal, Requirement, Agent.
- E_{kaos} is the set of edges, can be one of two types: $\{E_{GRL}, E_{RL}\}$.
- $Ctrl_{kaos}$ (Goal): {Atomic; 3} (the ports).
- $Ctrl_{kaos}$ (Requirement): {Atomic; 2}.
- $Ctrl_{kaos}$ (Agent): {Atomic; 1}.

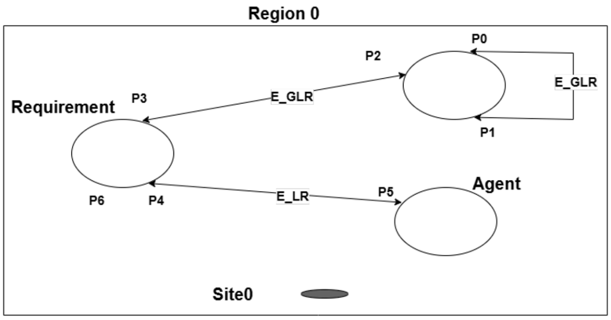


Fig. 7. A Bigraph-based definition for KAOS models.

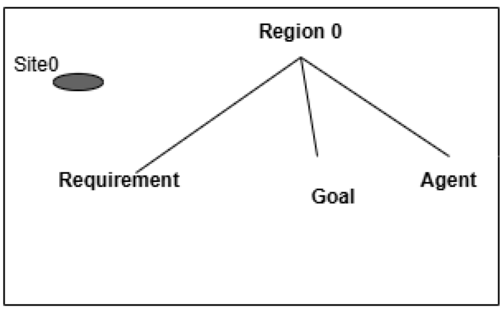


Fig. 8. Place graph for KAOS models.

- G^p_{kaos} is the place graph representing particularly the parent function defined as:
  - $prnt = \uplus V_{kaos} \rightarrow V_{kaos} \uplus \{Region0\}$, Knowing that, prnt(Goal) = prnt(Requirement) = prnt(Agent) = Region0.
- G^l_{kaos} is the graph of links representing particularly the link function defined as:
  - $link : \emptyset \uplus P \rightarrow E_{kaos} \emptyset \uplus$, P is the set of ports P_1, P_2, *etc.*
  - $link(P_2) = E_{GRL}$, $link(P_4) = E_{RL}$
- $I_{kaos} = <1, \emptyset>$, without inner names and having one site abstracting the possible insertion of other nodes.
- $J_{kaos} = <1, \emptyset>$, without outer names and having one region.

Fig. 7 shows graphically the application of Bigraphs on KAOS, we have one region, a site, the nodes (Goal, Requirement, Agent), their ports, and the relations between them. Fig. 8 shows the application of place graph on the KAOS models.

B. A Formal Semantics for BKAOS Description

In this part, we demonstrate how, by enhancing their element sets by a single node type, the Bigraphs defining the semantics in the context of BKAOS are inferred from those of KAOS:`BigdataRequirements` and one link type:`EBigdataRequirements`. Below are their respective formal definitions:

Definition 3. The BKAOS model semantics is defined by a Bigraph

$$Big_{bkaos} = \{N_{bkaos}, E_{bkaos}, Ctrl_{bkaos}, G^p_{bkaos}, G^l_{bkaos}\}: I_{bkaos} \rightarrow J_{bkaos}$$

where:

- $N_{bkaos} = N_{bkaos} \cup \{BigdataRequirements\}$
- $E_{bkaos} = E_{bkaos} \cup \{E_{BigdataRequirements}\}$
- Ctrl(BigdataRequirements) = {Atomic;1}
- G^p_{bkaos} is the place graph representing particularly the parent function defined as:
  - $prnt\{Site0\} = \uplus V_{bkaos} \rightarrow V_{bkaos} \uplus \{Region0\}$, Knowing that, prnt(Goal) = prnt(Requirement) = prnt(Agent) = prnt(BigdataRequirements) = Region0.
- G^l_{bkaos} is the graph of links representing particularly the link function defined as:
  - $link : \emptyset \uplus P \rightarrow E_{kaos} \emptyset \uplus$, P is the set of ports P_1, P_2, *etc.*
  - $link(P_2) = E_{GRL}$, $link(P_4) = E_{RL}$, $link(P_{01}) = E_{BigdataRequirements}$
- $I_{bkaos} = <1, \emptyset>$, without inner names and having one site abstracting the possible insertion of other nodes.
- $J_{bkaos} = <1, \emptyset>$, without outer names and having one region.

Fig. 9 shows graphically the Bigraphs associated with BKAOS. We have one region, a site, the nodes (Goal, Requirement, Agent, BigdataRequirements), their port, and the relations between them.

Fig. 10 shows the application of place graph on the BKAOS models.

We can conclude from the formal definition of KAOS and the formal definition of BKAOS, that:

$N_{BKASO} = N_{KASO} \cup BigdataRequirements$.

And $E_{BKASO} = E_{KASO} \cup \{E_{BigdataRequirements}\}$.

These formal definitions prove the integrity property of BKAOS.

The integrity property of BKAOS was ensured through formal proofs performed by applying Bigraphs-based description on both KAOS and BKAOS.

First, we did the mapping between KAOS-BKAOS and Bigraphs concepts. Second, we used Bigraphs formalism

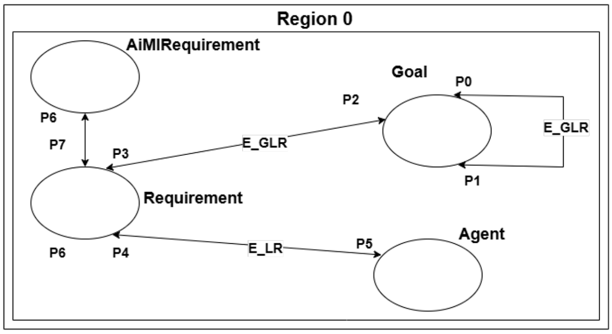


Fig. 9. A Bigraph-based definition of BKAOS models.

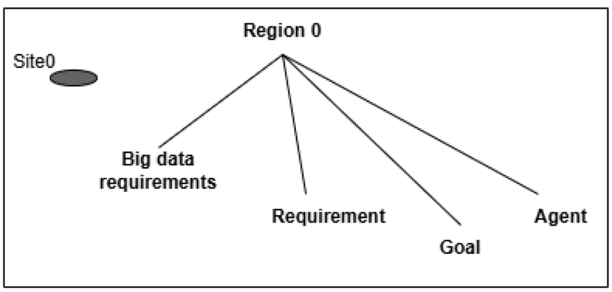


Fig. 10. Place graph for the BKAOS models.

to model and verify the structural and behavioral aspects of the KAOS method, thereby establishing its integrity property. Third, we applied Bigraphs formalism to model and verify the structural and behavioral aspects of the BKAOS method, which capture both the spatial and dynamic relationships of the system components. Fourth, using the obtained Bigraphs of KAOS and BKAOS, we concluded that the new extension (BKAOS) maintains the intended integrity constraints inherited and extended from KAOS.

This method ensured that the integrity property was not just informally argued but formally verified through mathematical reasoning and Bigraphs-based modelling, providing a high level of confidence in the correctness and soundness of BKAOS as an extension of KAOS.

VI. VALIDATION

To complete the formal verification and qualitative scenario analysis—provided in the previous sections, we conducted a quantitative empirical validation study to assess the practical benefits of the BKAOS extension in supporting big data projects, as recommended by many researchers like [67-69]. The primary objectives were to evaluate whether BKAOS improves the number of requirements captured, coverage of big data properties, modelling time and stakeholder satisfaction of requirements models compared to the baseline KAOS method in the context of big data projects. These metrics align with established requirements validation criteria widely discussed in empirical software engineering research.

Participants received training on KAOS and BKAOS before independently modelling the case studies. Fifteen ongoing ENSIA-partner projects were selected. Each was modelled, using KAOS and using BKAOS. The case

Table 4. KAOS metrics for 15 case studies

Case study domain	Requirements	Big data coverage	Modeling time	Stakeholder
Healthcare analytics	45	0	38	3.2
E-commerce recommendations	46	0	35	3
IoT sensor networks	38	0	40	2.9
Social media sentiment analysis	42	0	33	3.3
Financial fraud detection	44	0	36	3.1
Smart city traffic management	43	0	37	3.2
Genomic data processing	39	0	39	3
Supply chain optimization	41	0	35	3.1
Energy grid monitoring	42	0	36	3.2
Telecommunication network analysis	40	0	38	3
Climate modeling	38	0	40	2.9
Retail inventory forecasting	44	0	34	3.3
Cybersecurity threat detection	43	0	35	3.2
Agricultural yield prediction	39	0	38	3
Video streaming QoS optimization	41	0	36	3.1

studies reflected realistic big data project challenges, focusing on data volume, execution time, variety, and durability.

The ENSIA-partner made their feedback according to the metrics, which are. First, the number of requirements captured. Second, coverage of properties. Third, modelling time. Fourth, Stakeholder Satisfaction. Surveyed using a 5-point Likert scale on perceived clarity and usefulness of the models.

Tables 4 and 5 represent, respectively, the metrics values gathered from the users using KAOS and BKAOS. Big data coverage was assessed on a scale ranging from 0 to 4, based on the extent to which users were able to capture big data–specific characteristics using the BKAOS framework. Partial capture of a characteristic was reflected through the use of floating-point values to indicate intermediate levels of coverage. In contrast, the KAOS framework received a score of zero, as it lacked the capability to address big data–specific features. Modeling time was measured by calculating the total number of hours required to complete the modeling process.

Table 6 shows the metrics and the improvements from KAOS to BKAOS using the feedback obtained from the users. We can deduce that, First, the number of requirements was improved due to the use of the new concepts. Second, coverage increased from 0 to 1.87 (out of 4), representing a significant improvement, because KAOS does not take them into consideration. Third, the modelling time was reduced because the requirements are quickly found and gathered thanks to BKAOS. Fourth, the stakeholder's satisfaction was a bit improved.

The statistical analysis demonstrates that BKAOS significantly improves the number of requirements captured, coverage of properties, modelling time and stakeholder satisfaction of requirements models compared to the baseline KAOS method in the context of projects.

These quantitative findings empirically support the hypothesis that BKAOS better captures specific requirements and reduces modelling errors compared to KAOS. The improved stakeholder satisfaction indicates enhanced clarity and practical utility of BKAOS models. This empirical evidence complements the formal verification and qualitative scenario analysis, providing a comprehensive validation of BKAOS's effectiveness.

Table 5. BKAOS metrics for 15 case studies

Case study domain	Requirements	Big data coverage	Modeling time	Stakeholder
Healthcare analytics	65	3	37	3.7
E-commerce recommendations	60	1.8	34	3.4
IoT sensor networks	62	1.7	39	3
Social media sentiment analysis	64	2	33	3.5
Financial fraud detection	63	1.9	35	3.2
Smart city traffic management	66	2	35	3.3
Genomic data processing	61	0.8	38	3.1
Supply chain optimization	64	1.9	34	3.3
Energy grid monitoring	65	2	35	3.5
Telecommunication network analysis	62	3.6	29	3.1
Climate modeling	60	1.6	39	3
Retail inventory forecasting	65	2	33	3.5
Cybersecurity threat detection	64	1.9	34	3.3
Agricultural yield prediction	61	1.7	37	3.1
Video streaming QoS optimization	63	1.8	34	3.2

Table 6. Empirical validation results

Metric	KAOS	BKAOS	Improvement
Requirements captured (count)	41.3±2.6	63.1±1.8	+53%
Big data coverage (out of 4)	0	1.87±0.44	+1.87
Modeling time (hr)	36.7±2.3	35.3±2.1	-4%
Stakeholder satisfaction (1–5)	3.13±0.13	3.28±0.23	+5%

VII. CONCLUSION

The field of RE methods for aiding in the elicitation of projects has not received much attention in the existing research. Current RE methods are deficient in addressing the unique qualities of big data, particularly in terms of the volume, variety, execution time and validity inherent in such projects. Traditional methods frequently fall short by neglecting to consider these distinguishing characteristics. As a result, there is an urgent need to develop a unique method adapted to the needs of large data sets. For example, the widely used KAOS strategy fails to account for the special needs of projects, necessitating the development of a new method.

In this study, we provide BKAOS (KAOS), a novel extension of the KAOS method designed to effectively elicit requirements for projects. BKAOS is precisely designed to handle the specific aspects of, guaranteeing that the demand elicitation process is precise and thorough. To show BKAOS's usefulness and application, we conducted a detailed comparative case study on the healthcare domain that included both the conventional KAOS method and the newly proposed BKAOS. This strategy allowed us to objectively evaluate the benefits and drawbacks of each option in a genuine, real-world context.

We completed formal verifications of BKAOS's structural and functional integrity using the Bigraphs theoretical method. This rigorous verification approach has shown that BKAOS can capture the various and complicated requirements of initiatives. We also performed Bigraphs-based integrity checks on both the KAOS and BKAOS methods. These tests were critical in identifying and resolving any potential integrity issues that may develop as a result of extending and modifying existing methods.

In order to perform an empirically quantitative validation. We applied KAOS and BKAOS to 15 real-use cases. The feedback allows us to define four measurable evaluation metrics that are the number of requirements gathered, the properties covered, the modelling time, and the stakeholder satisfaction. This showed an important result of BKAOS for the projects.

Our findings demonstrate that BKAOS better meets the requirements of projects. BKAOS enhances previous methods by ensuring that the requirements for projects are correctly and reliably gathered. BKAOS significantly improves the discipline by adding the unique properties of into the RE process, paving the way for more effective and reliable RE in projects.

As perspectives, we suggest applying BKAOS to more case studies, integrating more characteristics, the variety will be treated more deeply to include the identification of data sources, and continuing to work on the other RE steps, such as analysis, documentation, and validation.

CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

ACKNOWLEDGMENTS

We express our sincere gratitude to all ENSIA partners who contributed to the successful application of the KAOS and BKAOS and provided their feedback on the various case studies explored in this work, which enabled us to perform the empirical quantitative validation.

REFERENCES

1. I. Varlamis, C. Sardianos, C. Chronis, G. Dimitrakopoulos, Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira, "Using big data and federated learning for generating energy efficiency recommendations," *International Journal of Data Science and Analytics*, vol. 16, no. 3, pp. 353-369. https://doi.org/10.1007/s41060-022-00331-2
2. S. Wang, Y. Wang, F. Sivrikaya, S. Albayrak, and V. W. Anelli, "Data science for next-generation recommender systems," *International Journal of Data Science and Analytics*, vol. 16, no. 2, pp. 135-145, 2023. https://doi.org/10.1007/s41060-023-00404-w
3. M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171-209, 2014. https://doi.org/10.1007/s11036-013-0489-0
4. H. Eridaputra, B. Hendradjaya, and W. D. Sunindyo, "Modeling the requirements for big data application using goal oriented approach," in *Proceedings of 2014 International Conference on Data and Software Engineering (ICODSE)*, Bandung, Indonesia, 2014, pp. 1-6. https://doi.org/10.1109/ICODSE.2014.7062702
5. K. M. Anderson, "Embrace the challenges: Software engineering in a big data world," in *Proceedings of 2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, Florence, Italy, 2025, pp. 19-25). https://doi.org/10.1109/BIGDSE.2015.12
6. D. Arruda, "Requirements engineering in the context of big data applications," *ACM SIGSOFT Software Engineering Notes*, vol. 43, no. 1, pp. 1-6, 2018. https://doi.org/10.1145/3178315.3178323
7. D. Arruda and N. H. Madhavji, "State of requirements engineering research in the context of big data applications," in *Requirements Engineering: Foundation for Software Quality*. Cham, Switzerland: Springer, 2018, pp. 307-323. https://doi.org/10.1007/978-3-319-77243-1_20
8. N. H. Madhavji, A. Miranskyy, and K. Kontogiannis, "Big picture of big data software engineering: with example research challenges," in *Proceedings of 2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, Florence, Italy, 2015, pp. 11-14. https://doi.org/10.1109/BIGDSE.2015.10
9. I. Noorwali, D. Arruda, and N. H. Madhavji, "Understanding quality requirements in the context of big data systems," in *Proceedings of the 2nd International Workshop on BIG Data Software Engineering*, Austin, TX, USA, 2016, pp. 76-79. https://doi.org/10.1145/2896825.2896838

10. C. E. Otero and A. Peter, "Research directions for engineering big data analytics software," *IEEE Intelligent Systems*, vol. 30, no. 1, pp. 13-19, 2015. https://doi.org/10.1109/MIS.2014.76
11. Sangeeta and K. Sharma, "Quality issues with big data analytics," in *Proceedings of 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016, pp. 3589-3591.
12. C. Djeddi, N. E. Zarour, and P. J. Charrel, "Extension of iStar for big data projects," in *Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE)*, Constantine, Algeria, 2018, pp. 9-16.
13. A. Van Lamsweerde, "Goal-oriented requirements engineering: a guided tour," in *Proceedings of the 5th IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 249-262. https://doi.org/10.1109/ISRE.2001.948567
14. M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, 2nd ed. San Rafael, CA: Morgan & Claypool Publishers, 2017.
15. B. Z. Mihret, E. Jee, Y. M. Baek, and D. H. Bae, "A collaboration policy model for system of systems," in *Proceedings of 2018 13th Annual Conference on System of Systems Engineering (SoSE)*, Paris, France, 2018, pp. 1-8. https://doi.org/10.1109/SYSOSE.2018.8428699
16. J. M. Morales, E. Navarro, P. Sanchez, and D. Alonso, "A controlled experiment to evaluate the understandability of KAOS and *i** for modeling Teleo-Reactive systems," *Journal of Systems and Software*, vol. 100, pp. 1-14, 2015. https://doi.org/10.1016/j.jss.2014.10.010
17. C. Djeddi, N. E. Zarour, and P. J. Charrel, "A requirement elicitation method for big data projects," in *International Conference on Managing Business Through Web Analytics*. Cham, Switzerland: Springer, 2022, pp. 231-242. https://doi.org/10.1007/978-3-031-06971-0_17
18. B. Bastani, "A requirements analysis framework for open systems requirements engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 2, pp. 1-19, 2007. https://doi.org/10.1145/1234741.1234753
19. S. Lee-Klenz, P. Sampaio, and T. Wood-Harper, "A requirements elicitation framework and tool for sourcing business-IT aligned e-services," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, Sierre, Switzerland, 2010, pp. 111-117. https://doi.org/10.1145/1774088.1774112
20. P. Massonet and A. Van Lamsweerde, "Analogical reuse of requirements frameworks," in *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (ISRE)*, Annapolis, MD, USA, 1997, pp. 26-37. https://doi.org/10.1109/ISRE.1997.566836
21. S. J. T. Fotso, M. Frappier, R. Laleau, and A. Mammar, "Back propagating B system updates on SysML/KAOS domain models," in *Proceedings of 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Melbourne, Australia, 2018, pp. 160-169. https://doi.org/10.1109/ICECCS2018.2018.00025
22. M. Fahmideh and G. Beydoun, "Big data analytics architecture design: an application in manufacturing systems," *Computers & Industrial Engineering*, vol. 128, pp. 948-963, 2019. https://doi.org/10.1016/j.cie.2018.08.004
23. C. Ponsard, P. Massonet, J. P. Molderez, A. Rifaut, A. V. Lamsweerde, and H. T. Van, "Early verification and validation of mission critical systems," *Formal Methods in System Design*, vol. 30, no. 3, pp. 233-247, 2007. https://doi.org/10.1007/s10703-006-0028-8
24. F. Semmak, C. Gnaho, and R. Laleau, "Extended kaos method to model variability in requirements," in *Evaluation of Novel Approaches to Software Engineering*. Heidelberg, Germany: Springer, 2008, pp. 193-205. https://doi.org/10.1007/978-3-642-14819-4_14
25. E. Dubois, E. Yu, and M. Petit, "From early to late formal requirements: a process-control case study," in *Proceedings of the 9th International Workshop on Software Specification and Design*, Ise-Shima, Japan, 1998, pp. 34-42. https://doi.org/10.1109/IWSSD.1998.667917
26. L. Baresi, L. Pasquale, and P. Spoletini, "Fuzzy goals for requirements-driven adaptation," in *Proceedings of 2010 18th IEEE International Requirements Engineering Conference*, Sydney, Australia, 2010, pp. 125-134. https://doi.org/10.1109/RE.2010.25
27. L. F. C. Rojas, S. M. Lince, and E. E. Pena, "Goal oriented requirements engineering supported by the SEMAT kernel," in *Proceedings of 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Puebla, Mexico, 2016, pp. 65-70. https://doi.org/10.1109/CONISOFT.2016.19
28. R. Hassan, S. Bohner, S. El-Kassas, and M. Eltoweissy, "Goal-oriented, B-based formal derivation of security design specifications from security requirements," in *Proceedings of 2008 3rd International Conference on Availability, Reliability and Security*, Barcelona, Spain, 2008, pp. 1443-1450. https://doi.org/10.1109/ARES.2008.77
29. F. Semmak, C. Gnaho, J. Brunet, and R. Laleau, "How to adapt the KAOS method to the requirements engineering of Cycab vehicle," in *Proceedings of the 4th International Conference on Evaluation of Novel Approaches to Software Engineering* (*ENASE)*, Milan, Italy, 2009, pp. 87-94.
30. A. Van Lamsweerde and L. Willemet, "Inferring declarative requirements specifications from operational scenarios," *IEEE Transactions on Software Engineering*, vol. 24, no. 12, pp. 1089-1114, 1998. https://doi.org/10.1109/32.738341
31. K. Cooper, S. P. Abraham, R. S. Unnithan, L. Chung, and S. Courtney, "Integrating visual goal models into the rational unified process," *Journal of Visual Languages & Computing*, vol. 17, no. 6, pp. 551-583, 2006. https://doi.org/10.1016/j.jvlc.2006.10.005
32. M. Tabatabaie, F. A. Polack, and R. F. Paige, "KAOS-B: a goal-oriented process model for EIS," in *Proceedings of the 8th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, Funchal, Madeira, Portugal, 2010, pp. 40-49. https://doi.org/10.5220/0003016000400049
33. M. Singh and S. Kim, "Reconcile security requirements for intelligent vehicles," in *Proceedings of 2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Jeju, South Korea, 2017, pp. 1646-1651. https://doi.org/10.23919/ICCAS.2017.8204251
34. M. Fahmideh and G. Beydoun, "Reusing empirical knowledge during cloud computing adoption," *Journal of Systems and Software*, vol. 138, pp. 124-157, 2018. https://doi.org/10.1016/j.jss.2017.12.011
35. C. Huang, J. Sun, X. Wang, and Y. Si, "Role engineering with SKAOS for systems employing RBAC," in *Proceedings*

of 2009 International Conference on Networking and Digital Society, Guiyang, China, 2009, pp. 56-60. https://doi.org/10.1109/ICNDS.2009.94
36. F. Semmak, R. Laleau, and C. Gnaho, "Supporting variability in goal-based requirements," in *Proceedings of 2009 Third International Conference on Research Challenges in Information Science*, Fez, Morocco, 2009, pp. 237-246. https://doi.org/10.1109/RCIS.2009.5089287
37. L. Duboc, E. Letier, and D. S. Rosenblum, "Systematic elaboration of scalability requirements through goal-obstacle analysis," *IEEE Transactions on Software Engineering*, vol. 39, no. 1, pp. 119-140, 2013. https://doi.org/10.1109/TSE.2012.12
38. A. Dias, V. Amaral, and J. Araujo, "Towards a domain specific language for a goal-oriented approach based on KAOS," in *Proceedings of 2009 3rd International Conference on Research Challenges in Information Science*, Fez, Morocco, 2009, pp. 409-420. https://doi.org/10.1109/RCIS.2009.5089305
39. P. Patricio, V. Amaral, J. Araujo, and R. Monteiro, "Towards a unified goal-oriented language," in *Proceedings of 2011 IEEE 35th Annual Computer Software and Applications Conference*, Munich, Germany, 2011, pp. 596-601. https://doi.org/10.1109/COMPSAC.2011.83
40. S. Tueno, R. Laleau, A. Mammar, and M. Frappier, "Towards using ontologies for domain modeling within the SysML/KAOS approach," in *Proceedings of 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, Lisbon, Portugal, 2017, pp. 1-5. https://doi.org/10.1109/REW.2017.22
41. W. Wirasta, H. L. Soemitro, and B. Hendradjaya, "Utilization of AHP method in elicitation process for goal oriented implementation using KAOS modelling," in *Proceedings of 2016 International Conference on Data and Software Engineering (ICoDSE)*, Denpasar, Indonesia, 2016, pp. 1-6. https://doi.org/10.1109/ICODSE.2016.7936144
42. J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Mate, E. Paja, M. Salnitri, J. Mylopoulos, and P. Giorgini, "Goal-oriented requirements engineering: a systematic literature map," in *Proceedings of 2016 IEEE 24th International Requirements Engineering Conference (RE)*, Beijing, China, 2016, pp. 106-115. https://doi.org/10.1109/RE.2016.41
43. J. M. Carrillo de Gea, C. Ebert, M. Hosni, A. Vizcaino, J. Nicolas Ros, and J. L. Fernandez-Aleman, "Requirements engineering tools: an evaluation (dataset)," 2025 [Online]. Available: https://doi.org/10.5281/ZENODO.14738683.
44. L. Ramingwong, "A review of requirements engineering processes, problems and models," *International Journal of Engineering Science and Technology*, vol. 4, no. 6, pp. 2997-3002, 2012.
45. D. Zowghi and C. Coulin, "Requirements elicitation: a survey of techniques, approaches, and tools," in *Engineering and Managing Software Requirements*. Heidelberg, Germany: Springer, 2005, pp. 19-46. https://doi.org/10.1007/3-540-28244-0_2
46. S. Madden, "From databases to big data," *IEEE Internet Computing*, vol. 16, no. 3, pp. 4-6, 2012. https://doi.org/10.1109/MIC.2012.50
47. A. Katal, M. Wazid, and R. H. Goudar, "Big data: issues, challenges, tools and good practices," in *Proceedings of 2013 6th International Conference on Contemporary Computing (IC3)*, Noida, India, 2013, pp. 404-409. https://doi.org/10.1109/IC3.2013.6612229
48. M. Harasic, F. S. Keese, D. Mattern, and A. Paschke, "Recent advances and future challenges in federated recommender systems," *International Journal of Data Science and Analytics*, vol. 17, no. 4, pp. 337-357, 2024. https://doi.org/10.1007/s41060-023-00442-4
49. O. C. Marcu and P. Bouvry, "Big data stream processing," Ph.D. dissertation, University of Luxembourg, 2024. https://hal.science/hal-04687320/
50. M. M. Bersani, F. Marconi, M. Rossi, and M. Erascu, "A tool for verification of big-data applications," in *Proceedings of the 2nd International Workshop on Quality-Aware DevOps*, Saarbrucken, Germany, 2016, pp. 44-45. https://doi.org/10.1145/2945408.2945419
51. D. Arruda and N. H. Madhavji, "Towards a requirements engineering artefact model in the context of big data software development projects: research in progress," in *Proceedings of 2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 2314-2319. https://doi.org/10.1109/BigData.2017.8258185
52. D. N. Jutla, P. Bodorik, and S. Ali, "Engineering privacy for big data apps with the unified modeling language," in *Proceedings of 2013 IEEE International Congress on Big Data*, Santa Clara, CA, USA, 2013, pp. 38-45. https://doi.org/10.1109/BigData.Congress.2013.15
53. L. Lau, F. Yang-Turner, and N. Karacapilidis, "Requirements for big data analytics supporting decision making: a sensemaking perspective," in *Mastering Data-Intensive Collaboration and Decision Making*. Cham, Switzerland: Springer, 2014, pp. 49-70. https://doi.org/10.1007/978-3-319-02612-1_3
54. S. Supakkul, L. Zhao, and L. Chung, "GOMA: supporting big data analytics with a goal-oriented approach," in *Proceedings of 2016 IEEE International Congress on Big Data (BigData Congress)*, San Francisco, CA, USA, 2016, pp. 149-156. https://doi.org/10.1109/BigDataCongress.2016.26
55. V. Sachdeva and L. Chung, "Handling non-functional requirements for big data and IOT projects in Scrum," in *Proceedings of 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, Noida, India, 2017, pp. 216-221. https://doi.org/10.1109/CONFLUENCE.2017.7943152
56. H. M. Chen, R. Kazman, S. Haziyev, and O. Hrytsay, "Big data system development: an embedded case study with a global outsourcing firm," in *Proceedings of 2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, Florence, Italy, 2015, pp. 44-50. https://doi.org/10.1109/BIGDSE.2015.15
57. A. Van Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *Proceedings of the 22nd International Conference on Software Engineering*, Limerick, Ireland, 2000, pp. 5-19. https://doi.org/10.1145/337180.337184
58. V. M. B. Werneck, A. D. P. A. Oliveira, and J. C. S. do Prado Leite, "Comparing GORE Frameworks: i-star and KAOS," in *Proceedings of the 12th Workshop on Requirements Engineering (WER)*, Valparaíso, Chile, 2009. https://werpapers.dimap.ufrn.br/proceedings/WER2009/wer200901.html
59. M. Manikandan, P. Venkatesh, T. Illakya, M. Krishnamoorthi, C.R. Senthilnathan, and K. Maran, "The significance of big data analytics in the global healthcare market," in *Proceedings of 2024 International Conference on Communication, Computing*

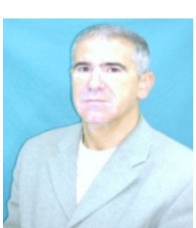
and Internet of Things (IC3IoT), Chennai, India, 2024, pp. 1-4. https://doi.org/10.1109/IC3IoT60841.2024.10550417
60. C. Djeddi, N. E. Zarour, and P. J. Charrel, “Formal verification of extension of istar to support big data projects,” *Computer Science*, vol. 22, no. 3, pp. 321-344, 2021.
61. R. Ali, F. Dalpiaz, and P. Giorgini, “Requirements-driven deployment: customizing the requirements model for the host environment,” *Software & Systems Modeling*, vol. 13, no. 1, pp. 433-456, 2014. https://doi.org/10.1007/s10270-012-0255-y
62. A. Guzman, A. Martinez, F. A. V. Agudelo, H. Estrada-Esquivel, J. P. Ortega, and J. Ortiz, “A methodology for modeling ambient intelligence applications using i* framework,” in *Proceedings of the Ninth International i* Workshop co-located with 24th International Conference on Requirements Engineering (iStar@RE)*, Beijing, China, 2016, pp. 61-66.
63. J. Lockerbie, N. A. M. Maiden, J. Engmann, D. Randall, S. Jones, and D. Bush, “Exploring the impact of software requirements on system-wide goals: a method using satisfaction arguments and i* goal modelling,” *Requirements Engineering*, vol. 17, no. 3, pp. 227-254, 2012. https://doi.org/10.1007/s00766-011-0138-8
64. J. N. Mazon, J. Pardillo, and J. Trujillo, “A model-driven goal-oriented requirement engineering approach for data warehouses,” in *Advances in Conceptual Modeling – Foundations and Applications*. Heidelberg, Germany: Springer, 2007, pp. 255-264. https://doi.org/10.1007/978-3-540-76292-8_31
65. M. Morandini, L. Penserini, A. Perini, and A. Marchetto, “Engineering requirements for adaptive systems,” *Requirements Engineering*, vol. 22, no. 1, pp. 77-103, 2017. https://doi.org/10.1007/s00766-015-0236-0
66. O. H. Jensen and R. Milner, “Bigraphs and mobile processes (revised),” Computer Laboratory, University of Cambridge, *Technical Report No. UCAM-CL-TR-580*, 2004. https://doi.org/10.48456/tr-580
67. M. V. Zelkowitz and D. Wallace, “Experimental validation in software engineering,” *Information and Software Technology*, vol. 39, no. 11, pp. 735-743, 1997. https://doi.org/10.1016/S0950-5849(97)00025-6
68. K. P. Srinivasan and T. Devi, “Software metrics validation methodologies in software engineering,” *International Journal of Software Engineering & Applications*, vol. 5, no. 6, pp. 87-102, 2014.
69. J. Cederbladh, A. Cicchetti, and R. Jongeling, “A road-map to readily available early validation and verification of system behaviour in model-based systems engineering using software engineering best practices,” *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 5, article no. 151, 2025. https://doi.org/10.1145/3708520

Chabane Djeddi https://orcid.org/0000-0002-9810-3547

Chabane Djeddi received Ph.D. degree in computer science from the University of Constantine 2, Algeria in 2023. He joined in the National School of Artificial Intelligence, in September 2023. His research interests and publications include requirements engineering for big data and AI-based systems, formal methods for requirements verification.

Oussama Arki https://orcid.org/0000-0002-5943-1919

Oussama Arki received his Ph.D. degree in Computer Science from the University of Constantine 2, Algeria, in 2020. He joined the National Polytechnic School of Algiers in January 2021. His research interests and publications include information security, deep learning, and AI-based systems



Nacer Eddine Zarour https://orcid.org/0000-0001-9152-4405

Nacer Eddine Zarour has been a professor since 2009 at the Department of Software Technologies and Information Systems at Constantine 2 University, Constantine, Algeria. He received his Ph.D. in Cooperative Information Systems from Mentouri University, Constantine, in 2004. He chaired the Scientific Council of the Faculty of New Information and Communication Technologies (NICT) and was then responsible for the mathematics-computer science field. His current research activities are carried out at the LIRE laboratory. His research areas include advanced information systems, requirements engineering, cloud computing, and data science. He has been and is responsible for several international (PHC CMEP Tassili) and national (CNEPRU/PRFU) projects.