Chabane Djeddi
Nacer Eddine Zarour
Pierre-Jean Charrel

FORMAL VERIFICATION OF EXTENSION OF ISTAR TO SUPPORT BIG DATA PROJECTS

Abstract *Identifying all of the correct requirements of any system is fundamental for its success. These requirements need to be engineered with precision in the early phases. Principally, late correction costs are estimated to be more than 200 times greater than the cost of corrections during requirements engineering (RE), especially in the big data area due to its importance and characteristics. A deep analysis of the big data literature suggests that current RE methods do not support the elicitation of big data project requirements. In this research, we present BiStar (an extension of iStar) to undertake big data characteristics such as volume, variety, etc. As a first step, some missing concepts are identified that are not supported by the current methods of RE. Next, BiStar is presented to take big data-specific characteristics into account while dealing with the requirements. To ensure the integrity property of BiStar, formal proofs are made by performing a Bigraph-based description on iStar and BiStar. Finally, iStar and BiStar are applied on the same exemplary scenario. BiStar shows promising results, so it is more efficient for eliciting big data project requirements.*

Keywords big data, requirements engineering, iStar, iStar extension, formal checking

Citation Computer Science 22(3) 2021: 323–347

1. Introduction

This paper is an extension of the ”Extension of iStar for big data projects” manuscript, which was accepted as a full paper [13] during the Third International Conference on Advanced Aspects of Software Engineering (ICAASE 2018). The upgrading from the ICAASE conference to the journal are as follows: first – synthesizing the existing works on the convergence of RE and big data; second – the proposition of a formal semantics for the iStar description using Bigraphs; and third – the application of Bigraph on BiStar to show the integrity of our work.

Companies store and process data of hundreds of PB [10]; this data has structured, semi-structured, and non-structured natures. Unfortunately, only 9% of such projects were successful in large companies [11]. One of the major reasons for the success is a clear and precise statement of the requirements [11]. Therefore, the importance of RE seems evident. The goal of the big data application is to make good decisions on time; to meet this goal, several important aspects must be highly considered, such as the completeness and consistency of the collected requirements, adequate information storage to quickly find the desired information, and finally a good analysis. So, the result showed that the last two aspects are strongly dependent on the first. The reason why the focus is given to this aspect that is linked to the requirements discovery. As the decisions taken in big data applications are also time-constrained, the requirements must contain the maximum amount of useful knowledge; therefore, there is no need to consult the stakeholders again to ask them for more information.

RE is a field that focuses on ”requirements elicitation,” ”requirements analysis and negotiation,” ”requirements documentation,” ”requirements validation,” and ”requirements management.” RE uses several methods depending on the orientation of the approach that is followed (goal, scenario, or viewpoint). Being among them, iStar [36] is one of the main methods that is used to perform the elicitation of the requirements [35]. iStar is a RE method that is classified as GPML [15] because it is general; however, many fields such as cloud computing, security, big data, etc. need specific treatments. Scientists extend iStar to meet the needs of a field when they discover that it has its own characteristics that require a specific treatment of its requirements. This explains why there are so many extensions of iStar [2,16,22,25,26] that are meant to meet the requirements of each field. As big data [10] is an emerging area, it imposes its specific characteristics: first – the volume takes an important role in the creation of the big data concept since the data handled today amounts to zettabytes at most large companies (this is, of course, one of the limitations of traditional systems); second – the variety of manipulated data today is not from a single representation – there is structured data, semi-structured data, and even unstructured data (such as web pages and social networks); third – the velocity of incoming data from various sources is so critical, which makes it difficult for traditional systems to undertake such a situation; and fourth – the complexity is how to ensure the correlation and links among the data, because the latter is collected from several heterogeneous sources in big data. These characteristics generate specific requirements that must be dealt with during the RE process; therefore, extending iStar is evident to handle the specific requirements of big data projects. After analyzing the literature, the findings showed that the authors [3, 4, 6, 14, 24, 27, 29, 32] in fact emphasized the necessity of undertaking big data characteristics while dealing with requirements; therefore, it is necessary to either create new methods or extend existing ones.

This study focuses on the need and importance of extending an existing RE method for undertaking big data properties. A method called Bistar is precisely presented; it is an extension of the known iStar method to assist and take the characteristics of big data projects into account. We add notations to iStar (the execution time, the volume of data to be processed, the variety of the data, and the durability of the goal) that allow us to ensure that each requirement must specify the volume that must be treated, the nature of the data to handle, the execution time, and the durability that this requirement is available. Finally, to ensure the integrity property of the BiStar method, a Bigraph that is based on the semantic definition for iStar is proposed in addition to deducing a Bigraph-based semantic definition for BiStar. This study aims at finding ways of improving the speed and accuracy of big data projects and, thus improving the data analysis results.

This paper is organized as follows. Section 2 contains literature reviews on RE and its steps and also presents big data, its properties, and its importance. In addition, there is a description of the literature research that applies RE to big data projects. Section 3 involves an exemplary scenario of a sales company that will accompany us throughout the application of iStar and BiStar. After iStar is presented, the Strategic Dependency (SD) Model, the Strategic Rational (SR) Model, and the application of iStar on the exemplary scenario as a GPLM are introduced. Section 4 includes BiStar (the extension of iStar) for big data projects, starting from the needs for this extension and the concepts that must be added to iStar to undertake big data projects. After this, the application of BiStar is performed on the exemplary scenario of the sales company to show the use of BiStar and its benefits. In Section 5, a formal checking is performed using Bigraphs to verify the integrity of BiStar; first, a formal semantics for an iStar description using Bigraphs is proposed, then the same is done for BiStar.

2. Background

In this section, the RE and big data domains that are related to our work are briefly described. Also, a synthesis of the existing works that deal with the convergence of RE and big data can be given by analyzing those who posed the RE challenges in the context of big data applications as well as those who proposed solutions for these challenges.

2.1. Overview on Requirements Engineering

The primary criterion for the success of any software is the degree of satisfaction of the goals that are set by the stakeholders. RE is the process of discovering these goals [28]. It is the branch of software engineering that is concerned with real-world goals that motivate the development of a software system. Concerned with precise specifications that provide the basis for analyzing requirements, RE validates what the stockholders want, defines what the designers must build, and verifies that they (the designers) have performed the specifications correctly [28, 37].

The objective of RE is to know the requirements of the stakeholders and to verify them in order to reach an agreement on the requirements. One of the difficult parts of building a software program is to decide exactly what the software should do; hence, RE helps us understand the problem. By studying the RE specifications precisely, the cost of a project can then be estimated. Moreover, RE also helps us know the limits of our system [7].

RE is usually divided into five steps [30]: first – requirements elicitation; second – requirements analysis and negotiation; third – requirements documentation; fourth – requirements validation; and finally – requirements management. Figure 1 represents the steps of RE.

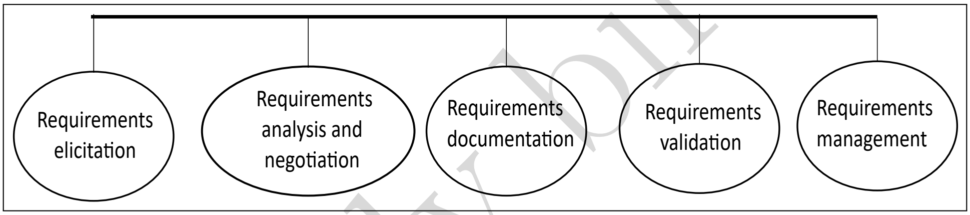

Figure 1. RE steps

The elicitation of requirements is perhaps the activity that is most often regarded as the first step in the RE process. The information gathered during requirements elicitation often must be interpreted, analyzed, modeled, and validated to ensure that the requirements that have been collected are adequately complete [28]. Requirements elicitation serves to capture the requirements – it is usually divided into five sub-steps [38]: first – understanding the application domain; second – identifying the sources of the requirements; third – analyzing the stakeholders; fourth – selecting the techniques, approaches, and tools to use; and finally – eliciting the requirements from the stakeholders and other sources. The elicited requirements are often incorrect, incomplete, inconsistent, or ambiguous. Analysis and negotiation must be done on the elicitated requirements. Manual detection of these flaws can be time-consuming, so automatic detection and correction are recommended [30]. Requirements analysis and negotiation focus on the review and understanding the elicited requirements as well as their verification for quality in terms of accuracy, completeness, clarity, and consistency. Requirements documentation aims to achieve the integrity and completeness of this documentation and has an important role to play in managing change [28]. Problems may occur while performing an agreement with all stakeholders, especially where the stakeholders have divergent goals [28]. Requirements validation is done for controlling the quality; this means confirming that the requirements are complete and well-written and supply the needs of the customer. This step may continue repeating other requirements development phases because of any identified deficiencies, the gaps between requirements, additional information, and other issues. Knowing that the implemented software product is validated in the software's life cycle test phase is based on its requirements.

In this work, the focus is on the first step, which is requirements elicitation; this is considered to be the most relevant activity in that, on the one hand, the other ones depend on it and, on the other, the sources in big data applications are numerous and the information collection requires a lot of time and effort.

2.2. Big data properties

In this section, big data will be presented by briefly passing through its definitions, properties, and importance. There is no exact definition of big data, even though several definitions have appeared. Big data means a large dataset that cannot be processed by traditional tools [10]. Big data can be seen from several perspectives; first – from the infrastructure perspective, big data is seen as a significant amount of data that is characterized by volume, velocity, variety, veracity, and value; second – from the analysis perspective, big data is seen as a dataset that is so large that it contains significant low-probability events that would be absent from traditional statistical sampling methods; and finally – from the business perspective, big data can be considered to be an output that can be used directly for the improvement of the work [29, 32]. The most crucial problem is not how to store data but rather how to analyze heterogeneous data in a short amount of time [23]. Adding to this, there is a solid relationship between big data and other technologies such as the cloud and IoT. The cloud can be an infrastructure for big data, and IoT is considered to be the most massive source of big data [10]. Consequently, our contribution to big data will influence other technologies. The variety of the data that is manipulated today is not from a single representation; there is structured data, but there is also semi-structured data and even unstructured data such as web pages and social networks that make it very difficult to manipulate this data by using traditional systems [10, 19]. The volume itself in the term "big data" means that volume plays an important role in the creation of the big data concept, since the data handled today can be in quantities of zettabytes at most large companies; this is of course one of the limitations of traditional systems [10, 19]. Velocity in big data deals with the speed of the data that comes from various sources. This is about the speed of the incoming data and also the speed at which the data flows; traditional systems cannot perform analytics on any data that is constantly in motion [19]. Hence, the complexity is how to ensure the correlation and links between the data, because the latter are collected from several heterogeneous sources in big data; it is very important to guarantee the integrity of the data and to not wind up in unmanageable situations [19]. Big data improves the productivity and competitiveness of enterprises and public sectors and creates huge benefits for consumers; thus, it create value [10].

2.3. Related work on RE and big data convergence

By searching electronic databases such as ACM Digital Library, Science Direct, Scopus, and IEEE Xplore (as they index a considerable number of papers, journals, and workshop proceedings), a critical reading should be made on the existing works of the convergence of big data and RE. This section is about clarifying the aspects in which RE can be useful to big data. The focus is on the systematic literature review (SLR) made by [6] and guided by [20]. The challenges of RE in the context of big data projects [6] are as follows: first – there is a clear need to address the big data characteristics in the elicitation step of RE, and it is important to define the characteristics along with the system quality attributes [14, 24, 27]; second – writing verifiable and testable requirements is very important since an agreement on the project properties must be achieved [29]. The work proposed in [8] presents a software-verification tool called DICE Verification Tool (D-VerT). This allows designers to evaluate a design system against safety properties such as the reachability of the undesired configurations of a system.

Furthermore, the work proposed in [1] is used to collect data that is relative to a well-defined objective – to improve the performance in big data projects. It uses a scenario-based method to collect information that will help us better select any collected data. The contribution of this work has three main advantages: first – making accurate decisions, as the collected data is exactly what is needed; second – reducing storage space, as it stores only that data that is relative to our objective; and third – reducing the time of analysis, as only relevant data should be analyzed. The work in [1] is proposed to answer the analysis challenge that was posed by the authors in [19]. However, more improvements are possible, like the verification that is needed to verify that the collected data is exactly what is being sought as well as the weighting between the selected data.

In [5], the authors proposed an RE artifact model in the context of big data software development projects. The model depicts the RE artifacts and inter-relationships that are involved in the development of big data software applications. In [18], the authors presented a privacy extension to UML use case diagrams to help software engineers visualize privacy requirements as well as design privacy into big data applications. In [21], the authors proposed a conceptual descriptive architecture to help understand the user requirements and propose the system characteristics of the Big Data Analytics software. The work proposed in [33] applies a goal-oriented method to create the value. This method called the goal-oriented modeling approach (GOMA) consists of capturing objectives and guides the decision-making. It gives propositions and validates them by analysis in order to determine whether something is confirmed or not. In [31], the authors proposed an approach that is composed of two processes for dealing with both privacy and performance requirements for IoT and big data projects in scrum.

Regarding the work that was proposed in [9], it systematically seeks combining architecture design with data-modeling approaches in the development of big data systems. The work proposed in [27] shows an approach for analyzing and specifying the quality requirements for big data applications. The work proposed in [14] tries to elicit generic requirements for big data based on the data characteristics (e.g., volume demands improved storage capacity, velocity demands database tools with high performance, etc.).

In sum, through an analysis of the existing works (and as it is clearly announced in the SLR [6]), there is a need to undertake big data characteristics by requirements engineering methods. This is our motivation in this paper.

3. iStar method and its models

In this section, the exemplary scenario is presented; the iStar method is also explained, along with it diagrams. iStar [12, 36] is a goal-oriented RE method that is commonly used for requirements elicitation. First, the start should be identifying the actors as well as the relationships of the strategic dependencies between them, after, the reasoning of each actor is well-detailed. It consists of two models: the strategic dependency (SD) model, and the strategic rationale (SR) model.

3.1. Exemplary scenario description

The exemplary scenario in this subsection is presented to be able to use it in the modeling with iStar and BiStar (iStar extended) that will be presented in the following sections. This example will accompany us throughout the paper; the advantages of using BiStar in the context of big data projects will clearly be seen.

A sales company example is chosen. Companies try always to maximize their sales in order to accomplish that, so they can create a project to study the behavior of their customers; this will allow them to know the keys on which they can focus to establish targeted advertisements in order to improve the sales of the companies. To do this, they collect data from social networks and analyze it to know the essential points in the opinion of the different categories of customers. On these points, they make a plan and present it to the customers. After this, they collect any feedback to apply changes to the plan and create targeted advertising.

This example is a big data project because it is related to manipulating a large amount of data with different natures (structured, semi-structured, and even unstructured) within a limited amount of time. Therefore, this data cannot be processed using traditional systems (as shown in the introduction section).

3.2. Strategic dependency (SD) model

The strategic dependency model represents a network of strategic dependencies between the different actors of a future system. One actor (the dependee) depends on another one (the depender) to accomplish a goal. There are nodes and links between them; the nodes represent the actors, and the links represent the dependencies. There are four types of dependencies: first – goal dependency serves to present a dependency to accomplish a goal; second – task dependency serves to present a task dependency between two actors; third – resource dependency serves to present a resource dependency where the depender depends on the dependee to offer it a resource; and fourth – softgoal dependency serves to present a dependency of performance between two actors.

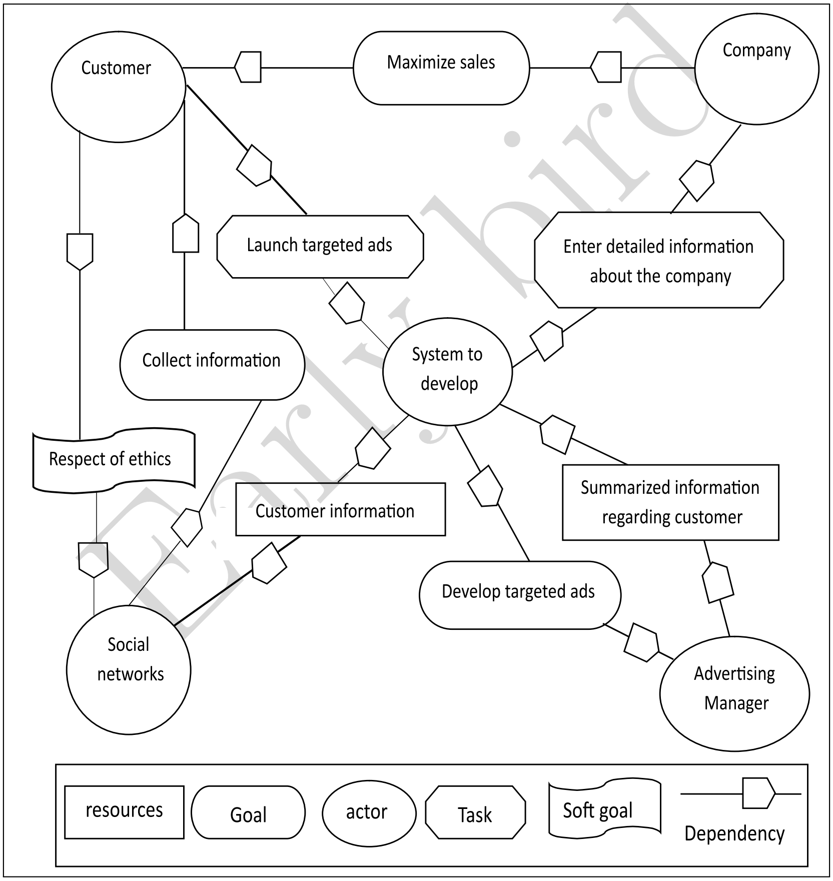

Figure 2. Strategic dependency (SD) model for sales company

Figure 2 represents the application of the strategic dependency (SD) model of the iStar method in the exemplary scenario of a sales company. The "company" depends on the "customer" for the goal of maximizing sales. The "system to be developed” depends on the ”company” to accomplish the task of offering him/her its information. The ”customer” depends on the ”system to develop” to launch targeted advertising. ”Social networks” depend on the ”customer” to collect information about their preferences. The ”system to be developed” depends on the ”social networks” to receive the customer information resource.

The “advertising manager” depends on the “system to be developed” to provide the summary information of the customers. The “system to be developed” depends on the “advertising manager” to accomplish the goal of developing targeted advertising.

3.3. Strategic rationale (SR) model

The strategic rationale (SR) model is used to detail the reasoning of each actor apart. A special focus is given to what happens inside an actor, which allows for a deep understanding of the process. Figure 3 shows the application of the strategic rationale model on the exemplary scenario of the sales company.

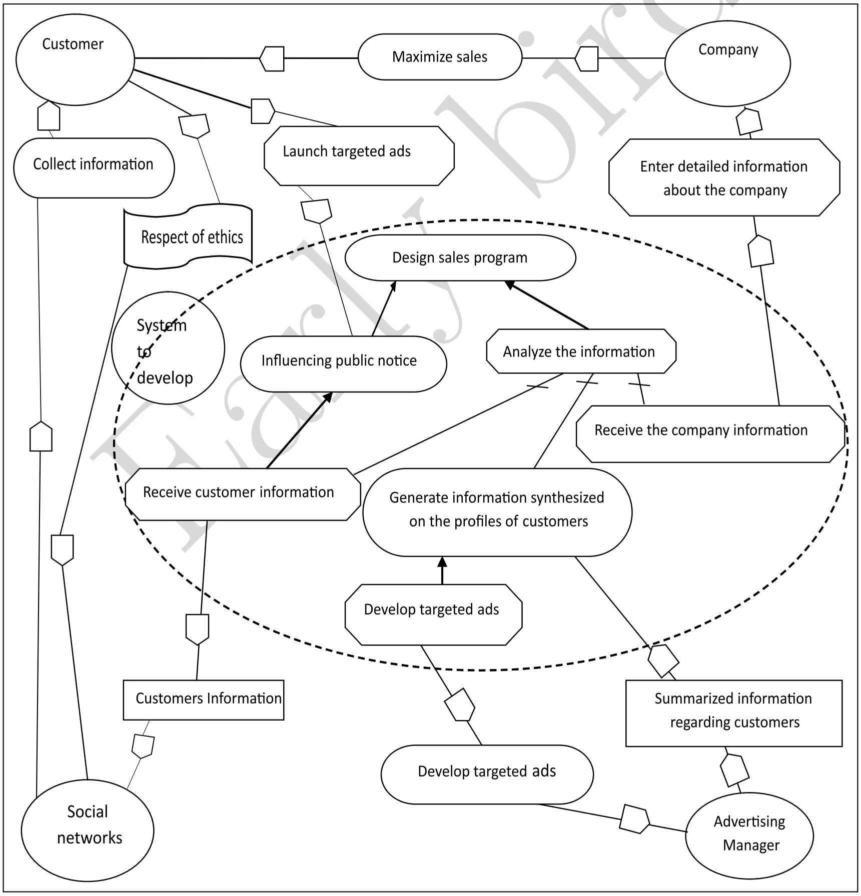

Figure 3. Strategic rationale (SR) model for sales company

4. Extension of iStar for big data projects

In this section, there is a presentation of BiStar (big data iStar), which consists of an extension of the iStar method for big data projects. This starts with clarifying the needs for an extension of iStar to support the elicitation of the requirements for big data projects; after the concepts to add are explained, then BiStar is performed on the sales company exemplary scenario.

4.1. Needs for extension of iStar

In this part, a light is shone on the situation as well as the important points that are considered critical. Elicitation is the most crucial step in RE [34]; if this is not well-done, it can lead to projects that do not respond well to the needs of the stakeholders. In the case of big data projects, this becomes more and more complicated. A big data project must not only meet a need but also respond in a very short time by processing a large amount (and a specific nature) of data (structured, semi-structured, or unstructured) [23]. It is on the big data properties (volume, velocity, and variety) that our study focuses to collect and model them by using BiStar in the RE stage. Also, [6, 29, 32] confirmed that there is a necessity for big data software to include all three parameters (functional feature, time constraint, verifiable during some period) to completely define the requirements specification for big data projects.

4.2. Added concepts to iStar

Based on the needs of the requirements for big data in the literature [10, 19, 23, 29, 32], the concepts to add are the execution time, the volume of the data to process, the variety of the data, and the durability of a goal. In the rest of this subsection, each concept will be explained (including details about the reason of use).

4.2.1. Required execution time

In a big data project, the execution time must be exact. A late result is considered to be an incorrect one. In the exemplary scenario of the sales company presented in Subsection 3.1, the stakeholder needs the goal "generate information synthesized on the profiles of costumer" and does not specify at what time it should be performed. The project will be well-done and finished. However, the goal must be met in 15 days; so, the project failed to satisfy the stakeholder's need. The conclusion is that the execution time of each goal must be specified at the beginning of a project.

4.2.2. Required volume of data to be processed

The volume of data is one of the most important features of big data projects. This is often large, but stakeholders are not aware of what can and cannot be done. Even using big data technologies like Hadoop and NoSQL systems, volume remains a crucial point when talking about zettabytes [19]. In the exemplary scenario of the sales company presented in Subsection 3.1, the stakeholder needs the goal "generate information synthesized on the profiles of costumers" and does not specify the volume of the data that must be processed. However, the goal needs to analyze 100 zettabytes of data. Important information regarding the goal is, therefore, incomplete. The volume of the data of each goal must also be specified at the beginning of a project.

4.2.3. Requirement of data variety

In big data projects, there is data with different presentations (structured, semi-structured, and unstructured data). Building a big data project that manipulates semi-structured data is different from one that employs unstructured data. In the example above (see Subsection 3.1), the stakeholder does not specify the nature of the data that must be processed. The goal needs to analyze semi-structured and unstructured data. Consequently, the nature of the data of each goal must be also specified at the beginning of a project.

4.2.4. Durability of goal

Big data projects are built to meet one's needs during specified times; it turns out that their goals may become dissatisfied for stakeholders, so reaching an agreement is important from the beginning of the time in which a requirement can be satisfied. In the exemplary scenario considered in Subsection 3.1, the stakeholder does not specify the durability of its goal. When the validation must be established on the project with the stakeholder, he says it is not what he wants; the goal must be satisfied during the whole session. So, the project failed to satisfy the need of the stakeholder. Also, the durability of a goal must be specified at the beginning of the project. iStar does not support the properties that are presented above; this does not allow for a complete and refined elicitation of the requirements for big data. To support big data projects by the iStar method, goals must be verified to check whether they are attached to their properties (the execution time, the volume of data to be processed, the variety of the data, and the durability of the goal). Figure 4 graphically shows the concepts that are added to the strategic dependency (SD) and strategic rationale (SR) models.

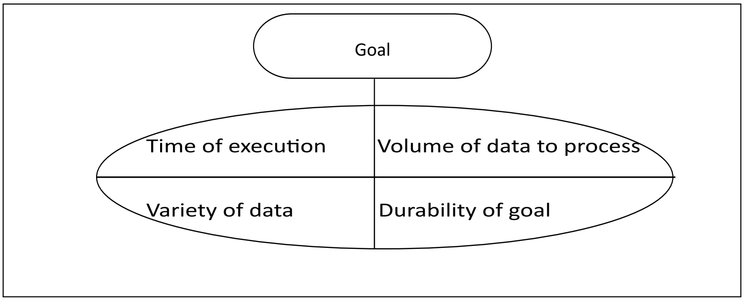

Figure 4. Concepts added to BiStar

The needs for extending iStar are different in each domain; this explains why there is such a large amount of extension of iStar [2, 16, 22, 25, 26]. It is necessary to make the model closer to reality by adding concepts for the purpose of improving the accuracy of the big data project that poses its specific challenges.

4.3. Applying BiStar on exemplary scenario

We keep the same meaning explained in Subsection 3.2; however, the new concepts are linked to the goal "develop targeted advertising" in BiStar, which means that this goal must be done within 10 days while analyzing 100 zettabytes of unstructured and semi-structured data. It also must be in operation during the session. In such a way, more completeness and refinement is given to the requirements. Figure 5 shows how to model the example of the sales company using the BiStar strategic dependency model.

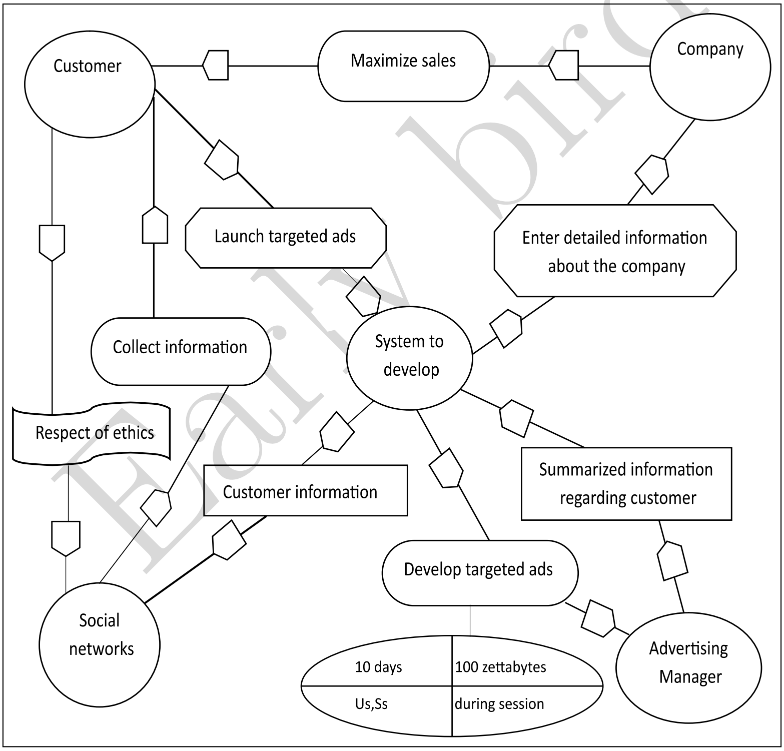

Figure 5. Strategic dependency model of BiStar for sales company

Figure 6 shows the application of BiStar's strategic rationale model on the example of the sales company. We keep the same meaning explained in Subsection 3.3; however, the new concepts in BiStar are also linked to the "design a sales program" and "generate synthesized information about the profile of customers" goals. The goal "design a sales program" must be done within 2 days by analyzing 30 petabytes of unstructured and semi-structured data, and it must be functional during the session. For the goal "generate information synthesized on the profiles of customers," this must be done within 15 days while analyzing 100 zettabytes of unstructured and semi-structured data, and it must be functional during the session.

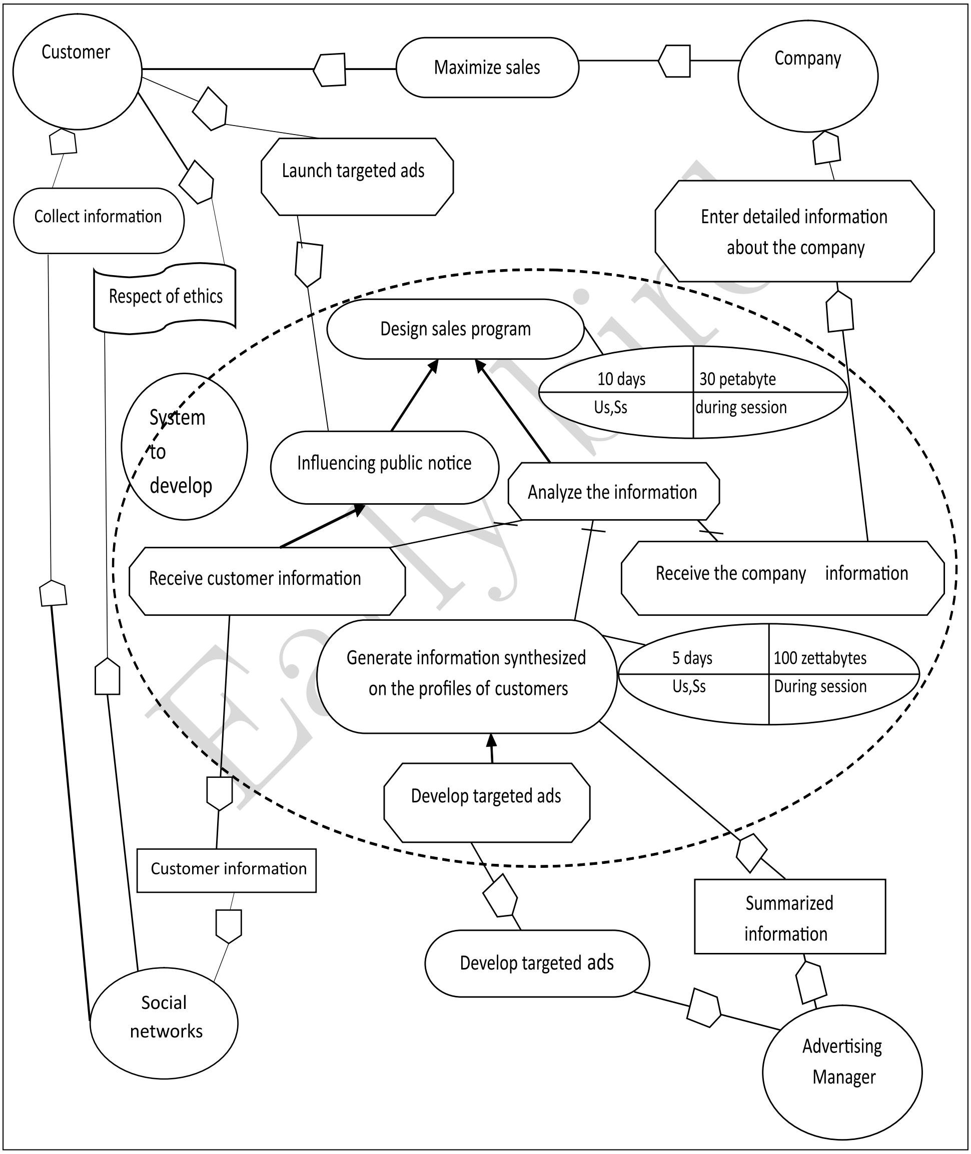

Figure 6. Strategic rationale model of BiStar for sales company

5. Formal checking of BiStar

This section aims at verifying the integrity property for BiStar by Bigraphs while extending iStar for big data projects. For this purpose, iStar should first be modeled by using two distinct Bigraphs; the first defines the semantics of its strategic dependency (SD) model, and the second is given for its strategic rationale (SR) model. Then, by checking the integrity of BiStar by deducing two extended Bigraphs from the former ones, we define the semantics of BiStar.

5.1. Bigraph definition

A Bigraph [17] is part of an emerging graphical formalism for designing, simulating, and analyzing ubiquitous computing systems. Structurally, a Bigraph is a graphical meta-model that emphasizes both the locality and connectivity of mobile systems. A Bigraph is formally defined by $G = (V, E, \mathit{ctrl}, GP, GL) : I-)J$, $I = (m, x)$, $J = (n, y)$, where:

- V and E represent finite sets of nodes and edges, respectively;
- $\mathit{ctrl} : V-)K$ is a control map that assigns a control to each node (signature K is a set of controls);
- GP and GL are place and link graphs, respectively;
- I and J represent the inner and outer names (interfaces), respectively, of Bigraph G. "m" and "n" are the numbers of sites and roots, respectively.

Bigraphs are used here to check the integration property of BiStar due to its five advantages: first – its clarity (it uses a graphical representation, which allows for better comprehension); second – it constitutes a mathematical base for specified systems so their extension or enrichment is made possible by using mathematical operations; third – its place graph is suitable for showing the hierarchy of the nodes; fourth – its two underlined structures (place and link graphs) are orthogonal, which independently specify the places and links of the system agents; and fifth – it is also possible to specify system behavior thanks to the reaction rules.

5.2. Formal semantics for iStar description

This subsection gives Bigraph-based definitions for the SD and SR models of iStar. Each element in these models has a formal semantics in terms of Bigraphs, allowing for a clear definition of the iStar extensions. Through the following formal definitions, more details are given in this regard.

5.2.1. Bigraph for SD model of iStar

Definition 1: the SD model semantics is defined by a Bigraph Bigsd = Nsd, Esd, Ctrlsd, Gpsd,Glsd: Isd -) Jsd, where:

- $Nsd = Actori, Actorf, Goal, Task, Resource, SoftGoal$
- $Esd = EGoal, ETask, EResource, ESoftGoal$
- $Ctrlsd(Actori) = Ctrlsd(Actorf) = atomic; 4$

- $Ctrlsd(Goal) = atomic; 1$
- $Ctrlsd(Task) = atomic; 1$
- $Ctrlsd(Resource) = atomic; 1$
- $Ctrlsd(SoftGoal) = atomic; 1$

Gpsd is the place graph that particularly represents the parent function defined as:

- prnt:site0UVsd-)VsdUregion0, knowing that:
- $prnt(Actor) = prnt(Ressource) = prnt(Goal) = prnt(SoftGoal) = prnt(Task) = region0$

Glsd is the graph of links that particularly represent the link function defined as:

- $link : UP-)EsdU$, P is the set of ports p11, p12, etc.
- $link(p11) = ERessource, link(p21) = Egoal, link(p31) = ETask, link(p41) = ESoftgoal$
- $Isd = (1,)$, without inner names and having one site that abstracts the possible insertion of other nodes.
- $Jsd = (1,)$, without outer names and having one region.

Figure 7 graphically shows the application of Bigraph on the SD model of iStar; there is one region, a site, the nodes (actor, resource, goal, task, and softgoal), their ports, and the relationships between them.

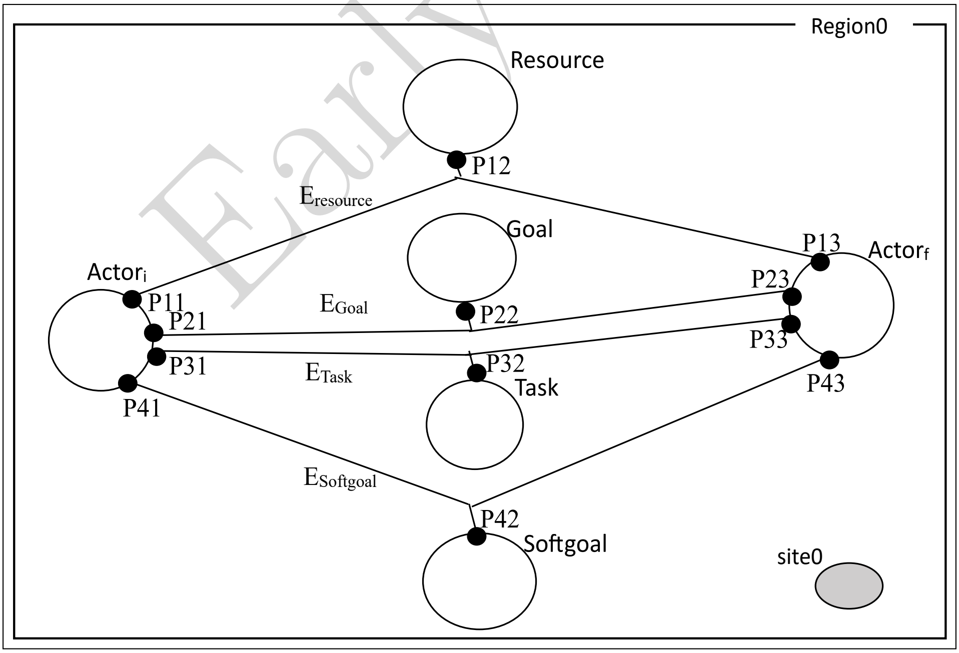

Figure 7. Bigraph-based definition of generic SD model of iStar

Figure 8 shows the application of the place graph on the SD model of iStar.

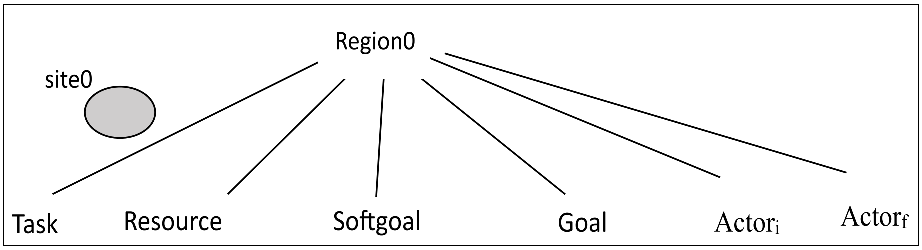

Figure 8. Place graph for SD model of iStar

5.2.2. Bigraph for SR model of iStar

Definition 2: the SR model semantics is defined by a Bigraph Bigsr = Nsr, Esr, Ctrlsr, Gpsr, Glsr: Isr -) Jsr, where:

- $Nsr = NsrUGoalActor, TaskActor, ResourceActor, SoftGoalActor$
- $Esr = EsrUETDL, EMEL$
- $Ctrl(GoalActor) = atomic; 2$
- $Ctrl(TaskActor) = atomic; 2$
- $Ctrl(ResourceActor) = atomic; 2$
- $Ctrl(SoftGoalActor) = atomic; 2$

Gpsr is the place graph that particularly represents the parent function defined as:

- $prnt : site0UVsr-)VsrUregion0$, knowing that:
  - $prnt(Actor) = prnt(Ressource) = prnt(Goal) = prnt(SoftGoal) = prnt(Task) = prnt(region1) = region0.$
  - $prnt(RessourceActor) = prnt(GoalActor) = prnt(SoftGoalActor) = prnt(TaskActor) = Actori.$

Glsr is the graph of links that particularly represent the link function defined as:

- $link : UP-)EsrU$, P is the set of ports p11, p12, etc.
- $link(p11) = ERessource, link(p21) = Egoal, link(p31) = ETask, link(p41) = ESoftgoal, link(p51) = ETDL, link(p61) = EMEL\ Isr = (1,)$, without inner names and having one site that abstracts the possible insertion of other nodes. $Jsr = (2,)$, without outer names and having one region.

Figure 9 graphically shows the application of Bigraph on the SR model of iStar; there is one region, a site, the nodes (actor, resource, goal, task, and softgoal), their ports, and the relationships between them.

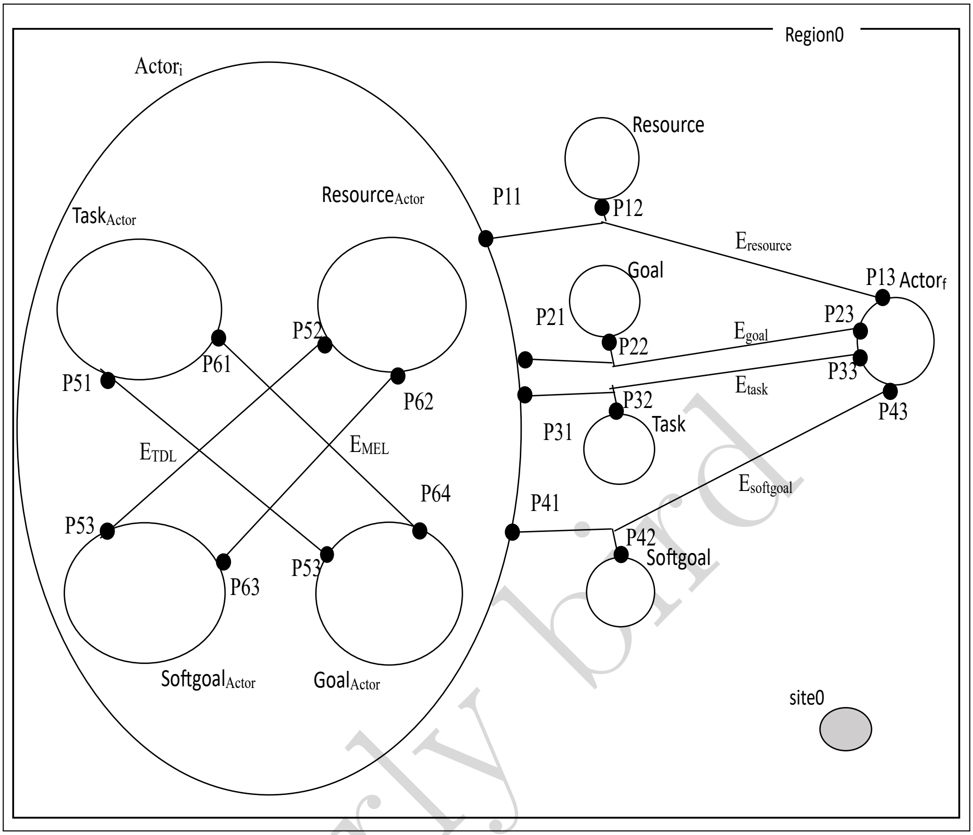

Figure 9. Bigraph-based definition of generic SR model of iStar

There is a hierarchy between the nodes; the node actor is a parent of the TaskActor, ResourceActor, SoftgoalActor, and GoalActor nodes. Figure 10 shows the application of the place graph on the SR model of iStar.

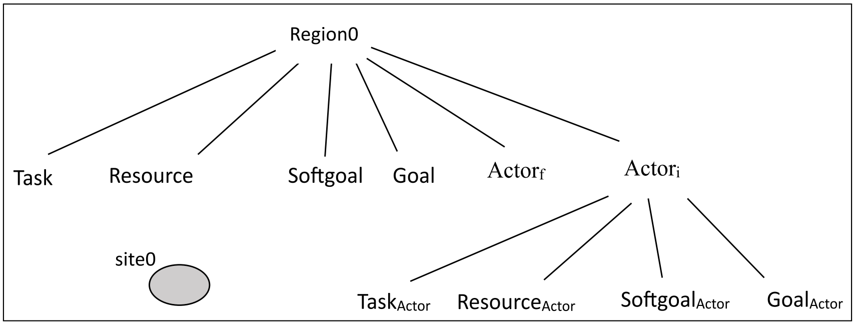

Figure 10. Application of place graph on SR model of iStar

5.3. Formal semantics for BiStar description

In this subsection, a light is shone on the idea that the Bigraphs that define the semantics of the SD and SR models in the context of BiStar are deduced from those of iStar by enriching their element sets by only one node type (BigdataRequirements) and one link type (EBigdataRequirements). Their respective formal definitions are given below:

5.3.1. Bigraph for SD model of BiStar

Definition 3: the SD model of the BiStar semantics is defined by a Bigraph Bigsdbi = Nsdbi, Esdbi, Ctrlsdbi, Gpsdbi,Glsdbi: Isdbi -) Jsdbi, where:

- $Nsdbi = NsdUBigdataRequirements$
- $Esdbi = EsdUEBigdataRequirements$
- $Ctrl(BigdataRequirements) = atomic; 1$

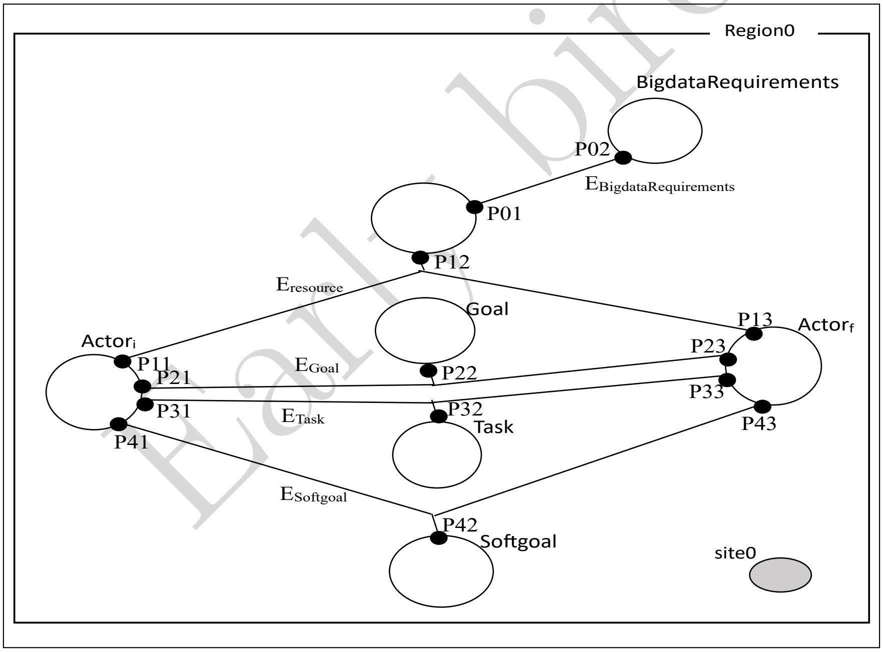

Figure 11. Bigraph-based definition of generic SD model of BiStar

Gpsdbi is the place graph that particularly represents the parent function, which is defined as follows:

- $prnt : site0UVsdbi-)VsdbiUregion0$, knowing that:
  - $prnt(Actori) = prnt(Actorf) = prnt(Ressource) = prnt(Goal) = prnt(SoftGoal) = prnt(Task) = prnt(BigdataRequirements) = region0$.

Glsdbi is the graph of the links that particularly represents the link function, which is defined as follows:

- $link : UP-)EsdbiU$, P is the set of ports p11, p12, etc.
- $link(p11) = ERessource, link(p21) = Egoal, link(p31) = ETask, link(p41) = ESoftgoal, link(p01) = EBigdataRequirements$
- $Isdbi = (1,)$, without inner names and having one site that abstracts the possible insertion of other nodes. $Jsdbi = (1,)$, without outer names and having one region.

Figure 11 graphically shows the Bigraph that is associated with the SD model of BiStar; there is one region, a site, the nodes (actori, actorf, resource, goal, task, softgoal, and BigdataRequirements), theirs port, and the relationships between them.

Figure 12 shows the application of the place graph on the SD model of iStar.

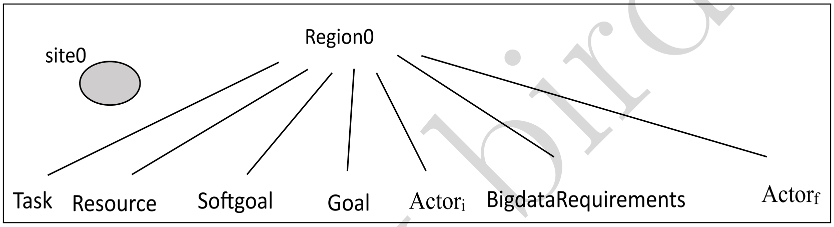

Figure 12. Place graph for SD model of BiStar

5.3.2. Bigraph for SR model of BiStar

Definition 4: the SR model of the BiStar semantics is defined by a Bigraph Bigsrbi = Nsrbi, Esrbi, Ctrlsrbi, Gpsrbi,Glsrbi: Isrbi -) Jsrbi, where:

- $Nsrbi = NsrUBigdataRequirementsActor$
- $Esrbi = EGoal, ETask, EResource, ESoftGoal, EBigdataRequirements$
- $Ctrl(BigdataRequirementsActor) = atomic; 1$

Gpsrbi is the place graph that particularly represents the parent function, which is defined as follows:

- $prnt : site0UVsrbi-)VsrbiUregion0$, knowing that:
  - $prnt(Actorf) = prnt(Ressource) = prnt(Goal) = prnt(SoftGoal) = prnt(Task) = prnt(Actori) = prnt(BigdataRequirements) = region0.$
  - $prnt(RessourceActor) = prnt(GoalActor) = prnt(SoftGoalActor) = prnt(TaskActor) = prnt(BigdataRequirements) = Actori.$

Glsrbi is the graph of links that particularly represents the link function, which is defined as follows:

- $link : UP-)EsrbiU$, P is the set of ports p11, p12, etc.

- $link(p11) = ERessource, link(p21) = Egoal, link(p31) = ETask, link(p41) = ESoftgoal, link(p51) = ETDL, link(p61) = EMEL, link(p01) = EBigdataRequirements.$
- $Isrbi = (1,)$, without inner names and having one site that abstracts the possible insertion of other nodes. $Jsrbi = (1,)$, without outer names and having one region.

Figure 13 graphically shows the Bigraph of the SR model of BiStar; there is a site, one region, primitive nodes (actorf, resource, goal, task, softgoal, and BigdataRequirements), and one possible composite node (Actori) that contains other nodes (GoalActor, ...), their ports, and the relationships between them.

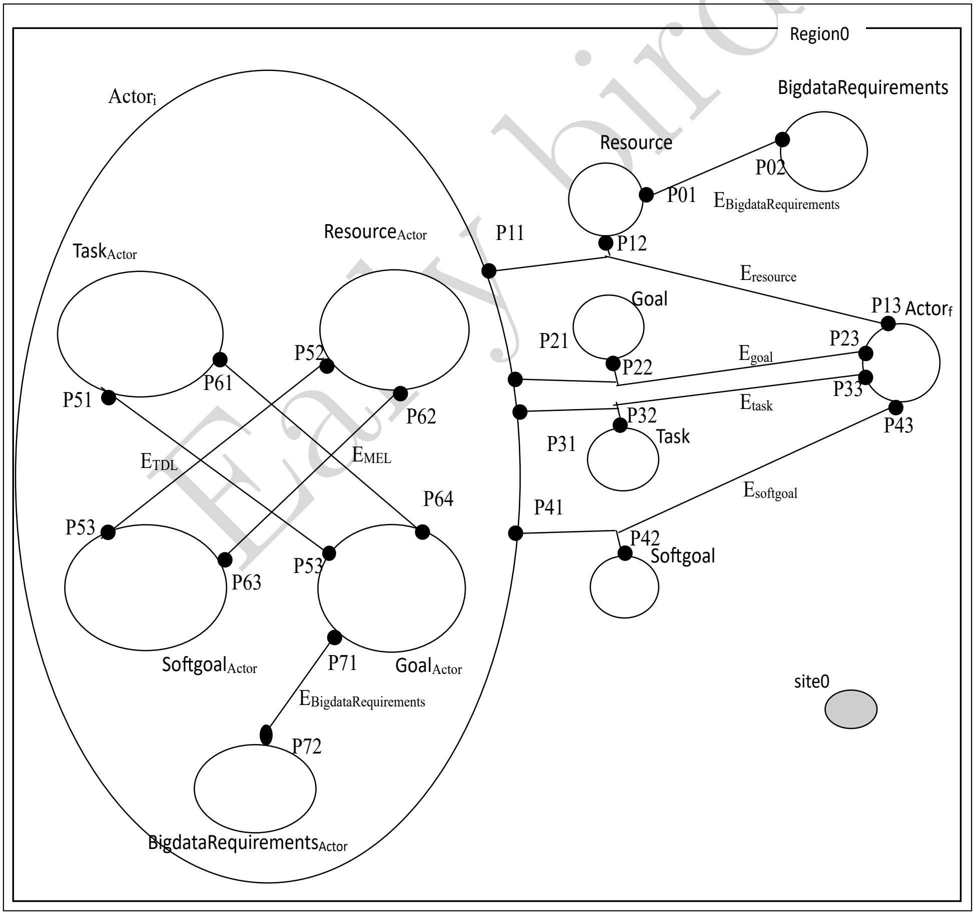

Figure 13. Bigraph-based definition of generic SR model of BiStar

There is also a hierarchy between the nodes; the Actori node is a parent of the TaskActor, ResourceActor, SoftgoalActor, GoalActor, and BigdataRequirementsActor nodes. This is illustrated in Figure 14.

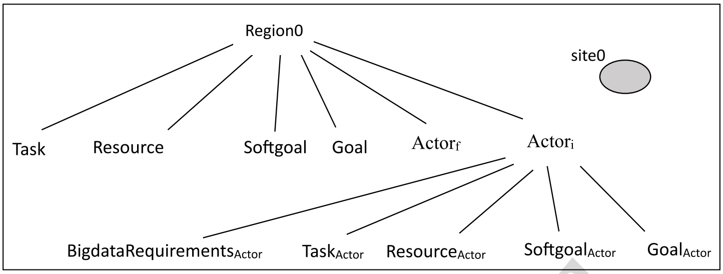

Figure 14. Place graph for SR model of BiStar

From the formal definition of the SD model of iStar (Bigsd), the formal definition of the SR model of iStar (Bigsr), the formal definition of the SD model of BiStar (Bigsdbi), and the formal definition of the SR model of BiStar (Bigsrbi), the following can be concluded:

- $Nsdbi = NsdUBigdataRequirements$
- $Esdbi = EsdUEBigdataRequirements$
- $Nsrbi = NsrUBigdataRequirements, BigdataRequirementsActor$
- $Esrbi = EsrUEBigdataRequirements$

These formal definitions of the BiStar models check the integrity property of BiStar. This contribution constitutes the first step towards BiStar formalization.

6. Conclusion and future work

In the literature, it has been shown that the construction of big data projects has shown a lot of failures; this is mainly due to the first stage of their development on which the following stages depend. This is why it has become obvious to review the methods and techniques that are used in the RE phase. RE methods have been extended to address specific problems for specific domains (security, health care, etc.). Big data is an emerging area that deserves a specific treatment due to its characteristics (volume, velocity, variety, veracity, and value). Many studies accentuate the importance of treating big data characteristics by the RE method.

In this work, we proposed an extension of the iStar method called BiStar that enables us to capture and manipulate the intrinsic characteristics (Vs) of big data applications. To do this, a synthesis of the work that has dealt with the problem under consideration was carried out. Then, the iStar method was illustrated on a big data case study (a sales company) to show its limitations and motivate the adopted extension. Also, a detailed description of the extension made to iStar is given.

Moreover, explanations are provided to clarify that BiStar supports the characteristics of big data projects quite well. In order to check that BiStar remains consistent with respect to iStar, a formal verification was made using Bigraphs to verify the integrity property of BiStar. The study shows that undertaking big data characteristics (Vs) is very useful during the elicitation of the requirements. As shown, Bistar catch more requirements for the Big data project.

Now, BiStar has not been applied to a lot of projects; as a perspective, more applications will improve BiStar with a large number of real projects. There is also the importance of undertaking big data characteristics during not only the elicitation step but also during the analysis, negotiation, documentation, and validation. Also, other properties of big data can be treated. As Bigraph provides a strong mathematical basis, another perspective is to build a Bigraph framework that allows for the validation of the integrated property of any iStar extension. This framework will help verify the integrity of other iStar extensions.

The improvements from the conference according to the reviewers' comments are applied in addition to searching in the literature to cover almost all of the relevant studies in the field. Modifications in the exemplary scenario were made in addition to the formal proofs on the integrity property of BiStar using Bigraphs. The study contributes to the general improvement of RE for big data projects by undertaking big data characteristics during elicitation. This helps to maximize the success rate of big data projects.

References

[1] Al-Najran N., Dahanayake A.: A requirements specification framework for big data collection and capture. In: *East European Conference on Advances in Databases and Information Systems*, pp. 12–19, Springer, 2015.

[2] Ali R., Dalpiaz F., Giorgini P.: Requirements-driven deployment, *Software & Systems Modeling*, vol. 13(1), pp. 433–456, 2014.

[3] Anderson K.M.: Embrace the challenges: Software engineering in a big data world. In: *2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, pp. 19–25, IEEE, 2015.

[4] Arruda D.: Requirements engineering in the context of big data applications, *ACM SIGSOFT Software Engineering Notes*, vol. 43(1), pp. 1–6, 2018.

[5] Arruda D., Madhavji N.H.: Towards a requirements engineering artefact model in the context of big data software development projects. In: *Proceedings of the IEEE International Conference on Big Data*, pp. 2232–2237, 2017.

[6] Arruda D., Madhavji N.H.: State of Requirements Engineering Research in the Context of Big Data Applications. In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 307–323, Springer, 2018.

[7] Attarha M., Modiri N.: Focusing on the importance and the role of requirement engineering. In: *The 4th International Conference on Interaction Sciences*, pp. 181–184, IEEE, 2011.

[8] Bersani M.M., Marconi F., Rossi M., Erascu M.: A tool for verification of big-data applications. In: *Proceedings of the 2nd International Workshop on Quality-Aware DevOps*, pp. 44–45, 2016.

[9] Chen H.M., Kazman R., Haziyev S., Hrytsay O.: Big data system development: An embedded case study with a global outsourcing firm. In: *2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, pp. 44–50, IEEE, 2015.

[10] Chen M., Mao S., Liu Y.: Big data: A survey, *Mobile networks and applications*, vol. 19(2), pp. 171–209, 2014.

[11] Clancy T.: The Standish Group Report CHAOS, *Project Smart*, pp. 1–16, 2014.

[12] Dalpiaz F., Franch X., Horkoff J.: istar 2.0 language guide, *arXiv preprint arXiv:160507767*, 2016.

[13] Djeddi C., Zarour N.E., Charrel P.J.: Extension of iStar for Big Data Projects. In: *International Conference on Advanced Aspects of Software Engineering*, pp. 9–16, 2018.

[14] Eridaputra H., Hendradjaya B., Sunindyo W.D.: Modeling the requirements for big data application using goal oriented approach. In: *2014 international conference on data and software engineering (ICODSE)*, pp. 1–6, IEEE, 2014.

[15] Goncalves E., Castro J., Araujo J., Heineck T.: A systematic literature review of istar extensions, *Journal of Systems and Software*, vol. 137, pp. 1–33, 2018.

[16] Guzman A., Martinez A., Agudelo F.V., Estrada-Esquivel H., Ortega J.P., Ortiz J.: A Methodology for Modeling Ambient Intelligence Applications using i* Framework. In: *iStar*, pp. 61–66, 2016.

[17] Jensen O.H., Milner R.: *Bigraphs and mobile processes (revised)*, Tech. rep., University of Cambridge, Computer Laboratory, 2004.

[18] Jutla D.N., Bodorik P., Ali S.: Engineering privacy for big data apps with the unified modeling language. In: *2013 IEEE International Congress on Big Data*, pp. 38–45, IEEE, 2013.

[19] Katal A., Wazid M., Goudar R.H.: Big data: issues, challenges, tools and good practices. In: *2013 Sixth international conference on contemporary computing (IC3)*, pp. 404–409, IEEE, 2013.

[20] Keele S., *et al.*: *Guidelines for performing systematic literature reviews in software engineering*, Tech. rep., Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[21] Lau L., Yang-Turner F., Karacapilidis N.: Requirements for big data analytics supporting decision making: A sensemaking perspective. In: *Mastering data-intensive collaboration and decision making*, pp. 49–70, Springer, 2014.

[22] Lockerbie J., Maiden N.A.M., Engmann J., Randall D., Jones S., Bush D.: Exploring the impact of software requirements on system-wide goals: a method using satisfaction arguments and i* goal modelling, *Requirements Engineering*, vol. 17(3), pp. 227–254, 2012.

[23] Madden S.: From databases to big data, *IEEE Internet Computing*, vol. 16(3), pp. 4–6, 2012.

[24] Madhavji N.H., Miranskyy A., Kontogiannis K.: Big picture of big data software engineering: with example research challenges. In: *2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering*, pp. 11–14, IEEE, 2015.

[25] Mazón J.N., Pardillo J., Trujillo J.: A model-driven goal-oriented requirement engineering approach for data warehouses. In: *International Conference on Conceptual Modeling*, pp. 255–264, Springer, 2007.

[26] Morandini M., Penserini L., Perini A., Marchetto A.: Engineering requirements for adaptive systems, *Requirements Engineering*, vol. 22(1), pp. 77–103, 2017.

[27] Noorwali I., Arruda D., Madhavji N.H.: Understanding quality requirements in the context of big data systems. In: *Proceedings of the 2nd International Workshop on BIG Data Software Engineering*, pp. 76–79, 2016.

[28] Nuseibeh B., Easterbrook S.: Requirements engineering: a roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*, pp. 35–46, 2000.

[29] Otero C.E., Peter A.: Research directions for engineering big data analytics software, *IEEE Intelligent Systems*, vol. 30(1), pp. 13–19, 2014.

[30] Ramingwong L.: A review of requirements engineering processes, problems and models, *International Journal of Engineering Science and Technology*, vol. 4(6), 2012.

[31] Sachdeva V., Chung L.: Handling non-functional requirements for big data and IOT projects in Scrum. In: *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pp. 216–221, IEEE, 2017.

[32] Sharma K., *et al.*: Quality issues with big data analytics. In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 3589–3591, IEEE, 2016.

[33] Supakkul S., Zhao L., Chung L.: GOMA: Supporting big data analytics with a goal-oriented approach. In: *2016 IEEE International Congress on Big Data (BigData Congress)*, pp. 149–156, IEEE, 2016.

[34] Van Lamsweerde A.: Goal-oriented requirements engineering: A guided tour. In: *Proceedings fifth ieee international symposium on requirements engineering*, pp. 249–262, IEEE, 2001.

[35] Werneck V.M.B., Oliveira A.d.P.A., do Prado Leite J.C.S.: Comparing GORE Frameworks: i-star and KAOS. In: *WER*, 2009.

[36] Yu E.: Modeling Strategic Relationships for Process Reengineering., *Social Modeling for Requirements Engineering*, vol. 11(2011), pp. 66–87, 2011.

[37] Zave P.: Classification of research efforts in requirements engineering, *ACM Computing Surveys (CSUR)*, vol. 29(4), pp. 315–321, 1997.

[38] Zowghi D., Coulin C.: Requirements elicitation: A survey of techniques, approaches, and tools. In: *Engineering and managing software requirements*, pp. 19–46, Springer, 2005.

Affiliations

Chabane Djeddi
Constantine 2-A. Mehri University, LIRE Laboratory, Algeria
chabane.djeddi@univ-constantine2.dz

Nacer Eddine Zarour
Constantine 2-A. Mehri University, LIRE Laboratory, Algeria
nasro.zarour@univ-constantine2.dz

Pierre-Jean Charrel
Toulouse 2 Jean Jaures University, IRIT Laboratory, France charrel@univ-tlse2.fr

Received: ??.??.2021
Revised: ??.??.2021
Accepted: ??.??.2021