Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL APPROACH TO GENERATE DISTRIBUTED GLOBAL AND LOCAL USE CASES: A NEW NOTATION

[1]MOHAMMAD I. MUHAIRAT, [2]AKRAM ABDELQADER, [3]MOHAMMAD AL RAWAJBEH
[1]Department of Software Engineering, Al-Zaytoonah University of Jordan, Amman, Jordan
[2]Department of Multimedia Systems, Al-Zaytoonah University of Jordan, Amman, Jordan
[3]Department of Computer Networks, Al-Zaytoonah University of Jordan, Amman, Jordan
E-mail: [1]drmohairat@zuj.edu.jo, [2]akarma@zuj.edu.jo, [3]m.rawajbeh@zuj.edu.jo

ABSTRACT

Requirements analysis is very important phase of the software development life cycle (SDLC). It is used to build domain or conceptual models from requirements analysis. These models are helpful for defining stakeholders and functional requirements. Also, these functional requirements are easy to define for local systems, but they are very difficult to define for distributed stakeholders or systems. In this paper, we proposed a new approach to generate integrated requirements for a distributed system. This approach is composed of four steps: The first one is to generate an event table with a proposed template that suitable for requirements collection. This event table will be filled for each department and branch. The second one is to generate a normalized event table for all departments and branches. The third one is building and analyzing an integrated table to define the redundancy of functional requirements for all departments and branches. The final one is to generate and propose a new UML notation for distributed functional requirements. This approach is applied to a real ERP system as a case study and shows a significant result.

Keywords: *Analysis Models, Distributed requirements, Transformation, Distributed Global Use Case, Distributed Local Use Case, Event Table, Use Case Notation.*

1. INTRODUCTION

The requirements gathering and analysis phase, is the most important phase in the software development life cycle (SDLC). In this phase, analysts communicate with stakeholders to acquire the requirements for a project. The requirements serve as inputs for upcoming phases of the SDLC. They need to be as complete as possible. Changes to requirements during subsequent phases of the project can be more expensive than during the requirements phase. A major problem with functional requirements is that of unstated or implicit requirements, which stakeholders presume the analyst knows. Also, if the functional requirements and stakeholders are distributed, this will cause many challenges for the analyst in order to build system models. Some stakeholders have trouble understanding the requirements presented in the form of text and analysts may also have some problems to build system models from the form of text.

While many tools exist to render and visualize requirements models, not many tools exist to assist the requirements analyst during the requirements analysis phase. For the purpose of visualizing functional requirements, the system models such as a use case model are valuable. A use case model was proposed by Jacobson, et al. [18], they proposed an approach to deal with documenting the stakeholder functional requirements in an easy way that leads to discover what classes are needed to be implemented. The same approach was developed by Cockburn [24] and other authors [19]. A use case is a description of user-system interaction expressed in natural language.

Use cases can be identified by using an events decomposition technique or by focusing on users and their goals. Also, a system analyst can identify an event table by using manual or brain storming techniques. Event table is a catalogue of use cases that lists events in rows and key pieces of information about each event in columns. The event table was used as the preferred event analysis technique in the requirements engineering process as in [20-22]. Event table is a list of actions that lists events in rows and the information about each event in columns. The results of event analysis are documented in an event table. The event table used in the both structured approach and object-oriented approach, in structured approach, event analysis

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

recognizes a basic set of processes. While it is an event discourses an essential use-case in object-oriented approach as in [23].

This paper includes the following phases: first the requirements for each branch will be entered into the proposed customized event table. Secondly, the normalization process will be applied on the customized event table. Thirdly, the integration process is used to identify and classify the requirements for local and global requirements. This work will adds significant improvements in the software engineering process by generating a new UML notation for local and global use cases. In addition, this work will be helpful in software analysis and design for classifying the requirements in the distributed systems. This research will address many literature problems such as the weight of the requirements, the redundancy of the requirements, the importance of the requirements and the local and global requirements while most applications today are works for local and global environment. Besides that, the main contributions of this paper are:

- Propose an adaptive event table to meet the proposed transformation approach.
- Propose a transformation model that generates use cases from an event table, even if the requirements are distributed globally or locally.
- Propose a novel UML notation for a distributed global and local use case.

This paper is structured as the following: Section 1 presents an introduction about the proposed approach and the importance of requirements and the event table. Section 2 presents a survey of the related work in this field. The methodology and the proposed work phases are presented in section 3. Section 4, illustrates an example that applying the phases of the proposed approach using rent a car system as a case study. Finally the conclusion and future work presented in section 5.

2. RELATED WORK

The process of generating the requirements analysis from the row requirements is one of the difficult tasks in the software development process. In addition, the transmission of analyzing requirements to design or implementation is also considered as a difficult activity during the software development process. The complexity related to it can cause many errors mainly if the ambiguity of requirements exists. To eliminate this ambiguity, many researches were done for providing semi-automated ways to generate analysis and design models or code artifacts. The traditional requirement engineering is facing many problems such as requirement prioritization, validation, communication gaps, over scoping, and customer involvement [25]

Various approaches were proposed to solve this problem. Some researchers focused on requirements in respect to their precision by defining new languages for this purpose, such as in [6] and [7]. The disadvantage of these approaches is that they do not propose further code generation. Some approaches can be distinguished by their use of use case scenarios for requirements specification. Giganto et al. [8] propose an algorithm to identify use case sentences from requirements specifications written in controlled natural language and as a result–automatically obtain classes from use cases. Whereas Mustafiz et al. [9] propose transformation rules for creating different types of behavioral diagrams from use case scenarios.

Deeptimahanti et al. [10] suggest analyzing requirements specifications presented in natural language, using Natural Language Processing techniques and generate use case diagrams and class models. Most of the solutions focused on code generation use graphical notations like UML [11] to specify static and dynamic aspects of systems. One example is the open source AndroMDA [12] code generation framework which supports the Model Driven Architecture [13] paradigm. As input,

Hussain and Clear in [27], they reported the results of a study in two global software development based on client-vendor relationships. In both cases, they used spreadsheet technology for use locally and for bridging across sites, these spreadsheet files were embedded within different collaborative technologies. They are practices co-evolved with the spreadsheet artefacts to managing requirements change. Each single spreadsheet cell content has far wider implications for global requirements change management. Other researchers show the importance of requirements agility in global environment [26] they reflected the differences of ‘traditional’ and agile requirements changes.

AndroMDA takes UML models from various CASE tools and provides generation of deployable applications and software components. In turn, textual specifications used as input, turn out to be often too formalized and thus difficult to understand by the end-users where the purpose is specifying requirements [14]. There are also numbers of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solutions focusing directly on graphical user interface generation. Many of them, however, use notations designed specifically for this purpose, e.g. the one proposed by Falb, et al. [15]. Some other solution uses the existing software development notations. However, these notations operate at significantly lower level of abstraction than requirements specification, as e.g. proposed by Janssen et al. [16]. There also exists a solution based on requirements scenarios [17], but it only generates a graphical user interface mock ups.

3. PROPOSED APPROACH

In this paper, we proposed a new novel approach to generate a global and local use cases for distributed systems. The methodology and the proposed work framework are illustrated in Figure 1. This approach composed of five phases: the first phase is filling up the requirements in the proposed event table. The second phase is the normalization process to normalize the requirements in an event table and eliminating the redundancy rows and actors in the local event table at branch level. The third phase is the integration process to concatenate the distributed event tables into integrated one and to define the local and global definitions. The fourth phase is to generate a new Use Case Notation for Global and Local distributed requirements. Finally, the result of this approach is to construct a new distributed notation for Global and Local distributed use cases.

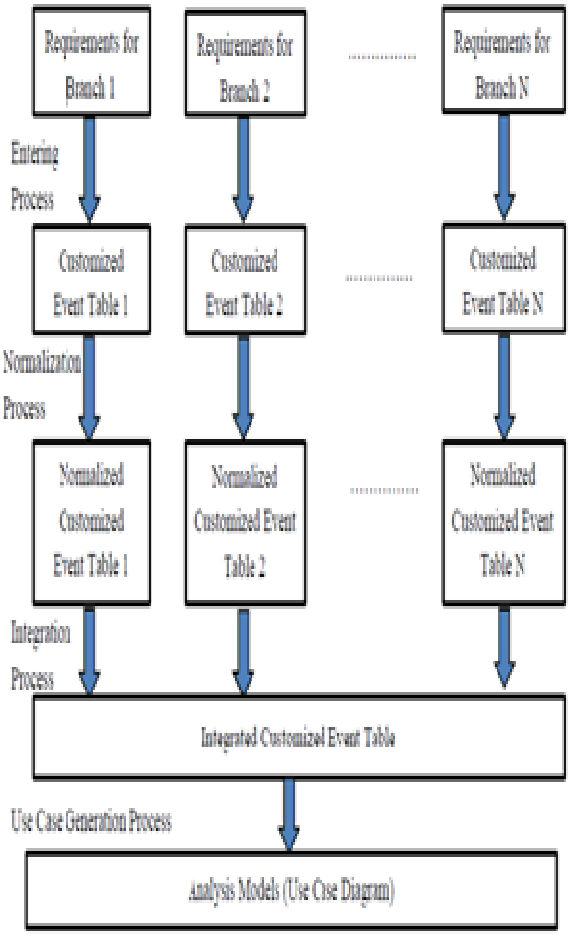

Figure 1. The methodology and the proposed work framework.

This approach will go throw many processes to get an analysis models mainly use case model (Global/Local Distributed Use Cases, Use Cases and Actors) and will consists of the following processes.

3.1 Entering Process

In this process, we propose the adaptive event table that is customizable and adapted to contain the requirements of local and global levels. This requirements entered by the system analyst as demonstrated in Table 1. This event table will be entered for each department in the branch and for all branches in the distributed system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. Customized event table

E	IM	SS	BV	OM	O	BN	BR	DN	DR

Where, E: is Event, IM: Input Message, SS: Source/Subject; BV: Behavior/Verb; OM: Output Message; O: Object; BN: Branch Number; DN: Department Number; BR: Branch Redundancy; and DR: Department Redundancy.

3.2 Normalization Process

This process will normalize each table by eliminating the redundancy of use cases in each branch and for each department. This normalization algorithm is demonstrated in Figure 2 and the result will be shown as assigned symbols in DR attribute in a new normalized customized event table.

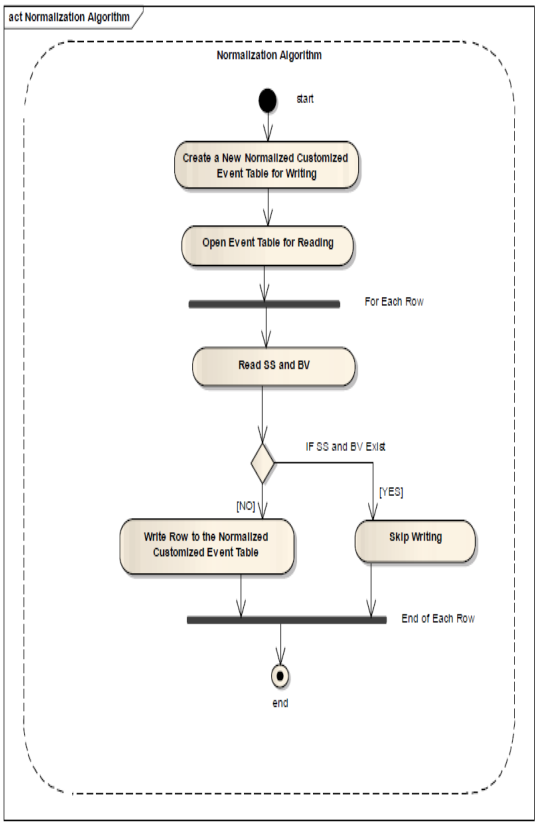

Figure 2. Normalization Algorithm

3.3 Normalization Process

After the normalization process, the row redundancy and data redundancy in each event table at department level will be eliminated. These normalized event tables will be integrated by applying integration and global use case definition Algorithms as demonstrated in Figure 3 and Figure 4. Also, the result will be shown as assigned symbol in BR attribute in a new integrated normalized customized event table.

Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

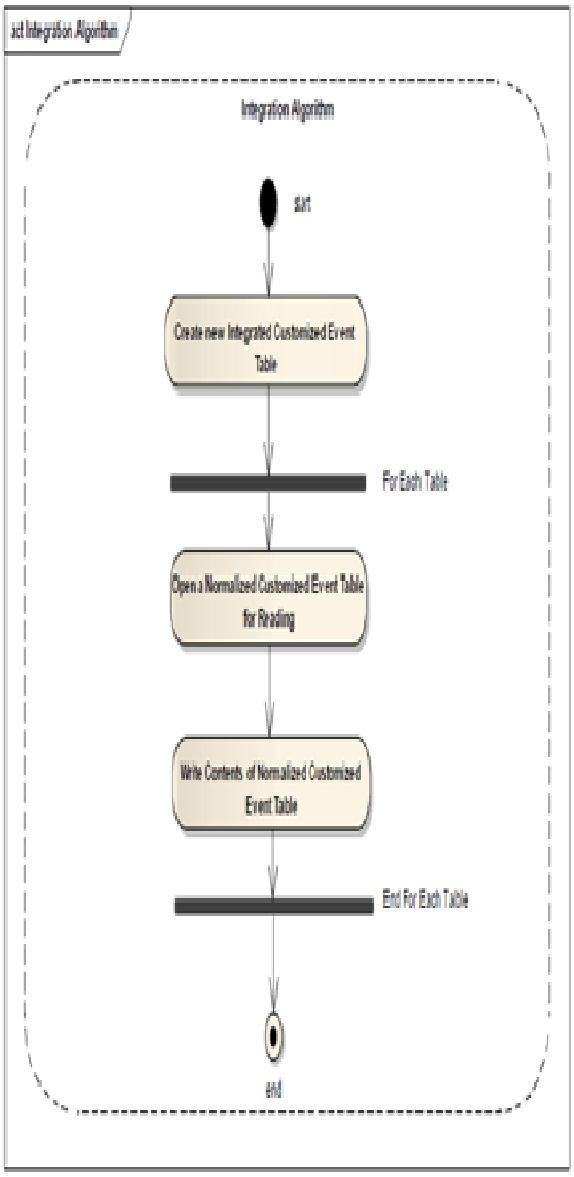

Figure 3. Integration Algorithm.

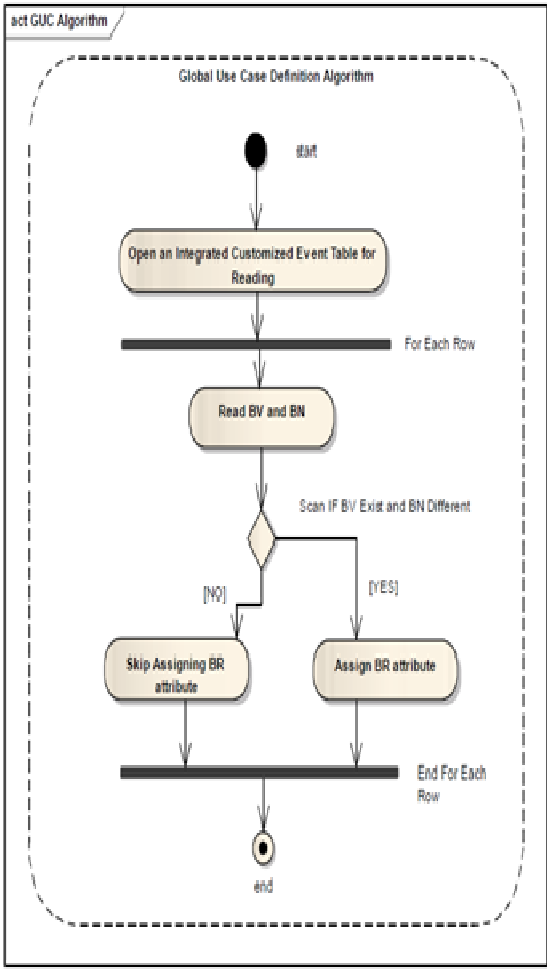

Figure 4. Global Use Case Definition Algorithm.

3.4 Normalization Process

This process will generate the use case notations such as: global distributed use cases, local distributed use cases, use cases and actors with their relations. The use case generation process algorithm is demonstrated in Figure 5. The results of this algorithm shown as a new notation use case diagram as demonstrated in Figure 6.

Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

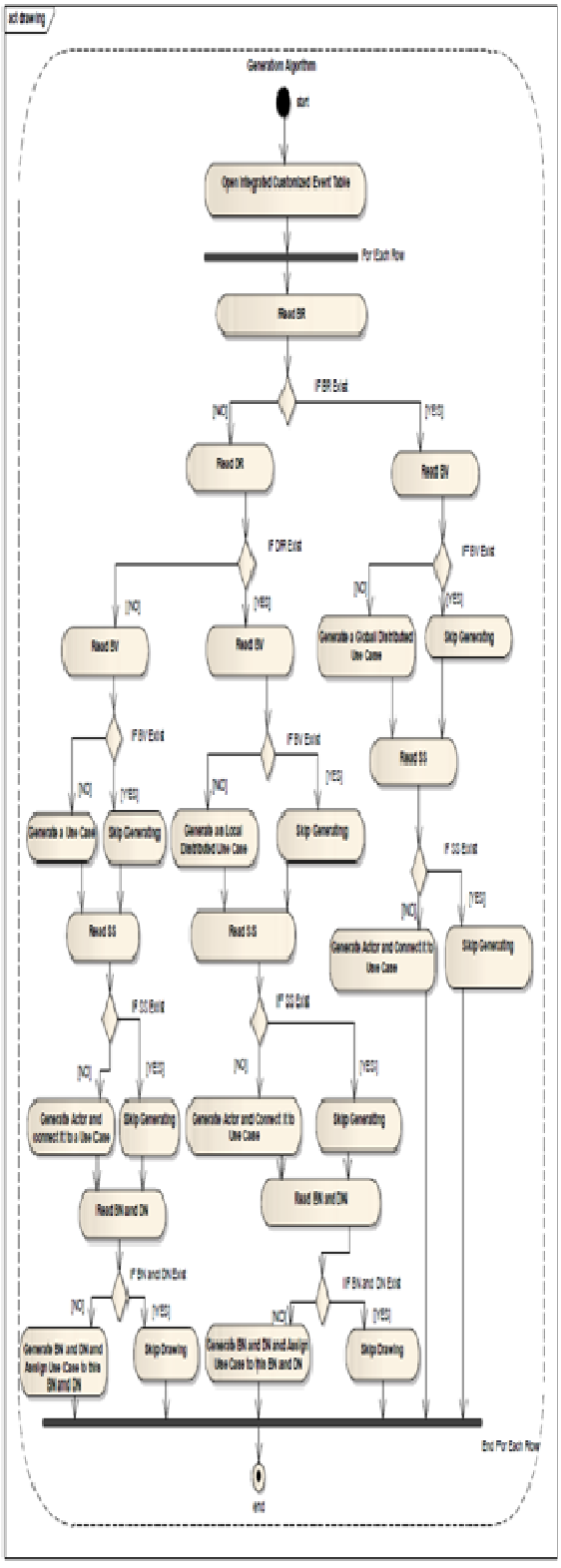

Figure 5. Use Case Generating Algorithm

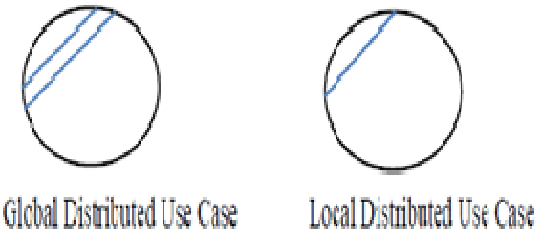

Figure 6. Notation of Global/Local Distributed Use Cases

To illustrate the proposed methodology and the framework we present a case study example that represents a distributed system for one of a "Car Rental System" in Jordan.

4. CASE STUDY OF THE PROPOSED APPROACH

This case study takes a car rental system a broad Jordan called "JOR Rent a Car". This company has many branches distributed across Jordan. The proposed approach will be implemented as the following:

4.1 Entering Process

The system analyst will enter his requirements in our template and the result will be saved in the following customized event tables as shown in Table 2 and Table 3.

Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2. Customized event table for branch 1.

No	E	DI	SS	SV	COI	O	BN	DN
1	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	1	1
2	Customer want to login	Login data	Customer	login	------	Customer	1	1
3	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	1	1
4	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	1	1
5	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	1	1
6	Customer cancel his reservation	Reservation data	Customer	Cancel reservation	------	Reservation	1	1
7	Customer view rental agreement	Rental agreement search data	Customer	View rental agreement	Rental agreements details	Agreement	1	1
8	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	1	1
9	Employee view customer registration	Registration data	Employee	View registration	Registration details	Registration	1	2
10	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	1	1
11	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	1	2
12	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	1	2
13	Employee search a car	Car data	Employee	Search a car	Car details	Car	1	2
14	Employee view rental agreement	Agreement data	Employee	View rental agreement	Agreement details	Agreement	1	2
15	Employee add a car	Car data	Employee	Maintains car	Car details	Car	1	1
16	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	1	1
17	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	1	1
18	Customer view rental agreement	Rental agreement search data	Customer	View rental agreement	Rental agreements details	Agreement	1	1
19	Manager view monthly rental report	Rental data	Manager	View monthly rental	Rental details	Rental	1	1
20	Employee search a car	Car data	Employee	Search a car	Car details	Car	1	2
21	Billing employee view daily billing reports	Bill data	Billing employee	View daily bill	Daily bill details	Bill	1	3
22	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	1	3

Table 3. Customized Table for Branch 2

No	E	DI	SS	SV	COI	O	BN	DN
1	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	2	1
2	Customer want to login	Login data	Customer	login	------	Customer	2	1
3	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	2	1
4	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	2	1
5	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	2	1
6	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	2	1
7	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	2	1
8	Employee view customer profile	Customer profile data	Employee	View profile	Profile details	Customer	2	1
9	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	2	1
10	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	2	1
11	Employee search a car	Car data	Employee	Search a car	Car details	Car	2	1
12	Employee add a car	Car data	Employee	Maintains car	Car details	Car	2	1
13	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	2	1
14	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	2	1
15	Employee view customer profile	Customer profile data	Employee	View profile	Profile details	Customer	2	1
16	Customer want to login	Login data	Customer	login	------	Customer	2	1
17	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	2	3
18	Technical employee generate maintenance report	Maintenance data	Technical employee	Generate maintenance report	Maintenance details	Car-Maintenance	2	4
19	Manager view monthly car maintenance reports	Maintenance data	Manager	View monthly maintenance	Maintenance details	Car-Maintenance	2	4

4.2 Normalization Process

This process will be implemented for each customized event table by using a normalization algorithm and the result will be saved in a normalized customized event table as demonstrated

Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in the following tables for each branch as shown in Table 4 and Table 5.

Table 4. Normalized Customized Event Table for Branch 1.

No	E	IM	SS	BV	OM	O	BN	BR	DN	DR
1	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	1		1	
2	Customer want to login	Login data	Customer	login	----	Customer	1		1	
3	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	1		1	√
4	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	1		1	√
5	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	1		1	
6	Customer cancel his reservation	Reservation data	Customer	Cancel reservation	------	Reservation	1		1	
7	Customer view rental agreement	Rental agreement search data	Customer	View rental agreement	Rental agreement details	Agreement	1		1	√
8	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	1		1	
9	Employee view customer registration	Registration data	Employee	View registration	Registration details	Registration	1		2	
10	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	1		2	
11	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	1		2	√
12	Employee search a car	Car data	Employee	Search a car	Car details	Car	1		2	√
13	Employee view rental agreement	Agreement data	Employee	View rental agreement	Agreement details	Agreement	1		2	√
14	Employee add a car	Car data	Employee	Maintains car	Car details	Car	1		1	
15	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	1		1	
16	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	1		1	
17	Manager view monthly rental report	Rental data	Manager	View monthly rental	Rental details	Rental	1		1	
18	Billing employee view daily billing reports	Bill data	Billing employee	View daily bill	Daily bill details	Bill	1		3	
19	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	1		3	

Table 5. Normalized Customized Event Table for Branch 2

No	E	IM	SS	BV	OM	O	BN	BR	DN	DR
1	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	2		1	
2	Customer want to login	Login data	Customer	login	------	Customer	2		1	
3	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	2		1	√
4	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	2		1	√
5	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	2		1	
6	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	2		1	
7	Employee view customer profile	Customer profile data	Employee	View profile	Profile details	Customer	2		2	
8	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	2		2	
9	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	2		2	√
10	Employee search a car	Car data	Employee	Search a car	Car details	Car	2		2	√
11	Employee add a car	Car data	Employee	Maintains car	Car details	Car	2		1	
12	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	2		1	
13	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	2		1	
14	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	2		3	
15	Technical employee generate maintenance report	Maintenance data	Technical employee	Generate maintenance report	Maintenance details	Car-Maintenance	2		4	
16	Manager view monthly car maintenance reports	Maintenance data	Manager	View monthly maintenance	Maintenance details	Car-Maintenance	2		4	

4.3 Integration Process

This process will integrate all the normalized event tables by implementing integration and defining global/local distributed use cases algorithms as mentioned above. Table 6 shows the results in one Integrated Normalized Customized Event Table.

Journal of Theoretical and Applied Information Technology
15th September 2016. Vol.91. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6. Integrated Normalized Customized Event Table.

No	E	IM	SS	BV	OM	O	BN	BR	DN	DR
1	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	1	√	1	
2	Customer want to login	Login data	Customer	login	-------	Customer	1	√	1	
3	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	1	√	1	√
4	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	1	√	1	√
5	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	1	√	1	
6	Customer cancel his reservation	Reservation data	Customer	Cancel reservation	----------	Reservation	1		1	
7	Customer view rental agreement	Rental agreement search data	Customer	View rental agreement	Rental agreements details	Agreement	1		1	√
8	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	1	√	1	
9	Employee view customer registration	Registration data	Employee	View registration	Registration details	Registration	1		2	
10	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	1	√	2	
11	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	1	√	2	√
12	Employee search a car	Car data	Employee	Search a car	Car details	Car	1	√	2	√
13	Employee view rental agreement	Agreement data	Employee	View rental agreement	Agreement details	Agreement	1		2	√
14	Employee add a car	Car data	Employee	Maintains car	Car details	Car	1	√	1	
15	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	1	√	1	
16	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	1	√	1	
17	Manager view monthly rental report	Rental data	Manager	View monthly rental report	Rental details	Rental	1		1	
18	Billing employee view daily billing reports	Bill data	Billing employee	View daily bill	Daily bill details	Bill	1		3	
19	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	1	√	3	
20	Customer want to register	Registration data	Customer	Maintains registration	Registration details	Registration	2	√	1	
21	Customer want to login	Login data	Customer	login	-------	Customer	2	√	1	
22	Customer want to check reservation	Reservation data	Customer	Check reservation	Reservation status	Reservation	2	√	1	√
23	Customer search a car	Car data	Customer Employee	Search a car	Car details	Car	2	√	1	√
24	Customer add new reservation	Reservation data	Customer	Maintains reservation	Reservation details	Reservation	2	√	1	
25	Customer make a payment	Payment data	Customer	Maintains payment	Payment details	Payment	2	√	1	
26	Employee view customer profile	Customer profile data	Employee	View profile	Profile details	Customer	2		2	
27	Employee make an offer	Offer data	Employee	Create offer	Offer details	Offer	2	√	2	
28	Employee want to check reservation	Reservation data	Employee	Check reservation	Reservation status	Reservation	2	√	2	√
29	Employee search a car	Car data	Employee	Search a car	Car details	Car	2	√	2	√
30	Employee add a car	Car data	Employee	Maintains car	Car details	Car	2	√	1	
31	Employee view daily rental report	Rental data	Employee	View daily rental report	Rental details	Rental	2	√	1	
32	Employee generate rental agreement	Agreement data	Employee	Maintains agreement	Agreement details	Agreement	2	√	1	
33	Billing employee confirm payment	Payment data	Billing employee	Confirm payment	Payment details	Payment	2	√	3	
34	Technical employee generate maintenance report	Maintenance data	Technical employee	Generate maintenance report	Maintenance details	Car-Maintenance	2		4	
35	Manager view monthly car maintenance reports	Maintenance data	Manager	View monthly maintenance	Maintenance details	Car-Maintenance	2		4	

4.4 Use Case Generation Process

The results of this process and the overall algorithm are shown in Figure 7. The proposed new notations are illustrated in the figure below. These notations are a new novel method to identify and manage the local and global requirements in the software engineering requirement process.

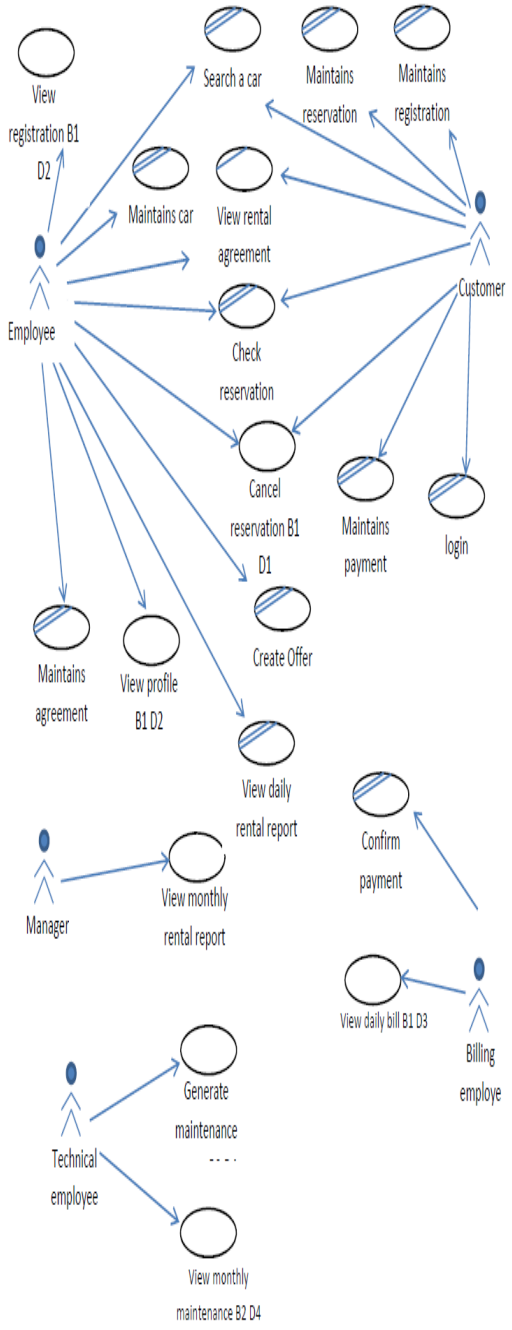

Figure 7. Rent a Car Use Case Diagram with New Notations

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION AND FUTURE WORK

This paper proposes a novel method and algorithm to identify and managing an enterprise requirements. These requirements are identified and classified into standard requirements, local requirements or global requirements. This identification will add a significant improvements of the enterprise system analysis since, this algorithm identified the department location, the branch and the importance of the entered requirement. This approach will help the software designer and analysts in the software engineering process.

The results of the case study show the importance of the proposed work in identifying the redundant requirements and to discard them, in addition to classifying the requirement into local and global one and to what branch or stakeholder it belongs to. This is illustrated in the tables resulting from applying the proposed algorithms.

Besides that, the proposed work needs a good starting point of requirements collection based on categorized questions that are suitable to the system domain.

The future work is to implement the proposed algorithm and measure the accuracy and the performance of different type of requirements.

REFRENCES:

[1] Bruegge B, Dutoit AH (2004) Object-oriented software engineering using UML, patterns, and Java, 2nd edn. Prentice Hall

[2] Kleppe A, Warmer J, Bast W (2003) MDA explained—the model driven architecture: practice and promise. Addison-Wesley, Boston

[3] Abbott RJ (1983) Program design by informal English descriptions. Com ACM 26(11):882–894

[4] Larman C (2004) Applying UML and patterns, 3rd edn. PrenticeHall, New Jersey

[5] IEEE Std. 830-1998, IEEE Standard for Software Requirement Specification, 1998.

[6] P. Shaker, J. Atlee, and S. Wang, "A feature-oriented requirements modeling language,"inRequirements Engineering Conference (RE), 2012 20th IEEE International, 2012, pp. 151–160.

[7] M. El-Attar and J. Miller, "AGADUC: Towards a More Precise Presentation of Functional Requirement in Use Case Mod," in Software Engineering Research, Management and Applications, 2006. Fourth International Conference on, 2006, pp. 346–353.

[8] R. Giganto and T. Smith, "Derivation of Classes from Use Cases Automatically Generated by a Three-Level Sentence Processing Algorithm," in Systems, 2008. ICONS 08. Third International Conference on, 2008, pp. 75–80.

[9] S. Mustafiz, J. Kienzle, and H. Vangheluwe, "Model transformation of dependability-focused requirements models," in Modeling in Software Engineering, 2009. MISE '09. ICSE Workshop on, 2009, pp. 50–55.

[10] D. K. Deeptimahanti and R. Sanyal, "Semi-automaticgenerationofUMLmodelsfrom natural language requirements," in Proceedings of the 4thIndiaSoftwareEngineeringConference, ser. ISEC '11, 2011, pp.165–174. [Online]. http://doi.acm.org/10.1145/1953355.1953378

[11] Unified Modeling Language: Superstructure, version 2.2, formal/09-02-02, Object Management Group, 2009.

[12] "AndroMDA project home page," http:// andromda.org/.

[13] "MDA website," http://omg.org/mda/.

[14] Y. Wang and M. Wu, "Case studies on translation of RTPA specifications into Java programs," in Canadian Conference on Electrical and Computer Engineering, Vol. 2, 2002, pp. 675–680.

[15] J. Falb, S. Kavaldjian, R. Popp, D. Raneburger, E. Arnautovic, and H. Kaindl, "Fully Automatic User Interface Generation from Discourse Models," in Proceedings of the 14th International Conference on Intelligent User Interfaces, ser. IUI '09. New York,NY, USA: ACM, 2009, pp. 475–476. [Online]. http://doi.acm.org/10.1145/1502650.1502722

[16] C. Janssen, A. Weisbecker, and J. Ziegler, "Generating User Interfaces from Data Models and Dialogue Net Specifications," in Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, ser. CHI '93. New York, NY, USA: ACM, 1993, pp. 418–423. [Online]. http://doi.acm.org/10.1145/169059.169335

[17] M. ElKoutbi, I. Khriss, and R. Keller, "Generating user interface prototypes from scenarios," in Requirements Engineering, 1999. Proceedings. IEEE International Symposium on, 1999, pp. 150–158.

[18] Jacobson, I., M. Christerson, et al. (1992). Object-Oriented Software Engineering: A Use

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Case Driven Approach. Wokingham, England, Addison-Wesley.

[19] Rumbaugh, J., I. Jacobson, et al. (1999). The Unified Modeling Language Reference Manual. Reading, Addison Wesley.

[20] McMenamin, S.M. and J.F. Palmer, 1984. Essential Systems Analysis. 1st Edn., Yourdon Press, New York, USA., ISBN: 978-0917072307 pp: 408.

[21] Yourdon, E., 1988. Modern Structured Analysis. 1st Edn., Yourdon Press, Englewood Hills, NJ., USA., ISBN: 978-0135986240, pp: 688.

[22] Page-Jones, M., 1999. Fundamentals of ObjectOriented Design in UML. 2nd Edn., AddisonWesley, Boston, MA., USA., ISBN: 978- 0201699463, pp: 480.

[23] Stumpf, R. and L. Teague, 2005. Teachings objectoriented system analysis and design with UML. Proceedings of the Information Systems Education Conference (ISECON'05), October 6-9, Columbus, OH, USA., pp: 1-14. http://www.csupomona.edu/~rvstumpf/isecon/tea c hing_OO.ppt

[24] Cockburn, A., 2000. Writing Effective Use Cases. 1st Edn., Addison-esley, Boston, MA., USA., ISBN: 978-0201702255, pp: 304.

[25] Inayat, I., Moraes, L., Daneva, M. and Salim, S.S. (2015) A Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed. Scientific Workshop Proceedings of the XP2015, Trondheim, p. 6. http://dx.doi.org/10.1145/2764979.2764985.

[26] Papadopoulos, G. (2015) Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. Procedia—Social and Behavioral Sciences, 175, 455-463. http://dx.doi.org/10.1016/j.sbspro.2015.01.1223

[27] W. Hussain and T. Clear, "Spreadsheets as collaborative technologies in global requirements change management," in Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on, 2014, pp. 74–83.