ISSN 2348 – 9928
IJAICT Volume 1, Issue 11, March 2015 Doi:01.0401/ijaict.2015.11.16 Published on 05 (04) 2015

PROBLEM SOLVING USING CASE BASED REASONING METHODOLOGY

Prof. Swati Shekapure
MCOE, Karve Nagar, Pune, India

Prof. Vilas Thakare
Amravati, Maharastra, India

Abstract*— *Case-based reasoning (CBR) is a methodology for solving problems. These problems may be of a variety of natures. In principle, no problem type is excluded from being solved with the CBR methodology. The problem types range from exact sciences to mundane tasks. Case-based reasoning (CBR) is an approach to problem solving that emphasizes the role of prior experience during future problem solving. New problems are solved by reusing and if necessary adapting the solutions to similar problems that were solved in the past. This paper illustrate how case based reasoning approach is used for the solving the problem.

Keywords*— *CBR-Case Based Reasoning, Adaption, Classification, Diagnosis, Configuration, Fuzzy

I. INTRODUCTION

The term case-based reasoning consists of three words and they need a short explanation. A **case** is basically an experience of a solved problem. This can be represented in many different ways. A case base is a collection of such cases. The term **based** means that the reasoning is based on cases, that is, cases are the first source for reasoning. The term most characteristic of the approach is **reasoning**. The kind of reasoning is, however, quite different from reasoning in databases and logic. The most important characteristic that distinguishes case-based reasoning from other kinds of reasoning is that it does not lead from true assumptions to true conclusions. This means that even if the solution in a recorded case were correct for its original problem, this may not be the case for a new problem. This possibility is based on the general fact that the situation in the recorded experience may not be exactly the same as that in the new problem. In fact, to be reused, it only has to be "similar". Therefore, the result of making use (or reuse) of the experience may only be "close" to the correct solution of the new problem. This means that applying CBR is a kind of approximate reasoning. [4]

1.1 Case

A case can be said to be the record of a previous experience or problem. The information recorded about this past experience will, by necessity, depend on the domain of the reasoner and the purpose to which the case will be put. In the instance of a problem solving CBR system, the details will usually include the specification of the problem and the relevant attributes of the environment that are the circumstances of the problem. The other vital part of the case is the solution that was applied in the previous situation. Depending on how the CBR system reasons with cases, this solution may include only the facts of the solution, or, additionally, the steps or processes involved in obtaining the solution.

It is also important to include the achieved measure of success in the case description if the cases in the case base have achieved different degrees of success or failure. When a comparison is made between the knowledge stored in a model/rule based system and that stored in a case base, it is apparent that the information in the case base is of a more specific nature than that of the model/rule based system. While the knowledge in a model/rule based system has been abstracted so that it is applicable in the widest variety of situations as possible, the knowledge contained in a case base remains specific to the case in which it is stored. Because of the specific knowledge of a case base, we find that related knowledge and knowledge applicable in a specific circumstance is stored in close proximity. Thus, rather than drawing knowledge from a wide net, the knowledge needed to solve a specific problem case can be found grouped together in a few, or even one location. The case base in the CBR system is the memory of all previous stored cases. There are three general areas that have to be considered when creating a case base.

- The structure and representation of the cases themselves
- The memory model used for organizing the entire case base
- The selection of indices which are used to identify each case [7][4]

1.2 Experiences

Experiences are essential for CBR. In general, an experience is a recorded episode that occurred in the past, such as "Remember, last year in Italy we had a similar problem with our car. Cases can be quite complex and consist, as mentioned, of whole stories. CBR uses them for solving problems;

therefore, there must be something in the experience that talks about a problem and its solution. In a simple view, CBR divides an experience into two parts:

(a) A problem part (or a description of a problem situation).

(b) A solution part that describes how one has reacted.

Often one restricts CBR to solutions that have been successful, but that is by no means necessary or adequate. A failed solution is also an important piece of information that states what one has to avoid. Positive experiences (cases) implement successful solutions and lead to the advice:
Do it again! Negative experiences (cases) implement failed solutions and lead to the advice:
Avoid this! [8]

Major types of experiences occur in:
(a). Classification: Decide the class to which an object belongs. For instance, classify mushrooms into the two classes "edible" and "poisonous".
(b). Diagnosis: Decide what the diagnosis of a problem is. For instance, determine whether what causes a car to malfunction is lack of gas
(c) Prediction: Decide what happens tomorrow. For instance, for predict expenses for a firm for a given month in a given year.
(d) Planning: Decide on a sequence of actions to reach a given goal. For instance, make travel plans.
(e) Configuration: Decide which elements to include. For instance, decide how to select technical features and components of equipment.

While humans can understand accounts of experiences told in everyday language, computers require some formality. Although natural to humans, the recognition of similarity and the consequent ability to reuse experiences requires an analogy when using a computer. This is a formal system that is intended to represent experiences
so they can be reused.[4]

1.3 Problems and solution
Problems are central to CBR because the main purpose of the methodology is problem solving. The formulation of a problem is sometimes difficult because it refers to the context in which it is stated. So, each problem formulation requires a different kind of solution. For example:

What is the price of this car?
One answer could be: Too expensive for us.
Another answer could be: $252,600.
It is obvious that one has to know the context in which the problem is stated in order to find out which answer is appropriate. In other words, for a precise statement the context has to be included in the problem formulation. Part of the context is often the inherited culture. For instance, what counts more, building a street or a school? Depending on the culture, laws may be different in different areas. Other cultures are provided by different sciences such as medicine, business and engineering; even large companies have developed their own culture. The CBR context has to take this into account because transferring solutions across cultures is problematic. For example, each bank has developed its own policy for giving loans to customers. The same bank may interpret the policy differently in each different country it operates; this becomes apparent during financial crises.

There are two types of problems in the context of the CBR methodology. The problems in the cases recorded as experiences are usually referred to as problems in CBR. The cases in the case base can sometimes be distinguished as candidate cases, as they are candidates for reuse. However, the entire CBR process is triggered by a problem. This is the new problem, or the actual problem that motivates a user to find a problem-solving method.

The possible ways of representing a solution vary:
It can be just a solution in the narrow sense.
It can contain in addition:
– Comments, illustrations, explanations.
– Advice on how to use the solution.
– The effect by describing what occurred with the solution in the past.
– Remarks on the strategy with which the solution was obtained.

In simple cases the solution contains a name or simple data, for instance, an object or an expected temperature. It may also be a project with values given to predefined attributes, such as jogging three times a week for 45 minutes. Solutions may also have a complex object-oriented structure as a technical object. Even more complex are solutions for planning and those in textual or image form. In a complex situation the solution is a decision for performing an action or even a process. Here one has to distinguish the decision from the action; the action refers to an implementation and run of a strategy that may change states of variables. While the decision is usually clearly formulated, the outcome of the action may be uncertain. Suppose, for instance, that we have the choice between the different lotteries $L1, \ldots, Ln$ and we want to choose a lottery that has maximal expected win. Then our solution can only present us a certain lottery; the win is represented as a probability distribution. Hence the computed probability has to be mentioned in the solution description. Another example is if we decide to fly to Toronto. The execution may fail or be postponed because of various

ISSN 2348 – 9928
IJAICT Volume 1, Issue 11, March 2015 Doi:01.0401/ijaict.2015.11.16 Published on 05 (04) 2015

unforeseen events. The latter means that the result of using a solution is uncertain because of unexpected external results like bad weather or an earthquake. If these are likely to happen one should extend the solution by an entry "effect" for describing what really happened. The user who sees the solution does not know this. If it is added then the user may get a hint for some possible adaptation. Finally, there are situations where the usefulness of the solutions can only be judged if they are executed in reality. This is the case with decisions for organizing city traffic, or, more generally, with making predictions.[11][2]

II. CASE BASED REASONING PROCESS

All case-based reasoning methods have in common the following process:

- Retrieve the most similar case (or cases) comparing the case to the library of past cases;
- Reuse the retrieved case to try to solve the current problem;
- Revise and adapt the proposed solution if necessary;
- Retain the final solution as part of a new case.

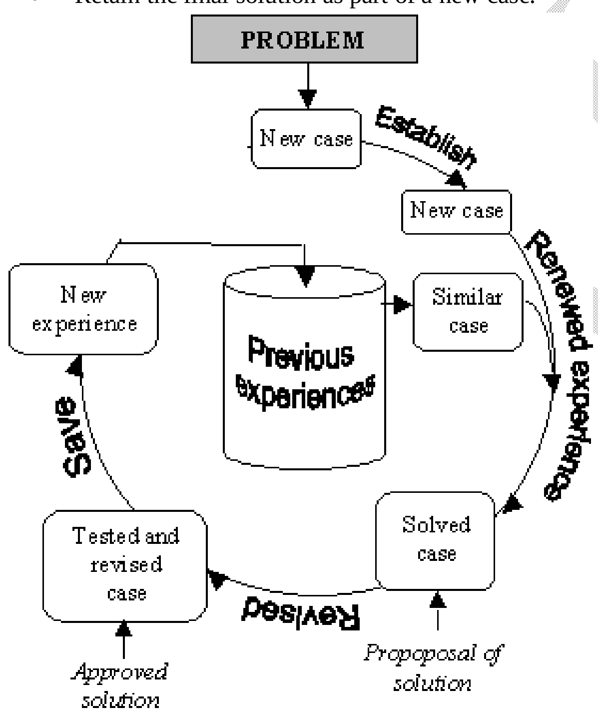

Fig1. Case Based Reasoning process ref. Aamodt & Plaza 1994

There are a variety of different methods for organizing, retrieving, utilizing and indexing the knowledge retained in past cases.

2.1 Retrieve:

Retrieving a case starts with a (possibly partial) problem description and ends when a best matching case has been found. The subtasks involve: 1.identifying a set of relevant problem descriptors; 2.matching the case and returning a set of sufficiently similar cases (given a similarity threshold of some kind); and 3.Selecting the best case from the set of cases returned. Some systems retrieve cases based largely on superficial syntactic similarities among problem descriptors, while advanced systems use semantic similarities.

2.2Reuse:

Reusing the retrieved case solution in the context of the new case focuses on: identifying the differences between the retrieved and the current case; and identifying the part of a retrieved case which can be transferred to the new case. Generally the solution of the retrieved case is transferred to the new case directly as its solution case. Reuse is the step of the process when one case is selected for its solution to be reused. It is completed when the new solution is proposed for the next task of the process revision. Reuse is about proposing a solution for solving the new problem by reusing information and knowledge in the retrieved case(s). Reuse is quite simple when the new problem is identical to the retrieved case problem. When they differ, they require adaptation.

Revise: Revising the case solution generated by the reuse process is necessary when the solution proves incorrect. This provides an opportunity to learn from failure. Revise starts when a solution is proposed to solve the new problem, and it is completed when it is confirmed. Revise aims to evaluate the applicability of the proposed solution. Evaluations can be done in the real world or in a simulation. Simulation is easier and cheaper but may neglect practically important aspects. In the real world, evaluation aspects may be present that one might not have considered in the model. In fact, this is an old phenomenon in Artificial Intelligence called the frame problem. It says that one can never completely formulate all possible facts that may occur in the real world.

2.3 Retain:

Retaining the case is the process of incorporating whatever is useful from the new case into the case library. This involves deciding what information to retain and in what form to retain it; how to index the case for future retrieval; and integrating the new case into the case library. When revising generates a new case, updating the case base with the new (learned) case for future problem solving takes place. Nevertheless, a confirmed solution may or may not be retained. Some systems learn new solutions adapted through use; others accept only actual cases. [2][4][8]

III. CASE BASED REASONING KNOWLEDGE MODEL

The knowledge container view of the CBR methodology is based on the perspective that CBR is a knowledge-based

system. Knowledge-based systems are a class of intelligent systems that are designed by having a knowledge base in an independent module. In CBR, extend this notion to emphasize how the methodology utilizes different kinds of knowledge in distinct repositories: the knowledge containers. CBR process point of view, one may also ask what kind of knowledge is represented and where it can be found. Knowledge can either be represented explicitly or be hidden in an algorithm. In any case, there must be some way to formulate the knowledge; say that knowledge is presented in some formulation. The formulation is stored in what is called a knowledge container. For the knowledge containers described next we state what kind of knowledge could be contained in them.
In CBR four major knowledge containers

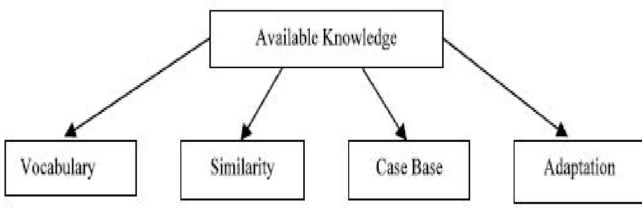

Fig 2: Knowledge in CBR

The knowledge containers represent one view of a CBR system; they are not modules that can perform certain subtasks. They contain certain knowledge units that in combination help solve a problem.

3.1 The Vocabulary Container

The vocabulary is basic for any knowledge-based system. This is not special to CBR. The vocabulary determines what one can discuss explicitly. The vocabulary plays a role in all levels of abstraction, which is illustrated by very simple examples:

1. If we do not know the word heart rate we cannot talk about it. It is knowledge that this term plays a role.

2. If the term tax cost is missing one cannot compute the tax correctly. Again, this is knowledge. This aspect plays a major role in different countries, where different tax regulations are involved.

The vocabulary container retains knowledge about how to explicitly describe the knowledge elements being used. This does not depend on the types of descriptions, ranging from logical constructs to free text. It is a classical observation in science that the solutions of difficult problems have been found only after some person introduced a new crucial notion. Therefore, there is usually much knowledge contained in the chosen vocabulary. For a real-world object there are in principle infinitely many terms that have something to do with the object but only a few are relevant for a specific task. That means an object can (and should) have different description terms for different tasks. In the vocabulary container one can identify various sub-containers that are useful for technical purposes as retrieval, input or output. These are, for example, names of employees, companies, products in a supermarket, and so on. These sub containers are frequently defined and used in application domains.

3.2 The Similarity Container

The knowledge in the similarity container consists of all knowledge needed to determine what makes a case similar to another such that their solutions can be reciprocally reused. There are multiple ways to ensure similarity knowledge accomplishes this: From the use of simple symbolic similarities where the values are either equal or not, through the use of weights to represent relative importance of the attributes, through the use of systems where relevance is computed at runtime, to the use of fuzzy algorithms that consider all attributes and their importance at once. The similarity is used for retrieval purposes. This means that something has to be known about the problem and what is required for the solution.

As an example consider the task of squaring numbers and assume we are unable to multiply and do not want to learn how to do so. Suppose we have a base of solved problems, say $Squ = (2, 4), (2.5, 6.25), (-3, 9), (-5, 25)$

As a special problem we take square (3) =? The answer is not in our list; therefore we have to look for the nearest neighbour of “3”. A first try is to take the Euclidean distance, which gives 2.5, and the answer 6.25. A much better method is to equip the similarity measure with the knowledge square(x) = square $(-x)$ for all x. Then we would retrieve −3 which gives the correct answer. The similarity measure is much easier to use than it is to learn multiplication.

For CBR and retrieval purposes it is important to quantify similarities. This is done by similarity measures, which can be defined as a mapping
$\text{sim} : U \times U \rightarrow [0, 1]$
where U contains the objects to be compared.

Not all aspects of a problem situation may be of equal importance. For example, the price of a car may be more important than the color. If the similarity knows this then it would pay more attention to the price attribute than to the color attribute. A way to make this possible is to assign weights to attributes. Let us have an example dealing with car repairs where the similarity measure was naively chosen but successful. It ranked the cases and we selected the most similar one because similarity tends to be an adequate proxy for utility.

3.3 The Case Base Container

The case base container contains experiences as cases. These experiences may be available from the past or may be constructed from variations of existing cases, or be completely artificial. The description of the case base as a knowledge container is straightforward as the case base is typically the main source of knowledge in CBR systems.

3.4 The Adaptation Container

The knowledge in the adaptation container will be used to adapt cases to solve new problems. The most common formalisms adopted for adaptation are rule bases; nevertheless, case bases can be used, and even existing cases from the case base have been used at runtime to extract adaptation knowledge. The knowledge in the adaptation container can be used to transform an existing solution or generate a new solution based on a strategy from a previous solution.

In the adaptation container one finds information on how to modify a solution. In the adaptation container rules are stored for adapting a retrieved solution to a new situation. Such rules are intended to perform a solution transformation that has to take care of the fact that the solutions obtained from the case base using the nearest neighbour principle may still be insufficient (either because of a not very well defined similarity measure or simply because the case base does not contain a better solution). In this situation the solution is adapted. Adaptation knowledge can

Drastically reduce the number of cases needed in the case base. [4][10][11]

IV. WHY USE CBR

Case-based reasoning (CBR) is an approach to problem solving that emphasizes the role of prior experience during future problem solving. Case-based reasoning (CBR) is a methodology for solving problems

1. Reduction of the Knowledge Acquisition Task. By eliminating the extraction of a model or a set of rules as is necessary in model/rule based systems, the knowledge acquisition tasks consists mainly of the collection of the relevant existing experiences/cases and their representation and storage.

2. Avoid repeating mistakes made in the past. In systems that record failures as well as successes, and perhaps the reason for those failures, the system can use the information about what caused failures in the past to predict any failures in future. An example of such a system could be one which stores successful or failed lessons.

3. Graceful degradation of performance. Some model based systems cannot even attempt to solve a problem on the boundaries of its knowledge or scope, or when there is missing or incomplete data. In contrast case-based systems can often have a reasonably successful attempt at solving these types of problem.

4. Able to reason in domains that have not been fully understood defined or modelled. While insufficient knowledge may exist about a domain to build a causal model of it or derive a set of heuristics for it, a case-based reasoned can function with only a set of cases from the domain. The underlying theory does not have to be quantified

5. May be able to make predictions as to the probable success of a proffered solution. Where information is stored regarding the level of success of past solutions, the reasoner may be able to predict the success of the suggested solution to a current problem. This may be done by referring both to the stored solutions and to the differences between the previous and current contexts of the solution.

6. Learn over time. As CBR systems are used, they encounter more situations and create more solutions. If cases are tested in the real world and a level of success determined, these cases can be added into the case base to reason with in future. As we add cases, a CBR system should be able to reason in a wider variety of situations, and with a higher degree of refinement/success.

7. Reason in a domain with a small body of knowledge. While a domain in which there is little known underlying knowledge and few cases from which to start limits the type of reasoning that can be done in it, a case based reasoner can start with the few known cases and incrementally increase its knowledge as cases are added to it. The addition of these cases will also cause the system to grow in the directions encountered by the system in its problem solving endeavours.

8.Reason with incomplete or imprecise data and concepts As cases are retrieved not just when identical to the current query case but when they are within some measure of similarity, incompleteness and imprecision can be dealt with. While these factors may cause a slight degradation in performance due to the current and retrieved having increased disparity, reasoning can still continue.

9. Avoid repeating all the steps that need to be taken to arrive at a solution. In problem domains that require significant processes to carry out the creation of a solution from scratch,

the modifying of a previous solution can significantly reduce this processing.

10. By reusing a previous solution, the steps taken to reach the retrieved solution can be reused themselves. Provide a means of explanation Case-based reasoning can supply a previous case and its (successful) solution to convince a user, or justify to a user, a solution it is providing to their current problem. In most domains, there will be times when a user wishes to be reassured about the quality of the solution they are being given.

11. By explaining how a previous case was successful in a situation, using the similarities between the cases and the reasoning involved in adaptation a CBR system can explain its solution to a user. Even in a hybrid system that may use multiple methods to find a solution, this explanation mechanism can augment the causal (or other) explanation given to the user.

12. Can be applied to a broad range of domains. As will be discussed in the section on application areas, CBR has many areas of application. Due to the seemingly limitless number of ways of representing, indexing, retrieving and adapting cases, CBR can be applied to extremely diverse application domains.

13. Reflects human reasoning. As there are many situations where we, as humans, use a form of case based reasoning, it is not difficult to convince implementers, users and managers of the validity of the paradigm. Likewise, humans can understand a CBR system's reasoning and explanations and are able to be convinced of the validity of the solutions they are receiving. If the human user is wary of the validity of the received solution, they are less likely to use the solution given to them by the reasoner. The more critical the domain, the lower the chances of use, and the higher the level of the user's understanding and credulity will need to be.[1][9][10]

V. APPLICATIONS

Case based reasoning first appeared in commercial tools in the early 1990's and since then has been sued to create numerous applications in a wide range of domains:

1. Diagnosis: case-based diagnosis systems try to retrieve past cases whose symptom lists are similar in nature to that of the new case and suggest diagnoses based on the best matching retrieved cases. The majority of installed systems are of this type and there are many medical CBR diagnostic systems.

2. Help Desk: case-based diagnostic systems are used in the customer service area dealing with handling problems with a product or service.

3. Assessment: case-based systems are used to determine values for variables by comparing it to the known value of something similar. Assessment tasks are quite common in the finance and marketing domains.

4. Decision support: in decision making, when faced with a complex problem, people often look for analogous problems for possible solutions. CBR systems have been developed to support in this problem retrieval process (often at the level of document retrieval) to find relevant similar problems. CBR is particularly good at querying structured, modular and non-homogeneous documents.

5. Design: Systems to support human designers in architectural and industrial design have been developed. These systems assist the user in only one part of the design process, that of retrieving past cases, and would need to be combined with other forms of reasoning to support the full design process. [2][8][9]

5.1 Suitability

Some of the characteristics of a domain that indicate that a CBR approach might be suitable include:

1 .Records of previously solved problems exist;
2. Historical cases are viewed as an asset which ought to be preserved;
3. Remembering previous experiences is useful;
4. Specialists talk about their domain by giving examples;
5. Experience is at least as valuable as textbook knowledge.
6. Case-based reasoning is often used where experts find it hard to articulate their thought processes when solving problems. This is because knowledge acquisition for a classical KBS would be extremely difficult in such domains, and is likely to produce incomplete or inaccurate results. When using case-based reasoning, the need for knowledge acquisition can be limited to establishing how to characterize cases.[9]
7. Case-based reasoning allows the case-base to be developed incrementally, while maintenance of the case library is relatively easy and can be carried out by domain experts.

VI. CONCLUSION

Using CBR we can be able to make predictions as to the probable success of a volunteer solution. Where information is stored regarding the level of success of past solutions, the reasoner may be able to predict the success of the suggested solution to a current problem. This may be done by referring both to the stored solutions and to the differences between the previous and current contexts of the solution. Systems to support human designers in architectural and industrial design have been developed. These systems assist the user in only one part of the design process, that of retrieving past cases, and would need to be combined with other forms of reasoning to support the full design process.

References

[1] URL http://www.aiai.ed.ac.uk/links/cbr.html

[2] URL http://www.ai-cbr.org

[3] URL http://www.cbr-web.org

[4] Michael M. Richter , Rosina O. Weber, " Case-Based Reasoning" A Textbook

[5] Danielle Monfet, Maria Corsi, Daniel Choinière, Elena Arkhipova, "Development of an energy prediction tool for commercial buildings using case-based reasoning", Elsevier, Energy and Buildings vol no,81 ,pp.152–160,2014.

[6] Julien Henriet , Pierre-Emmanuel Leni a, Rémy Laurent a , Michel Salomon, "Case-Based Reasoning adaptation of numerical representations of human organs by interpolation", Elsevier, Expert Systems with Applications 41 ,pp 260–266,2014.

[7] Martin, A. ; Research Scholar Department of Banking Technology Pondicherry University, Puducherry India ; Miranda Lakshmi, T. ; Venkatesan, V.Prasanna , "A business intelligence framework for business performance using data mining techniques" ,IEEE, Emerging Trends in Science, Engineering and Technology (INCOSET),13-14 Dec. 2012

[8] Zouhair, A. , Univ. of Abdelmalek Essaadi, Tangier, Morocco ; En-Naimi, E.-M. ; Amami, B. ; Boukachour, H. , "Intelligent tutoring systems founded of incremental dynamic case based reasoning and multi-agent systems",IEEE, Advanced Logistics and Transport ,pp.29-31,May 2013.

[9] Gouttaya, N. ; Fac. of Sci. & Technol., Modeling & Sci. Comput. Lab., Sidi Mohamed Ben Abdellah Univ., Fez, Morocco ; Begdouri, A. , "Integrating data mining with case based reasoning (CBR) to improve the proactivity of pervasive applications", IEEE,Information Science and Technology (CIST), 22-24 Oct. 2012

[10] Gatzioura, Anna ; Sanchez-Marre, Miquel, "A Case-Based Recommendation Approach for Market Basket Data" ,Intelligent System,IEEE,V.30,Issue 1,pp20-27,2015

[11] Bin Linghu ; Feng Chen,"An Intelligent Multi-agent Approach for Flood Disaster Forecasting Utilizing Case Based Reasoning Intelligent System Design and Engineering Application(ISDEA),Fifth international conference,pp.182-185,2014

[12] Goknil, A. ; van Domburg, R. ; Kurtev, I. ; van den Berg, K. ;Wijnhoven, F. , " Experimental evaluation of a tool for change impact prediction in requirements models: Design, results, and lessons learned", Model driven requirements Engineering workshop, fourth international conference,pp.57-66,2014