

# Approach of Case Base Reasoning in Handling the Unavailable Information in Project Handover of Marketing Research Domain Tools Using Language- Based on Real Integrated Price Format

Anshula<sup>1</sup>, Krishan Kumar<sup>2</sup>

Assistant Professor, Computer Science Department, Kalindi College , University of Delhi

**Abstract:** This research attempt aims to use a Case-Based Reasoning (CBR) approach for the unavailable information in a project handover of marketing research domain tools using language based on real integrated price format. Marketing research tools basically divided into the different parts language Scripting Editor, language debugger, Web Interface is an application for writing script for collecting the information for Marketing research tools. The collect the information from outer world by these process i) Crawl the Web Pages from the different -2 sites with the help of URLs ii) Generation of URLs iii) Extract the information from the web Pages iii) Creating the Final Report of the Extracted information iv) Showing the All Data on the GUI interface , But some time due to any reason at any moments information is not available on particular site or blocking of urletc, any type of problem effects the tools, and this missing information create the problem at the time of project handover from one team to another team in a service or operation companies, even this missing part some time skip due to that time url is working but in time phase of handover it is blocked or change in format but to handle the such type of problem it used CBR.

**Keywords:** URL-Uniform Resource Locator, CBR-Case Base Reasoning, Crawling, RIPL- Real Integrated Price Language

## I. INTRODUCTION OF CASE-BASE REASONING

CBR is a hurriedly growth research area in cognitive knowledge. It is a new approach for representing knowledge and using that knowledge to help user solve problems[1]. It's basic idea is that "a case-based reasoned solves new problems in handover missing information using CBR agents. Through the techniques of CBR, the experience can be captured and organized as a set of historical cases[2], cumulated in a case base, used to help problem solving or suggestions providing by recalling similar cases. The stored case base is similar to a database system with some particular features (fields), but it is more than a database system because it does its retrieval based on the specifics of a situation and finds partially matching cases that can be used to answer the specific question of the user. Moreover, it does not require the full matching of features, nor does it require a database administrator to formulate queries[3][4]. Thus, a CBR approach is quite suitable for the domains that are experience -rich and probably are hard to define features.

## II. INTRODUCTION OF REAL INTEGRATED PRICE LANGUAGE

Real Integrated Price Language is divided into the different parts RIPL is a scripting language like assembly. which is

used to write the scripts for some specific needs. Generation of Dynamic URLs.

1. Crawling.
2. Extraction.
3. Report Generation.

But it is further subdivided in more parts. The collect the information like Hotel Name, Check-date, Checkout-date, Room Rate and Room Description with the help of this application i) Crawl the Web Pages from the different -2 sites with the help of URLs ii) Generation of URLs iii) Extract the information from the web Pages iii) Creating the Final Report of the Extracted information iv) Showing the All Data on the GUI interface. For the above purposes write the scripts. Like C,C++, JAVA in RIPL for writing the scripts will use some Data type

## III. COMMANDS IN REAL INTEGRATED PRICE LANGUAGE

Some Data type used in RIPL[6][8].

int, string, mov, writestring, readline, writetech, writenewline, readpage, if=0, if<0, if>0, tostring, toint, print, goto, crawl, readpage, sethead, gethead, movehead, scanfwd, gettext, broadcast, exit.

a) **Int** :-This instruction is used for declaring a integer variable.

Syntax: Instruction Address int Valid RIPL variable name

**b) Mov :-**This instruction is used to assign a value to a given variable.

Syntax: Instruction Address mov Valid RIPL variable expression evaluating to integer

**c) Writestring :-**This instruction is used for copying string into HEAP. Primarily, could be used for copying string constants into HEAP and can also be used to make copy of string variables, in HEAP.

Syntax: Instruction Address writestring String\_Expression, Integer\_Expression

**d) readline :-**This instruction is used to read the specified Line into the RIPL interpreter file databuffer. This caching of data results in reduced disk access and faster processing of file data. Its not necessary to read the whole file, but offset within the file can be specified from which to read the file data. This turns out useful in reducing memory usage when we are sure data the info that we need from page is present after say 'x' bytes, then we can read the file starting from offset 'x'.

Syntax: Instruction Address readline String\_expression Integer\_Expression, Integer\_Expression

**e) Writtech :-**This instruction is used for accessing the HEAP at character level, to write value into it. This particularly turns out useful for string manipulation.

Syntax: Instruction Address writtech Integer\_Expression1 Integer\_Expression2

where Integer\_Expression1 is IndexInHeap and Integer\_Expression2 is value to be written.

**f) writenewline :-**This instruction is used to writing the specified Line into the a file interpreter databuffer. This caching of data results in reduced disk access and faster processing of file data. Its not necessary to read the whole file, but offset within the file can be specified from which to write the file data. This turns out useful in reducing memory usage when we are sure data the info that we need from page is present after say 'x' bytes, then we can read the file starting from offset 'x'.

Syntax: Instruction Address writenewline String\_expression, String\_expression

**g) Readpage :-**This instruction is used to read the specified file data into the RIPL interpreter file data buffer. This caching of data results in reduced disk access and faster processing of file data. Its not necessary to read the whole file, but offset within the file can be specified from which to read the file data. This turns out useful in reducing memory usage when we are sure data the info that we need from page is present after say 'x' bytes, then we can read the file starting from offset 'x'.

Syntax: Instruction Address readpage String\_expression Integer\_Expression

**h) If=0 :-**This instruction helps in performing conditional branching, if the first argument of this instruction evaluates to zero.

Syntax: Instruction Address if=0 expression1 jump expression2

**i) if>0 :-**This instruction helps in performing conditional branching, if the first argument of this instruction evaluates to greater than zero.

Syntax: Instruction Address if>0 expression1 jump expression2

**j) If<0 :-**This instruction helps in performing conditional branching, if the first argument of this instruction evaluates to less than zero.

Syntax: Instruction Address if<0 expression1 jump expression2

**k) tostring :-**This instruction helps in performing the type conversion. tostring can convert the value into string.

Syntax: Instruction Address tostring Integer\_Expression1 Integer\_Expression2

**l) tono :-**This instruction helps in performing the type conversion. toint can convert the value into Integer.

Syntax: Instruction Address toint Integer\_Expression1 Integer\_Expression2

**m) print :-**This instruction is currently used for monitoring purpose and will be trimmed in actual integration with EBW. It can be supplied string/integer expression and the same will be evaluated and written to a file named "xsaction.log".

Syntax: Instruction Address print string/integer expression

**n) goto :-**This instruction is used for performing unconditional branching.

Syntax: Instruction Address goto expression evaluating to integer

**o) crawl :-**This instruction is used for crawl the webpages corresponding to a particular URL.

Syntax: Instruction Address crawl pagename cookie URL proxy pagesize page state

**p) Readpage :-**This instruction is used to read the specified file data into the RIPL interpreter file data buffer. This caching of data results in reduced disk access and faster processing of file data. Its not necessary to read the whole file, but offset within the file can be specified from which to read the file data. This turns out useful in reducing memory usage when we are sure data the info that we need from page is present after say 'x' bytes, then we can read the file starting from offset 'x'.

Syntax: Instruction Address readpage String\_expression Integer\_Expression

**q) SetHead :-**This is instruction is similar in behavior to set the Cursor of currently used RIs. This instruction moves the file pointer by specified value.

Syntax: Instruction Address SetHead Integer\_Expression

**r) movehead :-**This is instruction is similar in behavior to Move Cursor of currently used RIs. This instruction moves the file pointer by specified value.

Syntax: Instruction Address movehead Integer\_Expression

**s) scanfwd :-**This is instruction is to take the file pointer to the location in file where a given search string exists.

Syntax: Instruction Address scanfwd String Expression Integer Variable

**t) gettext :-**This instruction starts copying from current file pointer position into HEAP at a specified address until the terminating string string expression is found.

Syntax: Instruction Address  
 getString\_ExpressionInteger\_ExpressionIntegerVariable  
**u) exit** :-This instruction is to terminate the program execution. As soon as this instruction is encountered, the program execution terminates.

Syntax: Instruction Address exit

**v) Broadcast** :-This instruction is used for printing the result which is on the Heap location.

Syntax: Instruction Address broadcast Heap Index

#### IV. CAUSE OF UNAVAILABILITY OF INFORMATION IN RIPL

There is various cause of non-availability of information but the main reasons are

- Non availability or logical error in URL calculation and collections  
 Some time in RIPL the is logical error occurred and it provide un-relevant data and sometime URL is not calculated or not available of any new demands or previous one
- Blocking of URL :- some time due to the abundant amount of hitting of a url for particular data ,the url's blocked for the accessing IP or Mac id, in this case we also loss the information
- Sometime page which is crawl down , we did not get any relevant data on the particular crawled page (may be site under construction , or some error problem of particular site etc)

#### V. CBR APPROACH IN HANDLING THE UNAVAILABLE INFORMATION MARKETING RESEARCH DOMAIN TOOLS FORMAT IN RIPL

CBR permit the precedent solutions to be compiled in a reusable manner. If a previous solution's conditions of applicability can be abstracted and indexed then the CBR system can re-use these solutions. This ultimately saves computational time and allows the problem solver to avoid past mistakes. If the conditions that caused a previous failure can be abstracted and indexed then the user can be saved the futility of repeating mistakes. Human knowledge acquisition often involves the use of experiences and cases. Approach in handling the unavailable information marketing research domain tools format in inripl.

For this CBR Agent [5] form which deal with the CBR solutions[7] for a particular case and for designing the case of missing or unavailable information the Identifier Agent is design, Correction agent take the solution from the CBR agent and update the information in marketing research tools format in RIPL (Real Integrated Price Language). The three tire architecture shows all processing and data flow information in this process of handling the unavailable information[8].

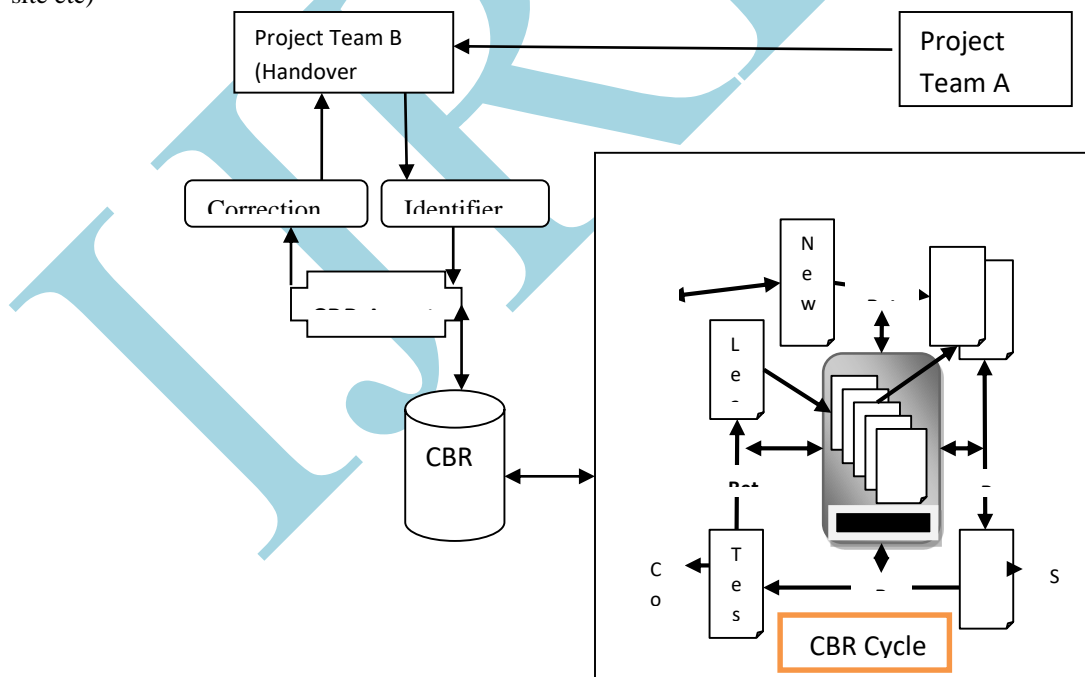


Fig.1.1

At that time CBR pick the previous data from the database and calculating the similarity ratio by nearest neighbour method, but here in this work we used centre of gravity (COG) method by which we calculate the similar case for the unavailable

information and by COG method, take all the previous data from the data base and calculate the probable information by CBR method and put it on interface.

## VI. CONCLUSIONS

The Various problem which occur in project handover which can be handle with the help of CBR. a Case-Based Reasoning (CBR) approach for the unavailable information in a project handover of marketing research domain tools using language based on real integrated price format. Any type of problem effects the tools, and this missing information create the problem at the time of project handover from one team to another team in a service or operation companies, even this missing part some time skip due to that time url is working but in time phase of handover it is blocked or change in format but to handle the such type of problem it used CBR.

## REFERENCES

- [1]. Aamodt, A. (1991). A knowledge-intensive, integrated approach to problem solving and sustained learning. PhD thesis, University of Trondheim, Norwegian Institute of Technology, Department of Computer Science. University Microfilms PUB 92- 08460.
- [2]. Cohen, P. R. (1989). Evaluation and case-based reasoning. In Proceedings of a Workshop on Case-Based Reasoning, pages 168–172, Pensacola Beach, Florida, USA.
- [3]. Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: A critical review. In Endsley, M. R. and Garland, D. J., editors, Situation Awareness Analysis and Measurement: Analysis and Measurement, pages 3–32.
- [4]. Matskin, M., Kirkeluten, O. J., Krossness, S. B., and Sæle, Ø. (2001). Agora: An infrastructure for cooperative work support in multi-agent systems. In Wagner, T. and Rana, O., editors, Infrastructure for Agents, Multi-Agents, and Scalable Multi-Agent Systems, volume 1887 of Lecture Notes in Computer Science, pages 28–40. Springer Verlag.
- [5]. Myrhaug, H. I. (2001). Towards life long and personal context spaces. In Workshop on User Modelling for Context-Aware Applications. Available at: <http://orgwis.gmd.de/gross/um2001ws/papers/myrhaug.pdf>.
- [6]. Deepthi Sehwat (2014). Approach of Real Integrated Price language in Automation of Marketing Research Tool -Resolving Issue with Backup of CBR, in Deepthi Sehwat et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 1, June 2014, pp. 78-81.
- [7]. Nwana, H. S. (1996). Software agents: An overview. Knowledge Engineering Review, 11(3):1–40.
- [8]. Vivek Jaglan et al (2013). Approach of Case Base Reasoning in Handling the Unavailable information Based on Real Integrated Price Language in a Marketing Research Tool, in International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 3.