ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Approach of Case Base Reasoning in Handling the Unavailable information Based on Real Integrated Price Language in a Marketing Research Tool

Manoj[1], Rinkal[2], Vivek[3],Mohit [4]

Assistant Professor,Dept. of CSE,Green Hills Engg. College, H.P, India[1]

Assistant Professor, Dept. of CSE,R.N College of Engineering& Technology, Haryana, India[2]

Faculty, Dept. of CSE, Amity Univesity, Haryana, India[3]

Student, Dept. of CSE,CDLU, Haryana, India[4]

ABSTRACT: This research effort aims to use a Case-Based Reasoning (CBR) approach for the unavailable information in a marketing research tool language based on real integrated price and attempts to investigate its advantages over traditional expert systems approach. Marketing research tool 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 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 , 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 , but to handle the such type of problem it used CBR

Keywords: URL-Uniform Resource Locator, CBR-Case Base Reasoning, Crawling.

I.I INTRODUCTION OF CASE-BASE REASONING

CBR is a rapidly growth research area in cognitive science and artificial intelligence. It is a new approach for representing knowledge and using that knowledge to help user solve problems. It's basic idea is that "a case-based reasoned solves new problems by adapting solutions that were used to solve old problems". Through the techniques of CBR, the experience can be captured and organized as a set of historical cases, 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 adatabase administrator to formulate queries. Thus, a CBR approach is quite suitable for the domains that are experience -rich and probably are hard to define features. Representation more flexible, and it also allows for storing cases and solutions for later on retrievals

I.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 somespecific 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

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

II. COMMANDS IN REAL INTEGRATED PRICE LANGUAGE

Some Data type used in RIPL

int, string, mov, writestring, readline, writech, 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 forcopying string constants into HEAP and can also be used to make copy of stringvariables, in HEAP.

Syntax: Instruction Address writestringString_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 filedata.Its not necessary to read the whole file, but offset within the file can be specifiedfrom which to read the file data. This turns out useful in reducing memory usagewhen 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 readlineString_expressionInteger_Expression, Integer_Expression

e) Writech :-

This instruction is used for accessing the HEAP at character level, to write value intoit. This particularly turns out useful for string manipulation.

Syntax: Instruction Address writech Integer_Expression1 Integer_Expression2

where Integer_Expression1 is IndexInHeap and Integer_Expression2 is value to bewritten.

f) writenewline :-

This instruction is used to writing the specified Line into the a file interpreter file databuffer. This caching of data results in reduced disk access and faster processing of filedata.Its not necessary to read the whole file, but offset within the file can be specifiedfrom 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 writenewlineString_expression ,String_expression

g) Readpage :-

This instruction is used to read the specified file data into the RIPL interpreter filedata buffer. This caching of data results in reduced disk access and faster processingof file data.Its not necessary to read the whole file, but offset within the file can bespecified 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 readpageString_expressionInteger_Expression

h) If=0 :-

This instruction helps in performing conditional branching, if the first argument ofthis 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 ofthis 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 ofthis 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 thevalue into string.

Syntex: Instruction Address tostring Integer_Expression1 Integer_Expression2

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

l) tono :-
This instruction helps in performing the type conversion. toint can convert the valueinto Integer.
Syntex: Instruction Address toint Integer_Expression1 Integer_Expression2
m) print :-
This instruction is currently used for monitoring purpose and will be trimmed inactual integration with EBW. It can be supplied string/integer expression and thesame 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 pagenamecoockie URL proxy pagesize
pagestate
p) Readpage : -
This instruction is used to read the specified file data into the RIPL interpreter filedata buffer. This caching of data results in reduced disk access and faster processingof file data.Its not necessary to read the whole file, but offset within the file can bespecified 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 readpageString_expressionInteger_Expression
q) SetHead :-
This is instruction is similar in behavior to set the Cursor of currently used RIs. Thisinstruction moves the file pointer by specified value.
Syntax: Instruction Address SetHeadInteger_Expression
r) movehead:-
This is instruction is similar in behavior to Move Cursor of currently used RIs. Thisinstruction moves the file pointer by specified value.
Syntax: Instruction Address moveheadInteger_Expression
s) scanfwd :-
This is instruction is to take the file pointer to the location in file where a given searchstring exists.
Syntax: Instruction Address scanfwd String Expression IntegerVariable
t) gettext :-
This instruction starts copying from current file pointer position into HEAP at aspecified address until the terminating string string expression is found.
Syntax: Instruction Address gettextString_ExpressionInteger_ExpressionIntegerVariable
u) exit :-
This instruction is to terminate the program execution. As soon as this instruction isencountered, the program execution terminates.
Syntax: Instruction Address exit
v) Broadcast :-
This instruction is used for printing the result which is on the Heap location.
Syntex: Instruction Address broadcast Heap Index

III. CAUSE OF UNAVAILABILITY OF INFORMATION IN RIPL

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

a) 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
b) 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
c) 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)

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

IV. CBR APPROACH IN HANDLING THE UNAVAILABLE INFORMATION IN RIPL

CBR allows past 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. CBR therefore, offers a manner to model the way human's reason, making it easier to extract and store the expert's knowledge. In RIPL, CBR notes the errors which are continuous in motion or regular or frequently occur

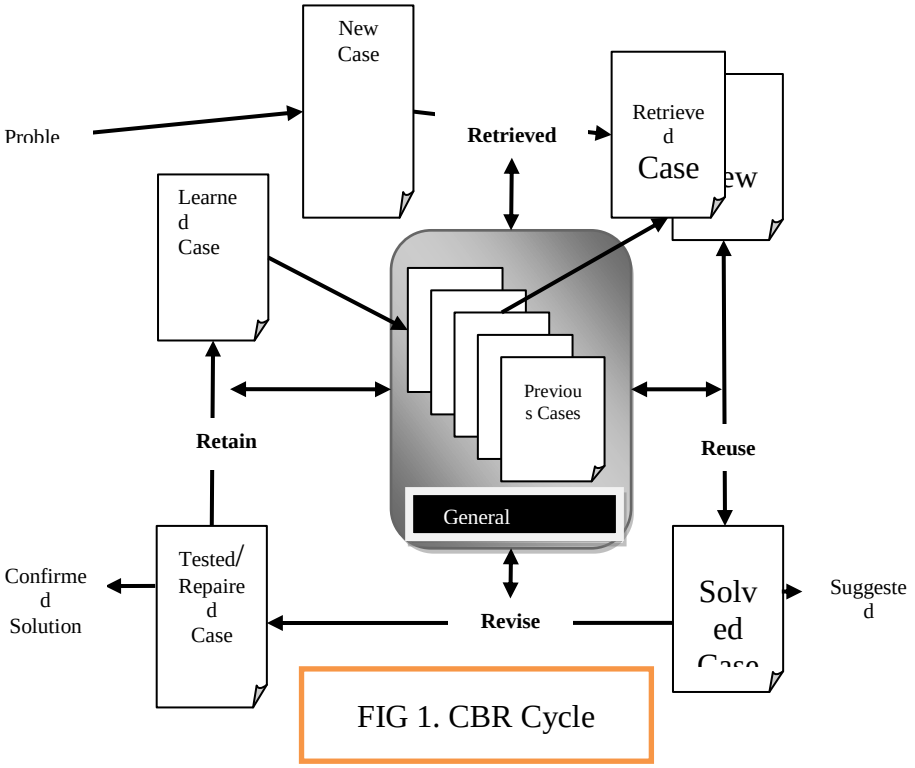

FIG 1. CBR Cycle

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 weight (COW) method by which we calculate the similar case for the unavailable information and by COW method we take all the previous data from the data base and calculate the probable information by CBR method and put it on interface by which a user cannot identify the unviability of information and by chance user pick the information which we provide by CBR method it is find 92% accurate.

V. CONCLUSIONS

The Various problem which occur in RIPL (Marketing Research Tool) can be handle by the help of CBR. In RIPL, CBR notes the errors which are continuous in motion or regular or frequently occur. At that time CBR pick the previous data from the database and calculating the similarity ratio by nearest neighbour method, and put it on interface by which a user cannot identify the unviability of information and by chance user pick the information which we provide by CBR method it is find 92% accurate

REFERENCES

[1]. Aamodt, A. (1991). A knowledge-intensive, integrated approach to problem solving and sustained learning. PhD thesis, University of Trondheim, Norwegian Instituteof Technology, Department of Computer Science. University Microfilms PUB 92- 08460.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

[2]. Aamodt, A. (1994a). Explanation-driven case-based reasoning. In Wess, S., Althoff, K., and Richter, M., editors, Topics in Case-Based Reasoning, pages 274– 288. Springer Verlag.

[3]. Aamodt, A. (1994b). A knowledge representation system for integration of general and case-specific knowledge. In Proceedings from IEEE TAI-94, International Conference on Tools with Artificial Intelligence, pages 836–839, New Orleans, USA. IEEE Confence Proceeding.

[4]. Aamodt, A. (2004). Knowledge-intensive case-based reasoning in Creek. In Funk, P. and Calero, P. A. G., editors, Advances in case-based reasoning, 7th European Conference, ECCBR 2004, Proceedings, pages 1–15. Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1):39– 59. Abowd, G. D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. Pervasive Computing, 38(4):508–530.

[5]. Abowd, G. D., Brumitt, B., and Shafer, S., editors (2001). Ubicomp 2001: UbiquitousComputing, International Conference, Proceedings, volume 2201 of Lecture Notesin Computer Science, Atlanta Georgia, USA. Springer Verlag.

[6]. Aha, D.W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms.Machine Learning, 6:37–66. 154 Bibliography

[7]. Bellifemine, F., Poggi, A., and Rimassa, G. (1999). Jade – a FIPA-compliant agentframework. Technical report, Centro Studi e LaboratoriTelecomunicazioni. Part of this report has been also published in Proceedings of PAAM'99, London, April 1999, pp. 97-108.

[8]. B´enard, R., Bossard, C., and Loor, P. D. (2006). Context's modelling for participativesimulation. In Proceedings of the 19th FLAIRS Conference, pages 613–618. AAAI Press.

[9]. Cohen, P. R. (1989). Evaluation and case-based reasoning. In Proceedings of a Workshopon Case-Based Reasoning, pages 168–172, Pensacola Beach, Florida, USA.

[10]. Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: A criticalreview. In Endsley, M. R. and Garland, D. J., editors, Situation Awareness Analysisand Measurement: Analysis and Measurement, pages 3–32.

[11]. Grimnes, M. and Aamodt, A. (1996). A two layer case-based reasoning architecturefor medical image understanding. In Smith, I. and Faltings, B., editors, Advancesin case-based reasoning, Proceedings of EWCBR 1996, volume 1168 of LectureNotes in Computer Science, pages 164–178. Springer Verlag.

[12]. Li, Y., Hong, J. I., and Landay, J. A. (2004). Topiary: a tool for prototyping locationenhanced applications. In UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology, pages 217–226, New York, NY,USA. ACM Press.

[13]. Lueg, C. (2002a). Looking under the rug: On context-aware artifacts and sociallyadept technologies. In Proceedings of the Workshop The Philosophy and Design ofSocially Adept Technologies at the ACM SIGCHI Conference on Human Factors inComputing Systems (CHI 2002). National Research Council Canada NRC 44918.

[14]. Matskin, M., Kirkeluten, O. J., Krossness, S. B., and Sæle, Ø. (2001). Agora: Aninfrastructure 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.

[15]. Myrhaug, H. I. (2001). Towards life long and personal context spaces. InWorkshop on User Modelling for Context-Aware Applications. Available at:http://orgwis.gmd.de/ gross/um2001ws/papers/myrhaug.pdf.

[16]. Nardi, B. A., editor (1996). Context and Consciousness. MIT Press, Cambridge, MA.

[17]. Newell, A. (1982). The knowledge level. Artificial Intelligence, 18:87–127.

[18]. Nwana, H. S. (1996). Software agents: An overview. Knowledge Engineering Review,11(3):1–40.

[19]. O¨ ztu¨rk, P. and Aamodt, A. (1998). A context model for knowledge-intensive casebasedreasoning. International Journal of Human Computer Studies, 48:331–355.