A USER-CENTERED APPROACH TO VISUALIZING ALGORITHM BEHAVIOUR FOR REAL-TIME DATABASES

OLUSEGUN FOLORUNSO, ADIO TAOFIKI AKINWALE & LATEEF YUSUF

Department Of Computer Science
University Of Agriculture, Abeokuta. Nigeria

ABSTRACT

Real Time Database Systems (RTDBS) designers are charged with systems that produces results in a timely and consistent fashion. Therefore, the design of this kind of databases needs user requirements to support both data structures and the dynamic behaviours of the database algorithms using visualization technique. In this paper, we present a user-centered visualization based on our understanding of the work RTDBS and the needs of designers derived from the first significant user study of RTDBS. The tool presents RTDBS designers with both 'at a glance' understanding of algorithm behavior, and low-level details. The overall satisfaction mean score was 4.83 (std. dev. = 1.27), with mean satisfaction score for the visualization component being 5.17 (std. dev. = 1.11). Users rated the ease of learning a mean score of 4.42 (std. dev. = 1.44). Also, users self-rated performance was strongly correlated to ease of learning (Pearson, 0.84, $p<0.001$), mirroring the desire for the tool to be easily learnable for success. Also, the user rated the ease of searching and of seeing patterns and anomalies in the data with a mean score of 5.25 (std. dev. = 0.96) and 4.42 (std. dev. = 1.56), respectively. Results from preliminary usability testing of a case study show that users performed better and found easier those RTDBS tasks.

Keywords: *Information visualization, RTDBS, User-Centered design, Usability testing*.

1.0 INTRODUCTION

A Real-Time Database System (RTDBS) which serves as the mainstream of computer operation is aptly described as a system that produces results in a timely and consistent fashion. The application requiring these services are enormous. The RTDBS application areas in the recent time includes process control, reservation, scientific data analysis, accounting, banking, law, medical records and multi-media.The concept of Real-Time Database system is designed to address the challenges and constraints of simultaneous data enforcement or assessment. The first major classical publication for RTDBS was by Abbott and Garcia-Molina, (1988). Sequel to this development, there were contributions from Son (1988), Ramamritham (1993), Ulusoy (1995b), Bestravros et al., (1997), Kuo and Lam (2000), Tesanovic et al(2003) and Idoudi et. al (2009) . They all signalled the fusion between the real-time and databases, their main objective was to track the evolutionary trajectories of RTDBS research.

Despite the complexities of these RTDBS tasks, the majority of the tools used by designers are textually based, when experimenting on the design. However, the user must be an expert in both the domain and activity. The additional tools that can transform the output from these experiments were placed into simple graphs and data summaries. However, there are currently few alternatives for users to seamlessly move back and forth between these high level overviews and the precise details of simulation experiment. But in order to accomplish the complex tasks in RTDBS, designer must do just that. Couple with this demanding change of context between high and low-level details is the fact that few tools support the simultaneous analysis of simulation experiments and changing of parameters to measure/observe the effect when they are perturbed. The prototype information visualization presented here support the analysis of states, where the user can drill down into the details of the designer work.

The study of human factors as regards the visualization of RTDBS should be of major interest and importance to all the stake holders of

Vol. 16 No.2 June, 2010 pp [135 – 144]

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

the system, especially RTDBS users and RTDBS visualisation developers who are competing in a very lucrative and competitive market, but unfortunately as stated in Folorunso and Ogunseye (2008), there have been very little concern shown towards the study of user acceptance and requirements in this sensitive area. We believe that the study of performance/Algorithm behaviour may be mitigated using visualization techniques. Keeping the human analyst "in the problem-solving loop" may be facilitated through information visualization, which uses computer graphics to amplify cognition by taking advantage of human perceptual abilities (Card, Markinlay and Shneiderman, 1999).

Our research seeks to gain and understand the human RTDBS related tasks in order to design support tools that build on the strengths of both human analytic skills and machine processing and display capabilities. This paper will present the results from our user requirements gathering activities for information visualization support for RTDBS and introduce a prototype visualization tool called RT-DANGO designed to aid analysis in one specific RTDBS task.

2.0 DESIGN VISUALIZATION

A novel collaborative visual analytics application for cognitively overloaded users in the astrophysics domain was described by Cecilia et al., 2009. The system was developed for scientists who needed to analyze heterogeneous, complex data under time pressure, and make predictions and time-critical decisions rapidly and correctly under a constant influx of changing data using visualization and analysis techniques to enable a team of geographically distributed domain specialists to effectively and remotely manoeuvre a custom-built instrument under challenging operational conditions. One of the most important lessons learnt was the importance of forming closely integrated, interdisciplinary teams where members from all fields are involved in the participatory design process from the beginning of the project. Noting that the effective, usable interfaces for complex, time-critical scientific decision-making requires both domain expertise and software expertise, and the two must be closely intertwined for the ultimate success of the project.

Folorunso & Akinwale (2010) proposed a visualization tool / framework which has good extensibility for plugging in new data sources, supporting new data models, visual presentation types and allowing new graph layout algorithms. They presented the tool as an aid to academic research and teaching in the field of relational database systems and in practice to help database developers compare different solutions for database schemes with respect to normalization.

Folorunso et al., (2008) described Real Time Database System (RTDBS) as the mainstream of computer operations which aptly described as a system that produces result in a timely and consistent fashion. From the result gathered, he observed that the RTDBS designers are interested in seeing how their algorithms behaved, he then proposed an extendable framework using Model-View-Controller paradigm which was adopted for this paper. We therefore, followed a high-level process model consisting of monitoring, exploratory/confirmatory analyses and response. In a real-time database system (RTDBS) transaction travel through various components until their termination. Here, a RTDB system model is presented with all its various components described in figure 2. The model used, though modified was first proposed by Stankovic et al. (1991).

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

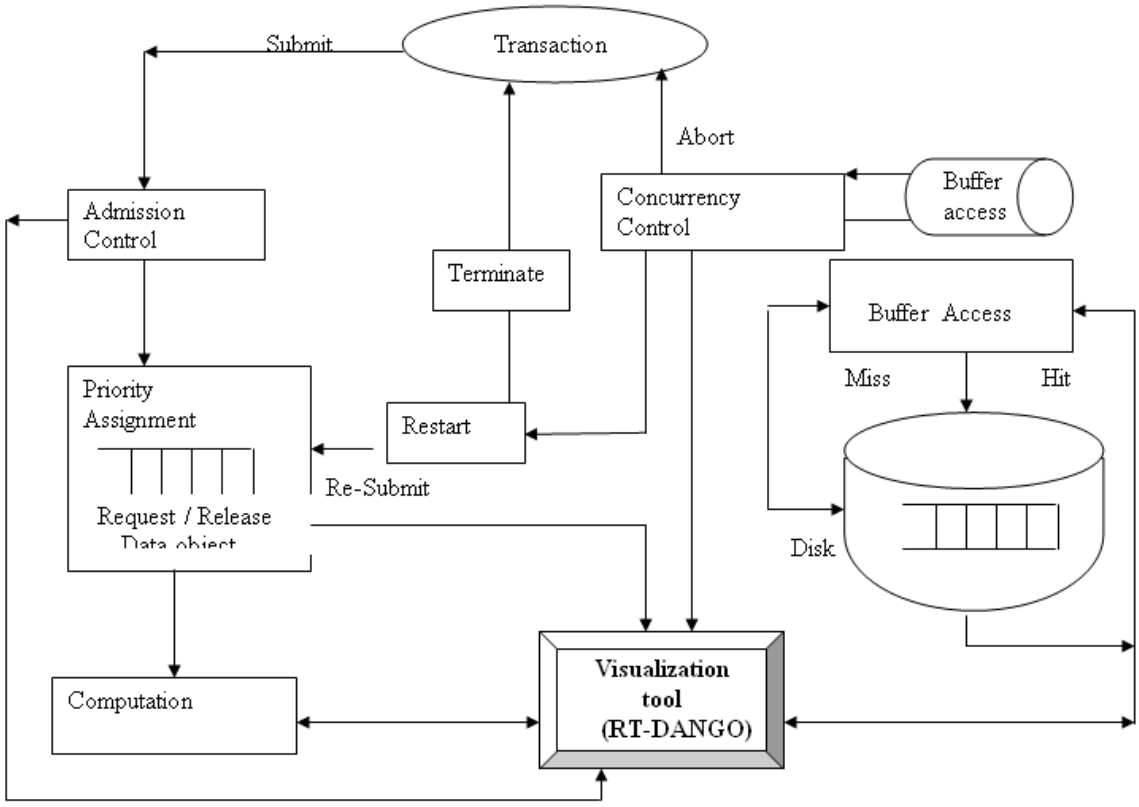

Figure 2: Architecture of the RTDBS with an embedded visualization output tool (RT-DANGO). Modified from (Stankovic et al, 1991)

3.0 USER REQUIREMENTS

In order to determine the actual needs of RTDBS experts, we conducted a user study through contextual interviews with diverse cross-section of RTDBS experts. Our findings revealed that most of our respondents are 'Application users' that have more than 5 years experience relating to banking applications, their major reasons for using RTDBS was for the realization of a key banking requirement which include the ability of customers to transact business from a branch other than that where his account is domiciled in Nigeria. The trend today is having a wide area network linking all the branches of the bank to central server at the head office. A further development in this direction is the extension of a bank's services to the internet. Customer accounts and other information will be stored in a bank-wide database hosted by the central server. Such architecture poses a number of challenges such as interruption of the communication link at the middle of an update and an account being updated simultaneously by two or more staff on different workstations and even different geographical locations. This is further complicated by a need for each branch to maintain a local database of its customers. The local and central databases are updated simultaneously each time a transaction occurs. The local database is reverted to when the central database is not available due to various reasons including a break in the communication link. These requirements have been realized in our applications by using features of database management systems such as transaction, two-phase commit, etc. Most users realized that, defining the deadline for a task has not been easy because several factors determine the speed of operations at any point in time. Such factors include the number of active workstations, the speed of the communication channels in use and the capacity of the servers. Furthermore, the version of the RDBMs we used does not seem to have facility for enforcing such deadlines if definable. Users also noted that the failure of one transaction jeopardizes the execution of the remaining transactions within the task. Users agreed that RTDBS run characteristics to be

Vol. 16 No.2 June, 2010 pp [135 – 144]

ISSN: 1817-3195 / E-ISSN: 1992-8615

considered in order of decreasing priority are transaction scheduling, memory management, admission control, I/O. They believe if the above typical characteristics are to be represented, the health of system will be quantitatively measurable, this will lead to the adoption of solutions to be adopted under such circumstances will be based on facts and not hunches. A user defined selection of representative transaction will also be an advantage. This is because where system degradation is caused by transactions some specific attributes or class of transactions may be the cause. Thus the ability of the user to select the set of transactions to be analyzed at any point in time will aid faster diagnosis of the problem.

The most common form of RTDB interaction is that of set-up and run; i.e. the algorithm and its parameters are defined in a set-up phase, and then executed in a run phase. The resulting output is typically examined after execution with any interesting solutions being further scrutinized at the user's discretion. Visualization however offers two-way interaction throughout a RTDB execution. This could be applied simply to permit some control over the speed of execution so as to further examine the visual representation of each transaction under study, or in a more direct manner to manipulate the algorithm's parameter or the current transactions internal values.

User's were asked how helpful, or destructive they will find execution controls through the use of a control panel to run, pause, step forward, step backwards, save a snapshot, and/or stop. The response was that such control panel will be helpful since it will make it easier to detect the exact step(s) in a process responsible for performance degradation which could lead to performance enhancement.

4.0 VISUALIZATION DESIGN FOR RTDBS

This section presents **X-builder**, a RTDBS-visualization framework. This framework is based on object-oriented development model. Wolfgang (1995) defined An application framework as an entity consisting of ready-to-use and semi finished building blocks. The idea of X-Builder came from the statement i.e. "extra building blocks".

Consider the figure below:

- Domain analysis
  - Architectural design
    - Framework design
      - framework implementation
        - Framework testing
          - Framework evaluation
            - Documentation

Figure 3. A Framework X-builder development Model

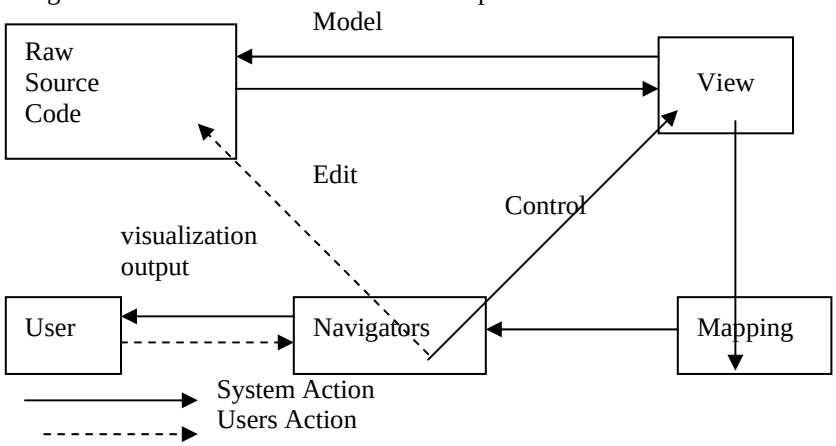

Figure 4. The Architecture of the RTDBS visualization framework. The three main modules: view, mapping and navigators, are shown in rectangular boxes. However, X-Builder links the navigator module

ISSN: 1817-3195 / E-ISSN: 1992-8615

to the visualization source module in order to enable the user's editing of the RTDBS parameters or components.

X-Builder, the primary data structures in the Database Visualizer, is a visualization specifically designed to support the detail oriented analysis task of RTDBS for RT-DANGO. This main visualization provides an overview of RT-DANGO using the X-Builder framework which can reveal patterns in the data and also help in understanding the runtime characteristics algorithm behaviour of Real-Time Database System. For example, in Figure 6, an example screen image taken from RT-DANGO is revealed using algorithm in figure 5. This example includes a single view, a coarse-grained (miss ratio versus number of transaction graph), button screen and two navigators and combo box generated data. This main visualization provides a high level starting point for analysis, but because details are crucial in this task, analysts can examine individual data packet details. RT-DANGO is **R**eal-**T**ime **D**atabase **AN**imation **G**enerati**O**n

The prototype was implemented with Visual Basic Version 6.0 on an IBM compatible PC running on Microsoft windows. Data can be captured live or read into the tool. Due to space limitation, only those features necessary for illustrating the functionality of X-Builder framework and building RT-DANGO were fully implemented; especially the 2D miss ratio line plot views, the fine-grained mapping and the control panel for navigation. These components were used to produce RT-DANGO Data computation, selector and Augmented Graphs. The miss-ratio versus number of transaction time graph uses the same DATASET as the fine-grained view of the visualization.

Within RT-DANGO, any changes made to the selection range will cause the entire miss-ratio versus number of transaction graph to be drawn again. Redrawing the contents of the graph involves clearing everything

```
Procedure GenerateGraphicalVisualization
  Declare InitialTime
  Declare Array FinalTime
For r = InitialTime yo FinalTime step 0.1
  If r mod FinalTime/20 = 0 then
   Array(r*10, 1) = “”
  Array(r*10, 2) = 0
Next r
Select case ChartType option
Case1 result.ChartType = Type2dBarChart
Case2 result.ChartType = Type2dLineGraph
Case3 result.ChartType = Type2dPieChart
Case4 result.ChartType = Type2dXY
Else Case resultChartType = Type2dLine
End Select
For r = InitialTime yo FinalTime step 0.1
Array(r*10, 1) = “”
ProbabilityRatio = r/(Exp(1-r)*WaitingTime))
TheItem = Format(r, “0.00” & vbtab & Format(ProbabilityRatio, “0.0000”)
MSFLEXGen value.AddItem TheItem
Array(r*10, 2) = r/(Exp((1-r)*WaitingTime))
MSCResult.ChartType = Array
Beep: Beep: Beep: Beep
Next r
End Procedure
```

Figure 5: ALGORITHM FOR CONCURRENCY CONTROL (MISS PROBABILITY)

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

A. From Requirements Gathering to Design

While functionality is currently limited, the basic structure takes into account several of the requirements from the users, notably the need for simultaneous high and low-level views within a single display. Because RT-DANGO is designed for analysis, exploration rather than speed is emphasized. In the analysis phase, the analyst has a hypothesis of what they are looking for and are often trying to prove or disprove the accuracy of an event and determine its severity. In learning how analysts perform the analysis task and how they transition between tasks, we determined that time would be the focus of the visualization because it is most often used as the starting point for the analysis task. Time is the central theme of RTDBS for several reasons:

- Result must be produced in a timely and consistent fashion.
- All data may not be permanent, some may be temporal
- Transaction has time constraints which require that the transactions are executed in such order that no deadlines are violated, requiring that scheduling algorithms and concurrency control are time cognizant.
- Meeting the deadline of a transaction might be more important than an exact result. Therefore temporal inconsistency is acceptable in some situations in order to guarantee that the deadline is met, i.e., the correctness of the result is traded for timeliness by relaxing consistency. Also by using imprecise computation techniques, which have significantly smaller execution times than the original transaction, the deadline can be met.

B. Data Source

RT-DANGO can be applied directly with its current functionality or extended through the application of the X-BUILDER framework to provide and enhanced data input and output

- *Familiarity*: All of the participants were already familiar with X-Builder interface, so we wanted to take advantage of this knowledge.
- *Flexibility*: A visualization that uses this source of data allows RTDBS analyst to collect data remotely on any platform, and bring the data set to their workstation for analysis.
- *Details*: Because RT-DANGO is designed to support the analysis task, the data needed to include low level details unavailable in many other data sources. Using this kind of data does involve a compromise, however, because data can grow extremely large very quickly, the current version of the prototype can run out of memory.

C. User Interface

This subsection describes the individual views and navigators available in RT-DANGO. Figure 6 showed example screen shot taken from RT-DANGO containing a coarse-grained miss ratio versus Number of Transactions, miss probability versus arrival rate, (Bottom), a fine-gained visualization (top right), a text boxes range selector (top left).

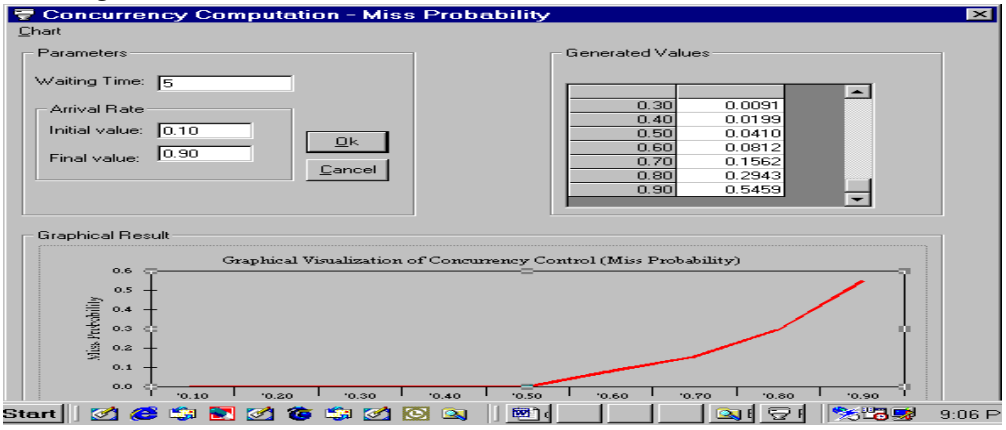

Figure 6: Description of an interface design of visualizing Miss Probability

ISSN: 1817-3195 / E-ISSN: 1992-8615

The design features of each component are described in the remainder of this subsection. The "augmented Miss probability versus Number of transaction graph" shows the result of the RTDANGO's run. The values of the miss Ratio computed are plotted on a single line graph (see figure 3). Visualizing the work of the algorithm presents the user with more direct view of the RTDBS behaviour than visualizations of the code or the "low-level" data values, this work may be a useful educational debugging aid used selectively as it presents so much fine-grained, detail of the RTDB runtime characteristics execution.

The "fine-grained view" presents the values of selected miss probability and their Number of Transactions values. This view is coupled to the visualization and supports the user's further investigation of the RTDBS behaviour. When the view is displayed the user can monitor the behaviour as it happens.

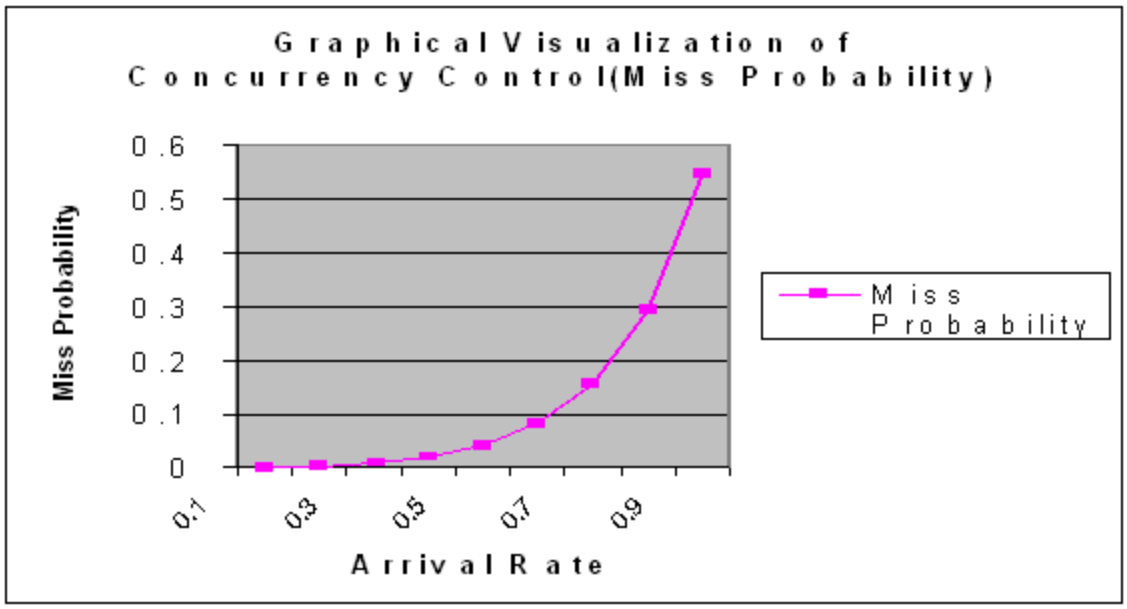

Figure 7: Offline visualization of Concurrency control (Miss Ratio)

Results from Graphical Package: MS-Excel

For comparative studies, figure 7 also shows the former method of displaying performance behaviour graphically through electronic spreadsheet program of which the features are not easily monitored dynamically. The inherent characteristics of the data being visualized are not visible. Comparative studies of figures 6 and 7 above answer the questions raised earlier.

The user has control over the various aspects of the display through the control panel. The waiting time is set using a combination of initial value and final value numeric text. By starting with a lower initial value to a higher final value, concurrent values are randomly generated which is visually animated in the graphical control area. If the values are changed, the random data is regenerated and the graph redrawn to reveal the present status of our RTDBS. The interval is calculated to display the entire data set; so an overview of the entire data set is presented to the analyst from the start. This combination of the time interval and the waiting time determines which portions of the data set are drawn on the screen. This two-stepped operation is somewhat clunky and will be modified in future versions.

5.0 USABILITY TESTING

We performed usability testing of RT-DANGO with 190 respondents. 29 are interested primarily in the theory of RTDBS and they are referred to as "RTDBS theory group. Those concerned in the application of RTDBS but as a direct result of their interest in RTDBS research and design are referred to as the "RTDBS

ISSN: 1817-3195 / E-ISSN: 1992-8615

research/designer". They are 79 in numbers. The last group of 82 respondents are concerned primarily with the using or solving problems, for which RTDBS offered an effective approach are referred to as the "Application group".

The format of the testing was split into three sections: a think-aloud exploration period to familiarize the users with the tool, a time-directed task period measuring performance and perceived difficulty, followed by a satisfaction questionnaire. Our RTDBS users are highly computer literate, often experienced in the use of more than one computer platform and familiar with programming in more than one computer language. Their area of application varied widely. Examples included research on different problem representations, the role of concurrency control and recovery, admission control, intelligent networks or cooperative systems. There were too few common domains to derive any application oriented categories although (when relevant), the respondents' motivation was used to form groupings. The respondents were motivated by their interest in RTDBS, or by a need to solve a problem for which RTDBS were appropriate problem solving approach. Those that were interested in RTDBS could also be further divided into those working on the problem-independent theoretical aspects of RTDBS.

All of the participants adhered to a similar, high-level task model consisting of three broad phases: monitoring, analysis, and response. The monitoring task focuses on monitoring the RTDBS itself, but also includes the surveillance of other monitoring systems as well. The transition between monitoring and analysis occurs when the RTDBS alert or other trigger event is generated. In moving from monitoring to analysis, the analyst will triage the event, determining instantly if further analysis is required. If the event is determined to warrant further investigation, the next phase is a much more detailed analysis of the event. In moving from analysis to response, a diagnosis of the accuracy and severity of the event is constructed. This section focuses on the monitoring and analysis tasks as they pertain to information visualization support. Participants were deeply mistrustful of techniques that used "black box" methods for clustering or organizing data. While clustering or aggregation of data is likely to be a necessary component of any information visualization with such vast amounts of data to visualize, participants wanted to understand *how* the data is clustered. That is, the visual layout should be intuitively understood by the user, exposing rather than hiding the reasoning for the structure. It is not enough to provide users with the ability to easily recognize an anomaly; a RTDBS analyst must know *why* it is anomalous in order to construct an accurate diagnosis.

The respondents were asked about the difficulties encountered while defining deadlines with task and transaction of RTDBS. They agreed that neither task were difficult or important to the outcome of the RTDBS. But researcher group considered the task both difficult and important. The advantages and disadvantages of visualizing the runtime characteristics of RTDBS was generally considered for identifying the deadline across different factors such as concurrency control, admission control, memory management, disk scheduling. Some respondents considered this to be of no advantage. This disadvantages noted here relate to the scalability of the visualization, in that it may present too much information and slow down the RTDBS, or the information presented may confuse the user or may be difficult to represent. The respondent's motivation did not appear to have a profound effect on their approach to using RTDBS. The difference between the opinions expressed by some of the theory group respondents and those of the research and applications groups were quite distinct. Those who belong to the theory group were primarily interested in their understanding of RTDBS. To them the application of RTDBS was to further their understanding through experimental fasting. The theory group respondents did not explicitly discuss their problems when they were asked to identify any difficulty they experienced. These users identified how interesting they found each step. The black-box approach of RTDBS produces results. However, users are unable to make improvements on the design or track down any possible causes of error because they do not know how the solutions are produced.

The respondents were generally in agreement that visualizing the runtime characteristics of RTDBS would be helpful for seeing performance behaviour of RTDBS most especially what goes on within the RTDBS as this would produce so much information to be of use. Enabling the user to select a subset of solutions from each

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

characteristic under observation would reduce the information overload but this introduces problems regarding the user's ability to select any of the runtime characteristics to visualize. However, emphasizing was for a flexible visualization environment in which the users can construct their own visualizations which is specific to the algorithm they are using or the problem domain being explored. Five of the respondents made comments as follows: The first suggested the development of a system monitoring program to interact with operating system. The second indicated a preference for supporting the design of a RTDBS rather than visualizing their execution. The other three respondents lay emphasis on the need for a tool that was generalizable and flexible. Generalizable in terms of being suitable for a range of RTDBS runtime characteristics and flexible in terms of being easy to change and suitable for visualizing information from an individual RTDBS run as well as from series of runs

Overall, users were generally satisfied with the tool and found it easy to learn and use, with one user commenting: "I like being able to look at the entire series of computers at a glance." The overall satisfaction mean score was 4.83 (std. dev. = 1.27), with mean satisfaction score for the visualization component being 5.17 (std. dev. = 1.11). Users rated the ease of learning a mean score of 4.42 (std. dev. = 1.44). Also, users self-rated performance was strongly correlated to ease of learning (Pearson, 0.84, $p<0.001$), mirroring the desire for the tool to be easily learnable for success. They also rated the ease of searching and of seeing patterns and anomalies in the data a mean score of 5.25 (std. dev. = 0.96) and 4.42 (std. dev. = 1.56), respectively. These preliminary results support the idea that visualization such as RT-DANGO can aid users in gaining an understanding and findings in RTDBS.

6.0 CONCLUSION

Usability testing of an information visualization is always a difficult task, and we present here one of the first evaluations of a visualization in the RTDBS domain. While the participants in our evaluation are relatively experienced in RTDBS, our hope is that a visual tool such as RT-DANGO will also assist RTDBS students' analysts to understand the concept behind RTDBS. We have demonstrated the importance of user involvement in both providing a baseline understanding of RTDBS work and how this understanding can facilitate the design of tools to support that work. From this understanding of RTDBS analyst's work we developed a simple task model: monitoring, analysis, and response. Many of the currently available information visualizations designed to support RTDBS appear most useful in the monitoring task, particularly in providing situational awareness. Instead, the RT-DANGO prototype focuses on the analysis task, providing a linked display of a high level overview that serves as the starting point for analysis with the low-level details needed to make an accurate diagnosis. RTDBS analysts often begin the analysis task with a hunch or hypothesis about an event based on experience and the event itself. The overview display creates a high-level picture of the RTDBS. RT-DANGO has many potential uses, including facilitating and aiding novice analysts in learning what constitutes normal on their Database system. Our future work will include broadening our understanding of RTDBS through additional field-work and refining RT-DANGO to better support the analysis task. In future revisions of RT-DANGO, we will include support for dynamic filtering based on RTDBS attributes, improve the link display, improve the user interaction mechanisms for controlling the visualization, and provide an intermediary step in moving from high- to low-level details.

REFERENCES

[1]. Abbott R. and Garcia-Molina (1988): "Scheduling Real-Time Transactions", SIGMOD Record, Vol. 17, No. 1.

[2]. Bestravros A, Lin K, and Son S editor (1997): "real-Time Database Systems"; Issues and Applications, Kluwer academic publishers.

[3]. Card, S. K., Mackinlay, J. D. , and Shneiderman, B. (1999): *Information Visualization: Using Vision to Think.* San Francisco, CA, USA: Morgan Kaufman Publishers, 1999.

[4]. Cecilia R. Aragon, Sarah S. Poon, Gregory S. Aldering, Rollin C. Thomas and Robert Quimby (2009): "Using visual analytics to develop situation awareness in astrophysics"; Palgrave Macmillan

ISSN: 1817-3195 / **E-ISSN**: 1992-8615

1473-8716 Information Visualization Vol. 8, 1, 30–41.

[5]. Tesanovic, D. Nystrom, J. Hansson, and C. Norstrom.(2003)Towards aspectual component-based real-time systems development. In *Proceedings of the 9th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA'03).*

[6]. Folorunso Olusegun & Ogunseye O. Shawn (2008) Challenges in the adoption of Visualization System: A Survey Kybernetes : The international journal of cybernetics, systems and management sciences vol. 37 number 9/10. pg 1530-1541.

[7]. Folorunso Olusegun, Longe H. O. D., and Akinwale A. T. (2008): Visualizing Concurrency Control Algorithms for Real-Time Database Systems. Data Science Journal, Volume 7, 20 October 2008

[8]. Harista, J.R., and Ramamritham(2000): "Real-Time Database Systems in the New Millenium"; online at http://www.ccs.cs.umass.edu/rtdb/splissue_intro.ps

[9]. Nizar Idoudi , Nada Louati , Claude Duvallet , Rafik Bouaziz , Bruno Sadeg and Faiez

[10]. Gargouri (2009) A Framework to Model Real-Time Databases *International Journal of Computing & Information Sciences Vol. 7, No. 1 .pg 1-11*

[11]. Kuo Tei-Wei and Lam-Kam-Yiu, editors (2000): "Real-Time Databases; Issues and Design", Kluwer Academic Publishers.

[12]. Olusegun Folorunso and Adio Taofeek Akinwale (2010) Developing Visualization Support System for Teaching / Learning Database Normalization. Campus-Wide Information Systems.U.K, Vol. 27 No.1

[13]. Ramamritham, K , (1993): "Real-Time Databases ", Intl. Journal of Distributed and Parallel Databases.

[14]. Son, S. Editor (1988): "Real-Time Databases", ACM Sigmod Record, 17(1).

[15]. Stankovic, J. Ramamritham K., and Towsley D. (1991): "Scheduling in Real-Time Transaction Systems", Foundations of Real-Time Computing: Scheduling and Resource Management edited by Andre van Tilborg and Gary Koob, Kluwer Academic Publishers, pp. 157-184.

[16]. Ulusoy O. (1995): "Research issues in Real-Time Database Systems", information Sciences 87, pp. 123-151

[17]. Wolfgang Pree,(1995) Design patterns for object-oriented software development, ACM Press/Addison-Wesley Publishing Co., New York, NY.

ISSN: 1817-3195 / **E-ISSN**: 1992-8615