Combined Event & Process Simulation Model of a Distributed Data Collection System

Hubert A. Bahr
SIMULATION, TRAINING AND INSTRUMENTATION COMMAND
12350 Research Parkway
Orlando, Florida 32826-3276
(407)-380-4477 bahrh@orlando-emh4.army.mil

Abstract:

Event and process driven simulation techniques are developed to assess loading and throughput in a multiprocessor-based distributed data collection system. Aspects of both simulation techniques are utilized to facilitate rapid model development while retaining flexible selection of configuration parameters. Statistics obtained for expansion capability and latency times for normal and priority players indicate bus traffic itself is not a limiting parameter unless process size exceeds local memory capacity. However, channel loading can cause significant delays unless properly balanced.

Background:

The Army's National Training Center(NTC) at Fort Irwin, California provides the most realistic training environment for heavy (armor, tank) combat forces in the world for a Brigade size task force. In order to enhance the training effectiveness of this simulated combat environment the Army has constructed a large automated Instrumentation System (IS). This system provides immediate feedback to the soldier by simulating the effects of real combat in providing weapon firing, target hit and target kill signatures in real time. It also provides the capability to collect these "events" as well as position and attitude information on the various "player" platforms in the field. The player platforms are the various combat vehicles (tanks, trucks, helicopters and fixed wing aircraft) and foot soldiers involved in this simulated combat. One of the key subsystems is the Range Data and Measurement Subsystem (RDMS) which collects and transfers the field data to the Core Instrumentation Subsystem (CIS).

System Requirements:

Some of the Key technical requirements pertinent to this analysis that drove the Motorola design as specified in the RDMS-Upgrade System Specification Document (ref. 14) are as summarized here. The system shall provide 2000 player capacity and designed for expansion to 4000 players. The player mix shall be 70% ground vehicle(GV), 4% rotary wing(RW) aircraft(helicopters), 19% dismounted troop(DT), and 7% Observer Controller(OC) units. The GV units shall report their position at a maximum rate of once every 5 seconds, the RW twice each second, the DT once every 45 seconds, and the OC units at a rate dependent on their current mounting configuration, i.e., either the same as the GV, DT, or RW unit dependent on where it is located. In addition the position location reports (x, y, z) require

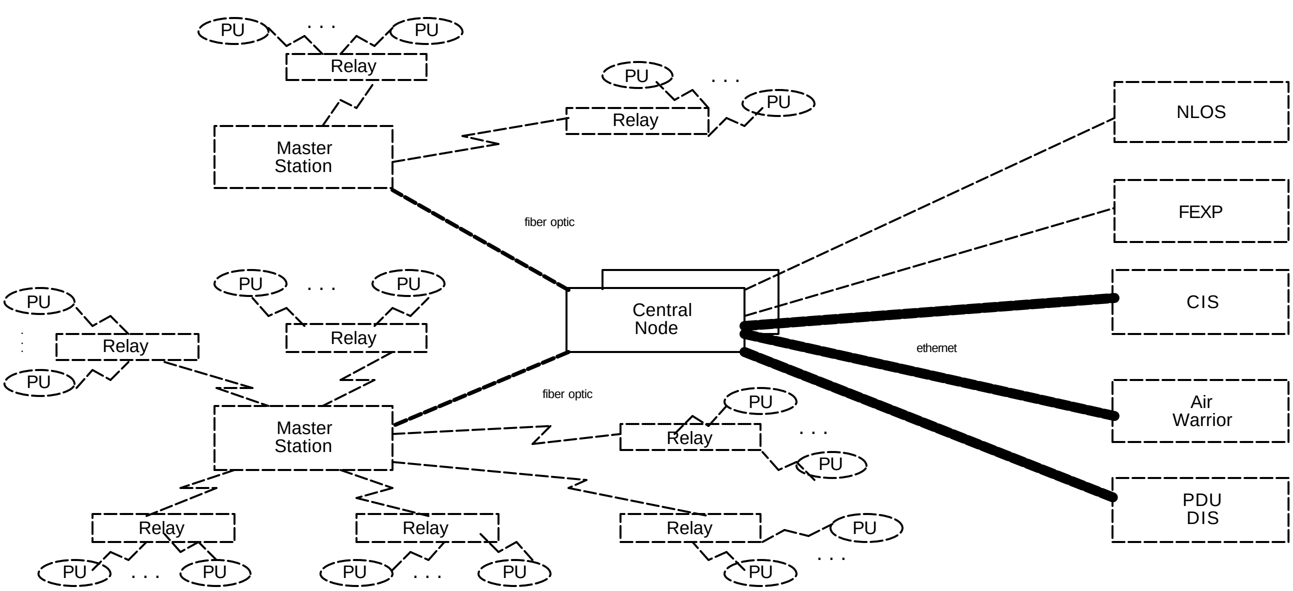

Figure 1

velocity and acceleration for GV and RW units but only position for the DT and OC units. In addition to the position reports the player units will be reporting events as they occur in the field. These events specified as occurring at a maximum rate of 6000 per minute with a peak of 120 per second. 95% of the data must be received by the CIS within 5 seconds of occurrence. This data must also be delivered in time ordered sequence. In addition to providing the data to the CIS the system was also provide the interface to the Air Warrior (Air Force fixed wing aircraft players) and the Distributed Interactive Simulation Network(DIS) using Protocol Data Units(PDU).

In addition to above performance requirements these resource constraints were also imposed. All computer equipment used must be programmed in Ada and use a validated compiler. No computer resource shall use over 50% of its capacity. The system was also expected to provide radio frequency coverage of 95% of Fort Irwin. No single point of failure should cause a data outage greater than 2 minutes. Each player unit shall provide the capability to store 2000 records pertinent to reconstructing its activities.

System Description:

The key features of the Motorola system (figure 1) reference(1,2) pertinent to this analysis are: The player unit consisted of an MC68332 based single board computer and a cellular telephone. The radio relay network consisted of two 39 channel Cellular radio base stations and 7 relay sites. One spare channel for each relay site was provided as well to allow switching between sites. For expansion to the 4000 player environment Motorola proposed doubling the number of channels at each site. They information is transferred to the central node(CN) of the RDMS-Upgrade on individual 4800 baud serial channels that are multiplexed on to a T1 carrier and sent over a fiber optic cable from the top of the mountains to Shelter 130 the site of the CN. At the CN the serial channels are de multiplexed and fed into each of the dual CN computers. The CN computers (figure 2. 1/2 Central Node Computer) are each a multiprocessor configuration in a VME bus chassis (ref. 2,3,13) consisting of one S-bus board interfaced through an S-bus to VME-bus adapter, multiple VME (ref. 12) based singleboard computers. Each of the VME computer boards (ref. 11) are populated with 4 Megabytes of RAM, A VMEbus adapter with DMA and burst mode capability, and an ether-net adapter with DMA. They are also populated with up to 4 mezzanine adapters that provide up to 8 serial channels on each adapter for a total of 32 channels available on one VME board. Memory access maps are provided on each board that allow the allocation to local only or local and global access. All memory

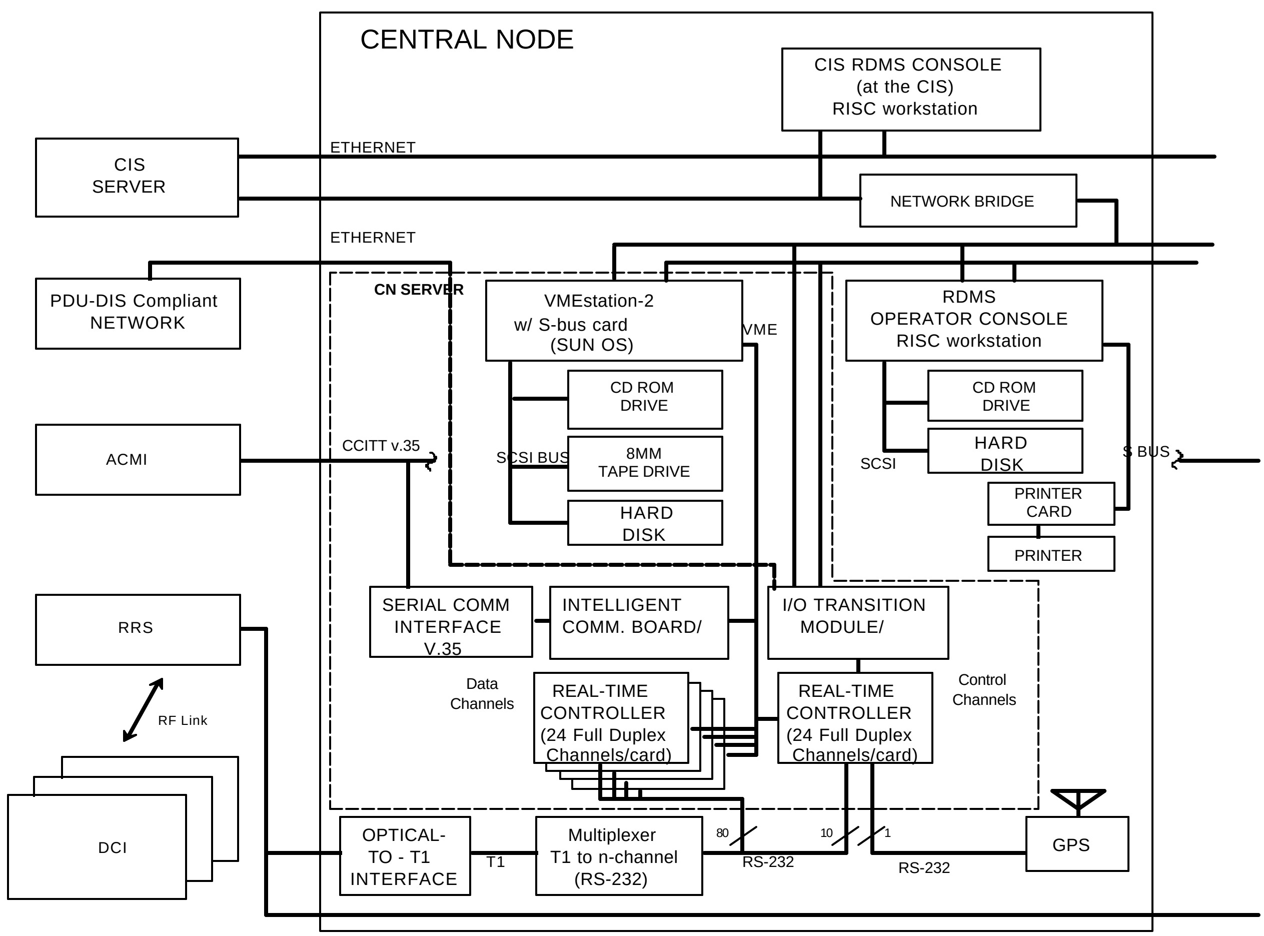

Figure 2 1/2 CENTRAL NODE SUBSYSTEM BLOCK DIAGRAM allocated for global access is directly accessible over the VME bus. In addition if data is transferred over VME bus via the VME-bus adapter DMA controller it can transfer n=1 up 256 Quad words(64 bits) in n+1 100 nsec cycles for a peak data rate approaching 80 Megabytes(640 Megabits) per second. The most typical transfer is one double word in 2 (100 nsec) cycles for a data rate of 20 Megabytes per second.

The assumed mode of operation for the VME boards is in non parity configuration with a cache write back policy for ram and write through policy for I/O. The burst cycle count is 4,1,1,1 for the first and successive double words up to 4 (a cache line) of 40 nsec cycles. I/O access is 4 clock cycles per access. These boards employ a local bus snoop policy for cache coherency.

Analysis Goals:

The initial focus of my analyses was directed toward the expansion capability of this architecture toward incorporation of more sophisticated player units on the battlefield. This was based on a recent request to estimate the cost of incorporating a guided missile system in the simulated combat. This new system would have the ability built in to generate target images. What it needed was a near-real-time identification and location of potential targets and then the capability of having the results of the engagement communicated to the target. Since the primary function of the RDMS was to collect the position location and activities of players on the battlefield and it also provided the communications path for sending results back to the players, it would be more cost effective if the needed functions could be incorporated into the RDMS-CN. This focus was later expanded to include the radio relay system to better understand and identify methods that might be used to reduce latency times for "priority players". The CN processor is also used govern the loading on the radio relay channels. The radio relay network dictates the maximum amount of data that is available for processing so initially this information flow was used as a constraint of the analysis. The next constraint was the number of processors that could be installed in the VME chassis. Motorola informed me (ref. 9) that real driver here was the physical connection of serial channels to back panels, as result they were using a chassis configuration that could accommodate up to 24 boards although a maximum of 11 boards were required for the 4000 player RDMS. As a result the investigation was isolated to looking at the VME-bus that was shared by the processors and the individual board local busses, this was later expanded to include the radio relay network and its channel allocation strategy. To do this analysis, I decided to try to model the architecture and bus traffic on each bus. Motorola had been tasked to model the communications network and later expanded their model (ref. 10) to include the CN as they were concerned about setting system allocations properly to insure compliance with specification requirements. A major by product of this investigation for me would be detailed insight into the capabilities of the proposed system so I would be able to make acceptance recommendations to the Project manager with confidence.

The Model:

My previous experience with modeling was to directly write the model in a programming language such as Pascal. In this situation I decided to look at other techniques. In the article CSIM: A C-Based, Process-Oriented Simulation Language, I was introduced to the concept of using a set of simulation functions from a precompiled library to assist in building the Model. Pritsker (ref. 5) provided examples that were used to gain familiarity with modeling techniques and Nance (ref. 4) provided the basic definitions. Jump (ref. 6-7) provided the Modeling library (YACSIM, NETSIM) and Users manuals that were actually used to construct the Model. For model performance (time to execute) and size considerations I used both processes and events (ref. 4,6) to model the activities of the RDMS subsystem.

The primary advantage of process-oriented techniques is that they can be used to model the behavior of independent components that operate concurrently. YACSIM (ref. 6) provides the capability to easily generate multiple instances of a process from the same process body (definition). In YACSIM, a process exists from the time it is created until the time it is terminated. Its main disadvantage is that the amount of memory resources required can be prohibitively large if numerous long life time operations occur over the duration of the simulation. On the other hand, events are only memory resident when active and thus reduce memory consumption and the likely hood of thrashing and virtual memory overhead. However events cannot exist over a period of simulation time hence are limited in use. An event can be used to schedule the creation of a process and thereby allows a process to exist only for the period of use. This is especially useful in this model for all messages, which are then created at the start of transmittal and disappear once they are delivered. If all messages were created at initiation of the model it would severely limit the size and duration of a model run.

The model consists of a process for each player unit, and a resource for each communications channel, processor board, each output interface, and the VME bus. These processes exist for life of the simulation. In addition the model creates a process for each event message that exists only for the life of the message. Since there will be 6000 event messages in one minute yet their life span could be as short as the message length, a significant amount of memory was saved by creating and destroying the event message processes. To schedule these messages they were initially activated as events which upon reaching the designated time would then create the event message process. Other temporary processes of even shorter lifetimes are used as well when an object (such as an event message) needs to use more than one resource at the same time. Resources are used to model the various resources that the objects use as they pass through the system. These resources act as servers and can be set up for various service strategies such as First In First Out with priority. The simulation library provides these resources and predefined statistic collection functions for them (ref. 6).

Starting at a vehicle player unit a position report is scheduled to occur every 5 seconds or 5000 milliseconds in the model timebase. In function VehMSG which was initially scheduled to randomly start sometime in a 5 second period, this object tries to access a channel resource keeping track of elapsed time from the start of this effort. If unsuccessful, it backs off for a random period as proposed by Motorola (ref. 10) and tries again. Once successful it ties up the channel for the length of the message. At the same time it periodically interrupts the serial Board that is associated with receiving the channel and uses that resource for the time taken to service the interrupt. (The time to service was determined both by a pipeline trace of the generated assembly code and a logic analyzer measurement made by AP Labs and reported to Shuman (ref. 9)). Upon message completion only the transfer time is modeled to move the over the VME-bus to the Router Card. All three resources are used for this transfer, the Serial Card, the VME-Bus and the Router Card. At this point the transfer interval time for the object is recorded. Then the resources for the additional destinations are used simulating the transfer of data to them. Other functions that currently share the resources with the messages are also modeled especially the output functions from the various boards to other destinations. They are in no means complete but primarily just impact individual boards. The other player objects are treated likewise with differences in message lengths, and periods of reoccurrence. A channel monitoring process is also running that attempts to monitor the channel usage and provide inputs into an algorithm used to reassign channels in attempt to load balance the system. Timings are derived from the technical references (ref. 11,12) and answers to questions (ref. 9,10). The algorithms used are based on Motorola's design (ref. 1,2). The figures are extracted from figures provided by Motorola in their design documentation.

ANALYSIS:

The initial concern about the use of the VME-bus was quickly diminished based on the results indicated in these statistic's records that shoes the average occupancy any millisecond of 0.28% although it does show that at times that more than one process needs to use the bus, but the max. time to execute the transaction was 400 nsec.

Statistics Record VME.length:
  Number of intervals = 1525719, Max Value = 2, Min Value = 0
  Sampling interval = 54999.9, Sampling rate = 27.7404
  Mean = 0.00277557, Standard Deviation = 0.0526402
End of Report

Statistics Record VME.time:
  Number of samples = 762859, Max Value = 0.0004, Min Value = 0.0002
  Sampling interval = 54999.9, Sampling rate = 13.8702
  Mean = 0.000200111, Standard Deviation = 4.02015e-06
End of Report

The next item noticed was although the system used a parallel hardware architecture it is serial software design. This is evidenced where all the data is transferred to the Router card prior to being sent to any of the other cards. In fact all VME bus transactions have the Router card as one terminus. In this case the real bottleneck is the Router card. This isn't as evident in the current run as other activities are shown on each card which mask the bus transactions. These activities were added as they can interact with activity on the bus although the VME-bus has higher local bus priority than the processor. The above results are with two postulated future boards added to the system.

Another portion of the system resources impacted by adding more sophisticated players is the communications network. Two primary area's of concern were the resolution of contention for a communications channel and the resulting impact on the message latency (time to traverse the total system), and the algorithm to balance the loading on the individual channels. These were investigated by using multiple runs changing both the balancing algorithm and the number of channels. If a position location message is delayed beyond the next scheduled updates, those updates would be skipped. The enclosed sample runs show the results of these variations. Note the improvement in the Vehlatency in the equalized vs. baseline situation. In the situation of 1.63 sigma + the mean for 95% of the messages the 39 channel equalized case is similar to the 43 channel unequalized case. This is probably due to the wider spread in channel utilization.

Statistics Record Vehlatency:{39 chan equalized}
  Number of samples = 14973, Max Value = 18570.9, Min Value = 63.4854
  Sampling interval = 54995.8, Sampling rate = 0.272257
  Mean = 911.848, Standard Deviation = 1354.31
End of Report
Statistics Record ComoUtilization:
  Number of samples = 2106, Max Value = 0.9111, Min Value = 0.0589439
  Sampling interval = 54020, Sampling rate = 0.0389856
  Mean = 0.80658, Standard Deviation = 0.118186

Statistics Record Vehlatency:{39 chan baseline}
Number of samples = 14883, Max Value = 33163.4, Min Value = 63.4854
Sampling interval = 54998.1, Sampling rate = 0.270609
Mean = 1018.13, Standard Deviation = 1865.95
End of Report
Statistics Record ComoUtilization:
Number of samples = 2106, Max Value = 0.9116, Min Value = 0.25828
Sampling interval = 54020, Sampling rate = 0.0389856
Mean = 0.776656, Standard Deviation = 0.13036

Statistics Record Vehlatency:{43 channel baseline}
Number of samples = 16432, Max Value = 24307.4, Min Value = 63.4854
Sampling interval = 59997.2, Sampling rate = 0.273879
Mean = 805.02, Standard Deviation = 1612.79
End of Report
Statistics Record ComoUtilization:
Number of samples = 2537, Max Value = 0.912053, Min Value = 0.227307
Sampling interval = 59020, Sampling rate = 0.0429854
Mean = 0.730067, Standard Deviation = 0.152388

CONCLUSIONS:

On the basis of this study it appears the expansion of processing for future players is more than adequate with more than 50% of the available bus slots still available after expansion to 4000 players. The bus traffic itself should not be limiting unless the process size would exceed local memory capacity. I would recommend the relook at the Software architecture to better take advantage of the hardware configuration. The Router board should not force all bus transfers to involve it. It is very important to have an adjustment procedure to equalize channel loading. It can have as much to do with data latencies as adding multiple channels. However, channel loading as currently postulated can cause some significant delays (as high as 30 seconds) in some instances so it is highly desirable to reduce this for priority players. It would be wise to establish some method of prioritization perhaps a few lightly loaded channels for those players. However, once the system starts distributing the load over both master stations, latencies drop significantly.

Bibliography:

1. **SYSTEM DESIGN DOCUMENT FOR THE *NATIONAL TRAINING CENTER (NTC) RANGE DATA MEASUREMENT SUBSYSTEM (RDMS) UPGRADE*, Motorola Inc., *Tactical Electronics Division.***

2. **SOFTWARE REQUIREMENTS SPECIFICATION FOR THE *NATIONAL TRAINING CENTER (NTC) RANGE DATA MEASUREMENT SUBSYSTEM (RDMS) UPGRADE***, Volumes II-IV. **Motorola Inc., *Tactical Electronics Division.***

3. IEEE Standard for A Versatile Backplane Bus: VMEBUS. ANSI/IEEE Std 1014-1987. Published by The Institute of Electrical and Electronics Engineers, Inc.

4. The Time and State Relationships in Simulation Modeling, Richard E. Nance, Virginia Tech, Communications of the ACM April 1981, Volume 24, Number 4.

5. Introduction to Simulation and SLAM II. A. Alan B. Pritsker, Third Edition. Systems Publishing Corporation 1986.

6. YACSIM Reference Manual, Version 2.1 March 1993, J. Robert Jump. Rice University.

7. NETSIM Reference Manual, Version 1.0, May 1993. J. Robert Jump. Rice University.

8. CSIM: A C-Based, Process-Oriented Simulation Language, Herb Schwetman, Proceedings of the 1986 Winter Simulation Conference.

9. Personal communications with Brian Shuman, Motorola Inc., Tactical Electronics Division. June 24, 1993.

10. Personal communications with Richard Ward, Motorola Inc., Tactical Electronics Division. July 6th 1993.

11. Embedded Controller Programmer's Reference Guide. Motorola, Inc., Computer Group.

12. MC68040 User's Manual, Motorola Inc.,

13. VMEbus for Software Engineers, Ed Skinner, ***Embedded Systems Programming, June 1993.***

14. **SYSTEMS SPECIFICATION FOR THE NATIONAL TRAINING CENTER(NTC) RANGE DATA MEASUREMENT SUBSYSTEM (RDMS) UPGRADE,** PMT-91-S016A, PREPARED BY U.S ARMY SIMULATION TRAINING AND INSTRUMENTATION COMMAND(STRICOM).