THIRTEENTH EUROPEAN ROTORCRAFT FORUM

S.1

PAPER No. 25

EH101 AVIONIC INTEGRATION PHILOSOPHY

E.GALLI

AGUSTA SISTEMI
Via Isonzo 33
21049 Tradate (VA) - ITALY

E.CAMBISE

DATAMAT S.p.A.
Via S. Martini, 126
00143 Roma - ITALY

September 8-11,1987
ARLES, FRANCE

ASSOCIATION AERONAUTIQUE ET ASTRONAUTIQUE DE FRANCE

1 ABSTRACT

The intent of this paper is to describe the approach employed by AGUSTA-DATAMAT during the HW/SW and system integration and testing of the Italian Navy EH101 Helicopter Aircraft Management System (A.M.S.) and Mission Avionic System (M.A.S). As previously described in Paper No. 24, the avionic system architecture is centered around a Mil 1553B Bus which connects the Computer units with the other Subsystems, and an Arinc 429 type connection mainly serving the comunication links between the Aircraft Management Computer and the Navigation Subsystems.

This paper will discuss all the different steps followed during the integration and testing of the avionic system in terms of:

a. Requirements
b. HW/SW integration
c. System integration

2. Foreword

In order to carry-out the entire integration and testing of the Avionic System the following requirements had been established as being the most important guidelines:

a. Support versus the Aircraft Management and Mission software development phase,

b. Software Acceptance Tests in contrasts to the software operation and requirements,

c. Integration of the the Aircraft Management System (M.A.S.) and Mission Avionic System (M.A.S) in separate environments,

d. Overall integration between A.M.S. and M.A.S with all their relevant links and data exchange.

3. Support to the SW development

Testing of the software at the development stage is carried out as two different phases:

- Host Testing

- Target Testing

4. Host testing

At this phase the Perspective Interpreter is used. It permits the following:

- Contemporary testing by different users as compared to Target level where only two users can work due to the number of systems available (two processors)

- Test correct processing of a single software module

Such an environment, as described in Fig.1, is based around the Software factory in which the module resides. At Host leve it is not possible to test the following items:

- Correct synchronization between the elements of the A.M.C and/or M.C.U. system

- Test the interaction between the operating system and the external world

- Evaluate the processing time of the various software modules. This evaluation is indispensable for the distribution of the CPU load on the processor and into each subframe

At first glance it seems that the workload carried out in this phase is not very meaningful, but the main purpose pervading this step is the checking of procedural correctness, and putting all the different users into the multiple module test in order to get a large amount of information for correct development.

5. Target testing

The Target testing is done using both Perspective tools and an In-Circuit-Emulator which dialoge with the Target. This makes possible the following steps:

- Interaction with the operating system,
- Data exchange between tasks,
- Real time interaction with Mil 1553B Bus Controller, Remote Terminal and Arinc 429 line protocol,
- CPU load at subframe level.

The user has available debugging tools such as the Perspective debugger or the I.C.E. that allow debugging of single

modules against anomalous processing but do not yet take into consideration the timing evolution of the tasks. In order to minimize the interaction of the debugging process with the computer unit it is very effective to suspend the target at every frame pulse or multiple, to allow the user to examine the task's status and then restart the execution from the point of interruption.

The execution of such an operation is provided for by an interrupt that can be (automatically or manually) driven by the the RIG computer or the operator at which the AMC/MCU unit is connected as shown in Fig 2. In doing so a "near real time" operational mode is obtained and the "RIG" has all the time necessary to perform data cheking and comparisons between one suspended status and the next.

When dynamic tests with no interventions are required, the "Real-time" mode is used to store all the bus traffic while monitoring of a limited amount of the parameters is allowed. The term "RIG" has just been introduced, let us explain what we intend by this term. The "RIG" is the environment, Hardware/Software, that allows the execution of the different Test Plans and permits the validation of the entire system. Due to the architecture of the EH101, Aircraft Management and Mission, a complex "Rig" structure has been developed; here below you will find a description of the items.

6. A.M.S. Rig

The Aircraft Management System (A.M.S.) Rig is dedicated to the testing of the A.M.S. software and is based around a Vax 11/750 family computer. In order to emulate and/or acquire Mil 1553B bus traffic, a set of Multiplex Bus Terminals enabled to emulate Bus Controller, Multiple Remote Terminal and Bus Monitor is connected via D.M.A. and is under control of the "RIG" software environment. The other main information medias are the Arinc 429 protocol lines. To acquire and/or emulte them, a Bus station is connected to the "RIG" via an IEEE488 line. The emulation of the navigation equipment (Rad-alt, Doppler, A.H.R.S., I.R.U and A.D.U) is made through dedicated single board computers on which the sensor protocol emulation has been implemented and upon which they receive the input characteristics from the "RIG" via a RS232 link. Doing it in such a way as this, the "Rig" has only to take care of the management of the input data and does not needs to dedicate its time to the timing or tagging functions. Another non-secondary function is the emulation of all the signals to be fed to the Sensor Interface Unit (such as, Electrical System, Hydraulic System, Engine System, Transmission, etc.) in order to create a full Helicopter system. A micro-processor approach has also been adapted and is able to be remotely controlled during the dynamic simulation. In this configuration the A model of the A.M.C. complete with a set of Common Control Units, a Symbol Generator and it's related

C.R.T are available. A serial link between the A.M.S. Rig and the A.M.C. is provided to utilize the triggering facilities of the Rig upon request of the operator. In Fig 3 the facilities have been schematized in terms of functional blocks.

7. M.C.U. Rig

The Mission Computer Unit (M.C.U) Rig, as stated in Paper No.24, is integrated with the Mission Software Factory creating an environment able to debug and test the entire scenario. This time the "RIG" is based on a Micro Vax II family computer.

A set of boards emulating the Mil 1553B functions (Bus Controller, Multiple Remote Terminal and Bus Monitoring), an In-Circuit Emulator, the A Model of the M.C.U. together with the Common Control Units, Common Waveform Unit including it's relevant mission monitors constitute the hardware of the "RIG". A software package controls the entire facilitie.

The data exchange between the "RIG", and the Host is done via a RS232 line, this is because the Perspective environmen exchanges data with others only through a serial line or a data file. Information exchange between the "RIG" and the Target is done through the Mil 1553B bus, Arinc 429 lines and a serial line on which, in debugging session, a limited data exchange between the "Rig" and the A.M.C. or M.C.U. memory takes place and permits the acquisition of a data package for comparison purposes.

8. Rig's environment

As stated above, the "RIG" is controlled by a set of software packages that allow the operator to move in different situations pursuant to a specific need. The environment can be driven in the following manners:

- Manual
  A single message is prepared by the operator and passed to the system (A.M.S. / M.A.S.) under test to be executed. Data is passed either in raw form or in engineering unit form.

- Semiautomatic
  Possibility of defining pre-built messages and storing them in a data file to be passed to the system.

- Automatic
  Possibility of defining the evolution law that effects the message items. Such laws could be random, sinusoidal, quadratical or via a user-defined law built by a procedure.

In order to simulate a real scenario evolution, other functions are provided within the "RIG" software package such as:

- Helicopter simulation

This function is present in both of the "RIG", A.M.S. and M.A.S., but with different porposes. In the A.M.S. Rig it furnishes the basic aircraft characteristics (velocity, height, body axis, accelerations, etc.) to the different boards emulating the navigation sensors. The model will follow a predefined route in terms of way points and it will also record the scenario evolution. In the M.A.S. Rig the major elements to be simulated are the target evolution associated with its motion law and steering commands. During the evolution of the integration and test phase other sets of functions are needed in order to simulate data processing and formating between the sub-systems, the Aircraft Management and Mission Computers. The sub-system simulation are sets of functions emulating the proper sub-system equipment (e.g., Radar, Sonar) with the capability of associating with them major characteristics and limited controls over data functionallity.

The two "Rigs" will also emulate one another, however, only for the essential data computation and exchangeability. The evolution of the scenario is also controlled by specific functions that allow the operator to override the state at any time.

We have, up to now, discussed the ways of stimulating the software, but a non-secondary function is the monitoring of the data flows. Main functions of the monitoring module are:

- Recording all the bus traffic (Mil 1553 and/or Arinc 429) on a file for off/line analysys, each message recorded is tagged by the bus station in terms of time, bus errors, exceptions

- Trigger recording, possibility to define various triggering events (value, bit match, bus error, frame number, etc.)

During the monitoring phase it is possible to display data (raw, engineering unit, message format, elapsed-time, etc.) and, if any, errors associated with the message under examination by the use of on-line functions.

While conducting testing activities (real-time or near-real-time) we do not have a lot of time available for data analysis. A set of functions is available for off-line display. It is possible to:

- Display all messages recorded, raw or enginering data,
- Display only specific messages (selective choice),
- Display error information recorded,

- Display time information,
- Display differences with respect to other session data,
- Statistical analysis,
    1) average, min/max values of intermessage gap and response time
    2) bus occupancy in percentage for major loop, or frame by frame
    3) percentage of errors
    4) message kinds exchanged on bus

All the computed data is presented in tabular form and/or in graphical format. In Fig. 3, a schematic is presented that summarize all the concepts described above.

9. Overall integration

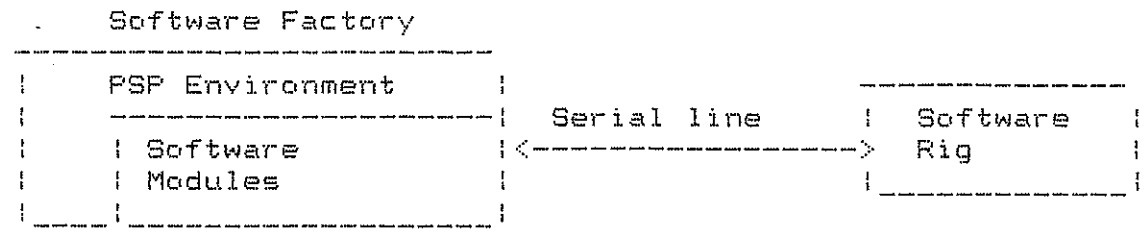
All we have discussed up to now is dedicated to the two world Aircraft and Mission integration and testing, the final step is the overall integration where the two worlds are integrated on a helicopter mock-up Rig utilizing the final model of the system (B model equipment). The hardware this time is based on three computing units; a vax 11/750 for the scenario and helo simulation, a microvax II for the Aircraft Avionic bus handling, and a microvax II for the Mission bus handling (see Fig. 4). The main problem in such a configuration is to synchronize the two integration Rigs in order to, (a) simulate a common scenario evolution and, (b) guarantee consistency of data between the two environments. The vax 11/750 performs the scenario and helo evolution; at a programmed time interval it supplies data to the two microvax units, each computing unit will then make available data to the respective avionic systems. In such phases all the debugging and monitoring facilities of the Rig are available for usage. Monitoring on both, basic and mission buses, is available and data is saved on disk files fo further analisys or system replay. The capabilities that we have during the overall integration are mainly focused on off-line data analysis and simulation replay. The function of simulation replay is deemed important for system debugging when anomalous operation appears, it is very useful to replay the event and analyze it. In such a phase it's also possible to replay the evolution up to a predefined point, after that, alternative evolutions can be taken for different evaluation. The result of all the above described architectures is an environment of total software/hardware integration and test support, taking also into consideration the complexity and differentiation that exists between the two worlds.

1Ø. Conclusion

In conclusion it can be stated that the philosophical approach at the integration and test phase provides a very powerful tool to the engineers. They can attend the different test procedures and also acquire statistical data items for a full project report.

SOFTWARE TESTING

Host testing



Target Testing

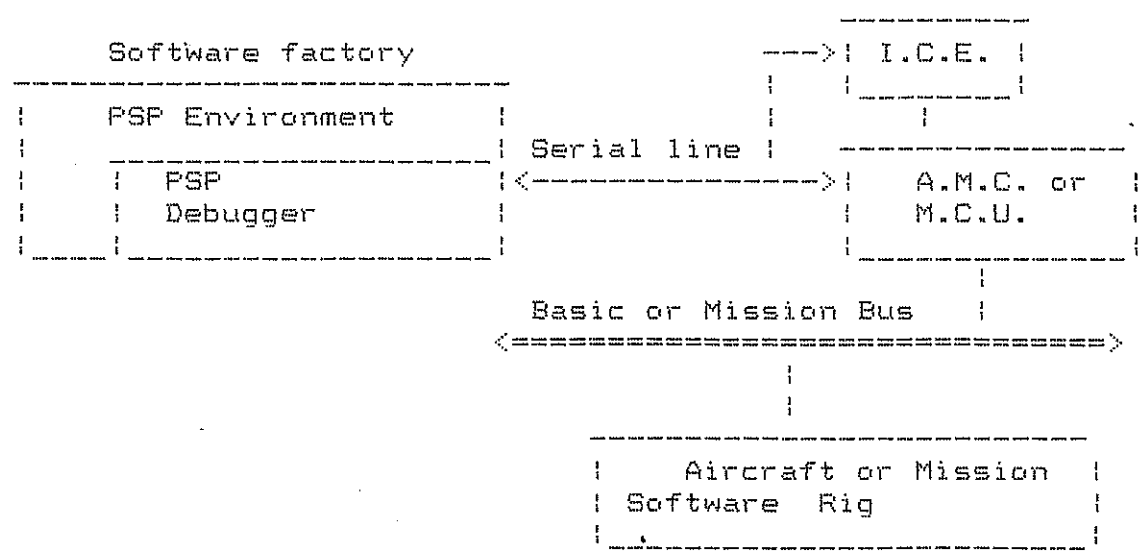


Fig. No.1

1.

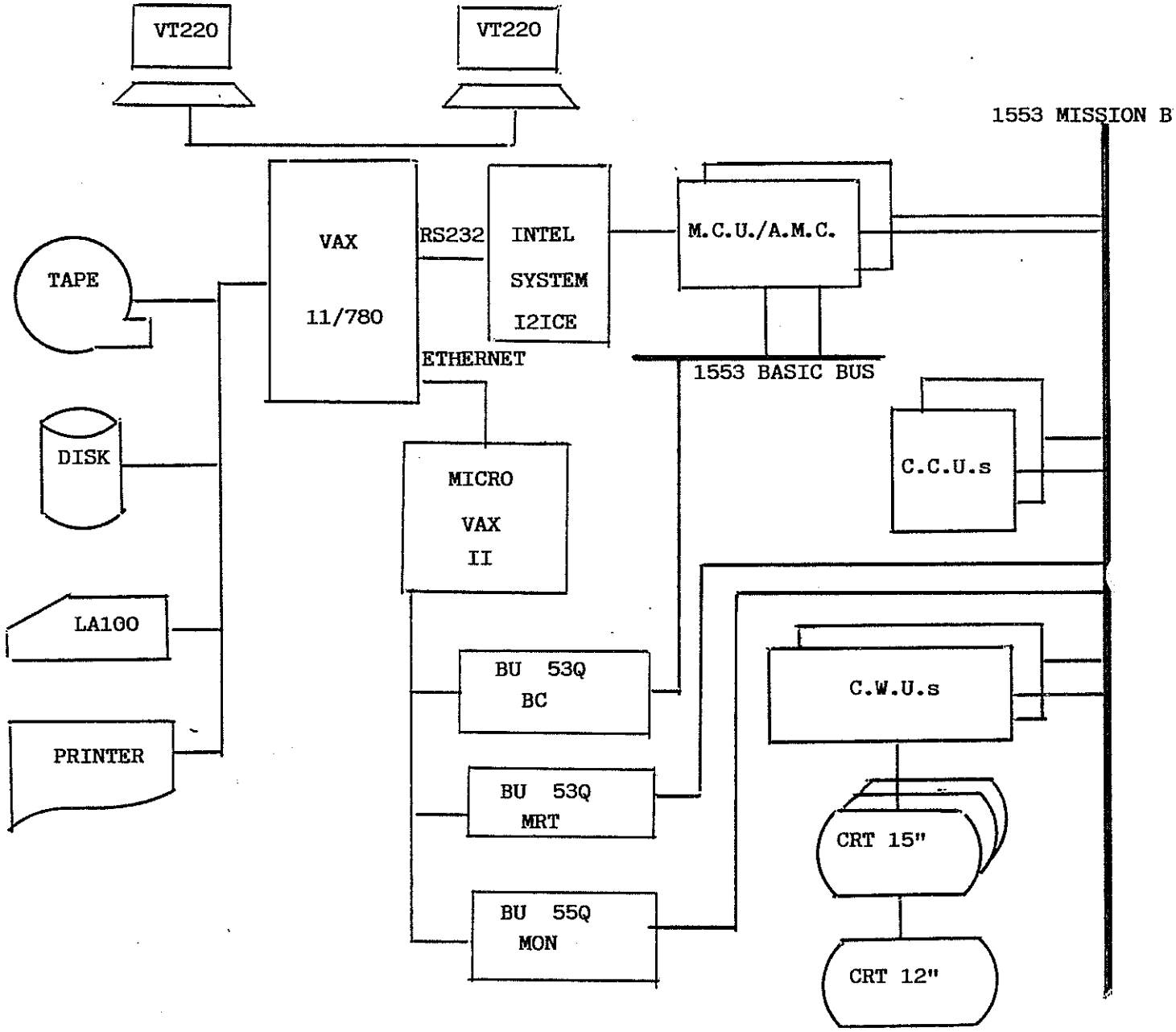
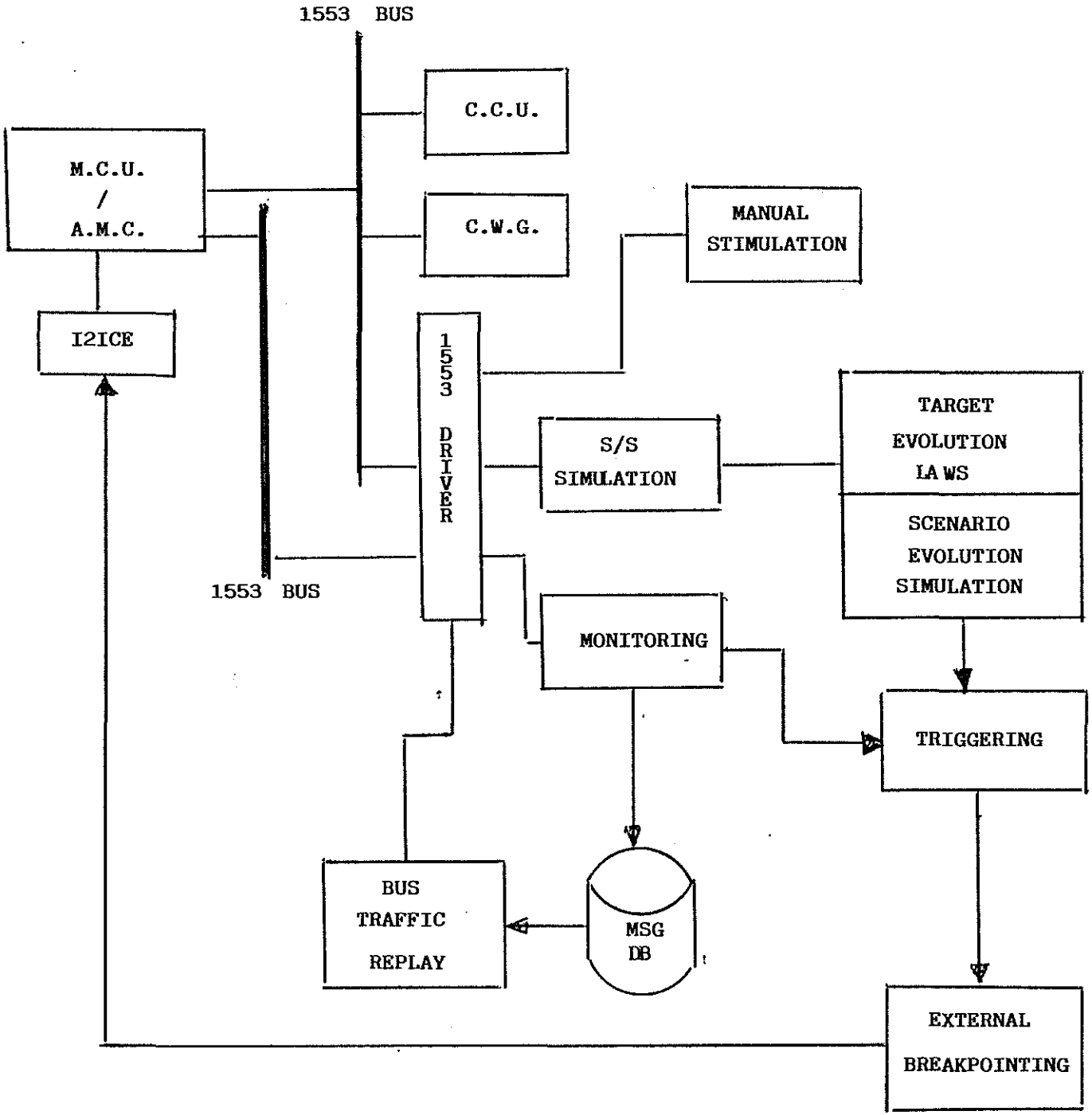
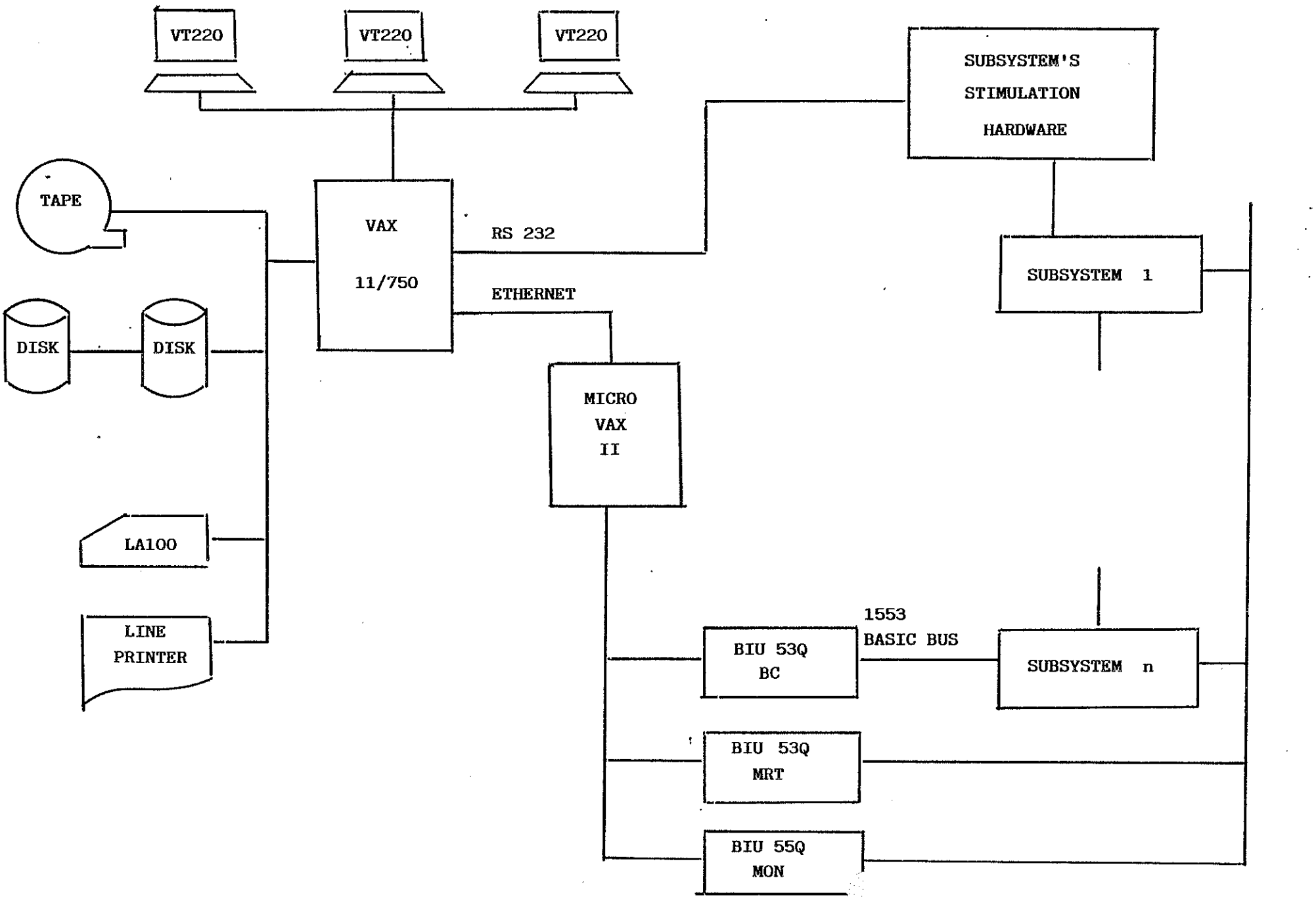


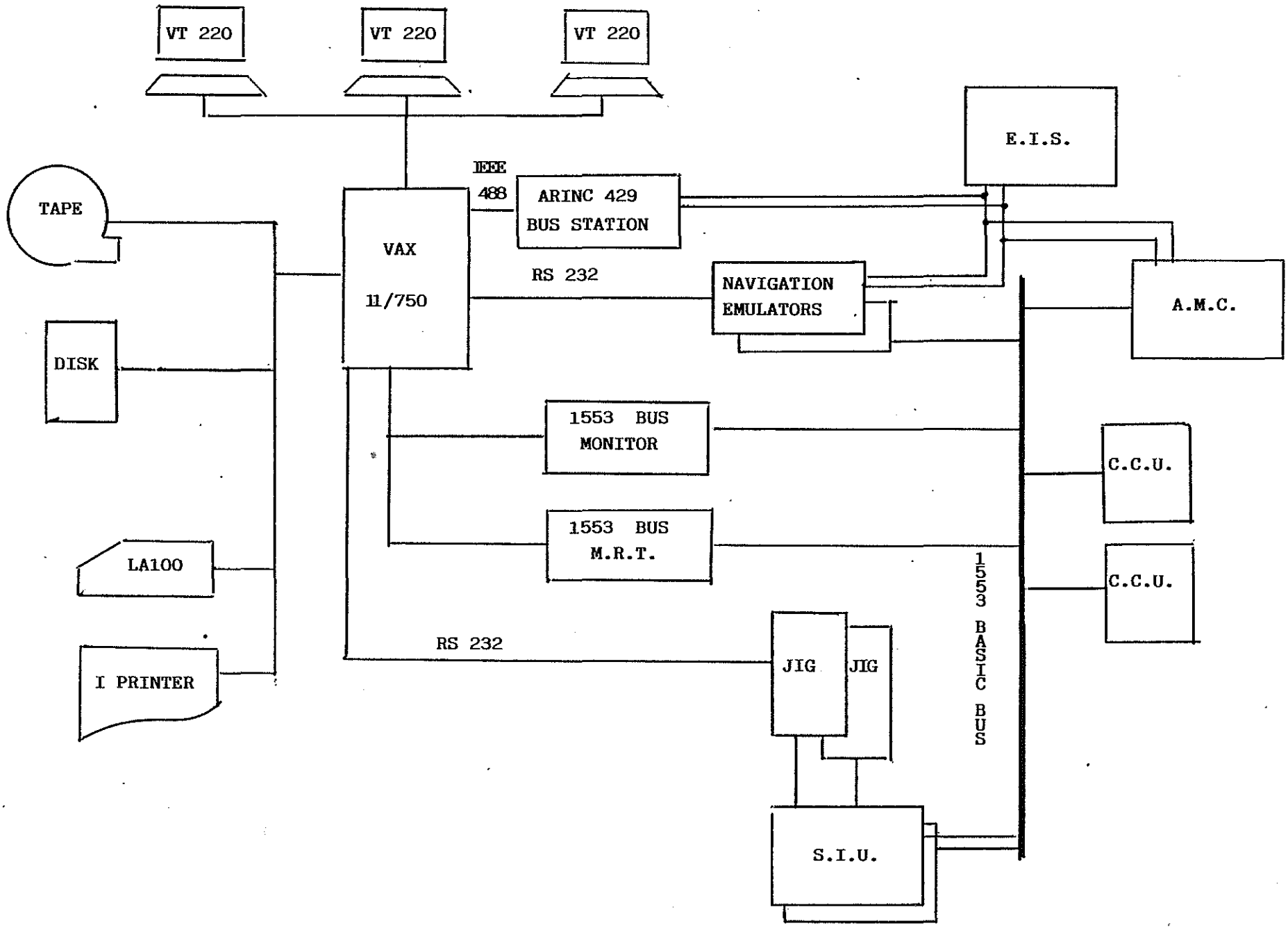
FIG. 2 - SOFTWARE DEVELOPMENT FACILITIES


1553 BUS
C.C.U.
M.C.U.
/
A.M.C.
C.W.G.
MANUAL
STIMULATION
I2ICE
1553 DRIVER
S/S
SIMULATION
TARGET
EVOLUTION
LAWS
SCENARIO
EVOLUTION
SIMULATION
1553 BUS
MONITORING
TRIGGERING
BUS
TRAFFIC
REPLAY
MSG
DB
EXTERNAL
BREAKPOINTING

MISSION INTEGRATION RIG



AMS RIG


VT 220
VT 220
VT 220
E.I.S.
IEEE
488
ARINC 429
BUS STATION
TAPE
VAX
11/750
RS 232
NAVIGATION
EMULATORS
A.M.C.
DISK
1553 BUS
MONITOR
C.C.U.
1553 BUS
M.R.T.
C.C.U.
LA100
1553 BASIC BUS
RS 232
JIG
JIG
I PRINTER
S.I.U.

OVERALL INTEGRATION AMS - MCS