TOWARDS AN INTEGRAL COMPUTER ENVIRONMENT SUPPORTING SYSTEM OPERATIONS ANALYSIS AND CONCEPTUAL DESIGN

E. Barro, A. Del Bufalo, F. Rossi
VITROCISET S.p.A.
Via Salaria 1027
00138 Roma - Italia

ABSTRACT

VITROCISET has in house developed a prototype tool named System Dynamic Analysis Environment (SDAE), which aim is to support system engineering activities in the initial definition phase of a complex space system.
The SDAE goal is to provide powerful means for the definition, analysis and trade-off of operations and design concepts for the space and ground elements involved in a mission.
For this purpose SDAE implements a dedicated modelling methodology based on the integration of different modern (static and dynamic) analysis and simulation techniques.
The resulting "system model" is capable of representing all the operational, functional and behavioural aspects of the system elements which are part of a mission.
The execution of customised model simulations enables:

- the validation of selected concepts w.r.t. mission requirements;
- the in-depth investigation of mission specific operational and / or architectural aspects;
- the early assessment of performances required by the system elements to cope with mission constraints and objectives.

Due to its characteristics, SDAE is particularly tailored for non conventional or highly complex systems, which require a great analysis effort in their early definition stages.
SDAE runs under PC-Windows and is currently used by VITROCISET system engineering group.
This paper describes the SDAE main features, showing some tool output examples.

1. INTRODUCTION

Modern space systems are evolving towards higher levels of complexity in both the functional and behavioural domain. This is a natural consequence of the increasing reliability of technologies based on intelligence and automation.
Spacecraft on board autonomy levels are progressively enhanced, and more "intelligent" and sophisticated operation control and support systems are conceived and developed.
Such a context demands for a complex engineering effort in the first phases of the system life cycle, when

- the suitable identification and / or selection of mission elements,
- the definition of system functions and functional sharing between elements,
- the establishment of a mission operations concept,
- the identification of system design and performance drivers,
- the validation of system conceptual definition w.r.t. mission objectives, requirements and constraints,

imply in depth analysis and trade-off among a wide scope of interdependent technology and implementation solutions.
The selection of an optimum mission configuration and operational strategy also affects heavily elements procurement or development and utilisation risks and costs.
In parallel with the evolution of space operations conduct and support technologies, it is therefore necessary to adequately improve engineering support aids to the conceptual design of the

mission and its constituting space and ground elements.
This can be achieved through extensive use of modern computer aided modelling and simulation methods and technologies.
VITROCISET is working since some years in this field, through:

- a methodological effort based on the definition of an integral modelling methodology for a complex system, capable to suitably support different kinds of representations (operational, functional, architectural) for conceptually different systems.
  Such a methodology has been derived by exploiting commonly adopted description, analysis and simulation synthaxes (e.g. OOA, SADT, Petri Nets).
- a development effort for the integration within a unique computer environment of system description and analysis capabilities, providing in this way the user with a single point of access to the whole system information, and means for information derivation, handling, consistency check and executable simulations preparation, execution and evaluation.
- an application effort, aimed at exploiting the computer environment capabilities in the frame of concrete projects and at deriving from the application experience requirements for environment upgrades.

System definition and analysis methodology has been already presented and discussed in precedent papers of the same Authors (Ref. 3, 5).
In parallel with the methodology development and refinement, VITROCISET has developed a PC based tool named System Dynamic Analysis Environment (SDAE), which has been progressively enriched in the last years up to covering with automated support a large part of the methodology characteristics.
The System Dynamic Analysis Environment finds its natural application in the fields of system operations analysis and systems engineering, in the frame of both high level (A and pre-B phases) studies related to satellite operations and in the system definition and design phase.
Currently, SDAE supports mainly the following activities:

- mission and system requirements definition and management;
- operations modelling;
- functional static and dynamic modelling;
- behavioural modelling;
- models parametrisation with operational and performance attributes derived from mission and / or system requirements;
- executable simulation and statistical evaluation of simulation results.

2. SDAE MAIN PRINCIPLES

SDAE tool is based on a layered modelling approach, depicted in figure 1.

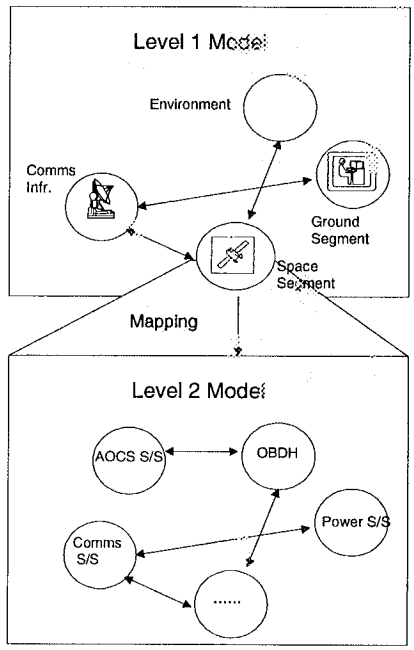

Figure 1: The layered Modelling Approach.

Each hierarchical layer is constituted by a set of models which structure and organise system information within well defined entities.
The scope and the purpose of the modelling activities vary according with the level of details of the system description.

On top layer, the entities managed by the tool are the main mission elements (physical or logical), such as the flight element(s) and its supporting ground facilities, or the spacecraft environment as well.
Entities can be functionally described as objects, in all those static and dynamic aspects which are of particular interest for the engineer in order to analyse a specific problem for the mission.
At this stage modelling supports initial mission analysis and operations concept definition activities, such as selection of mission support infrastructure, assessment of operational strategies and derivation of related design requirements and constraints.
A core modelling functionality enables the definition of dynamic relationships between objects (in terms of e.g. data exchange, events or dynamic modification of model parameters which affect objects behaviour).
Lower level models can be progressively defined for more specific analyses (e.g. command and control concept definition, budget analyses, element conceptual design and trade-off's).
The utilisation of a unique descriptive methodology at all the levels of details enables a straightforward traceability among the different modelling layers.
At bottom level, the tool can support the definition and description of end-to-end functional architecture models for the mission elements and their sub-components.
Any object at any level can be customised with characteristic parameters and reused in different contexts, even though at high level it constitutes only a partial view of the described element.
The execution of interactive simulations is therefore supported by a set of configurable library modules, including environmental models such as e.g. drag models and orbital propagators.
Simulation input parameters can be derived directly from associated requirements, as well as output parameters can be source for lower level requirements through dedicated derivation rules.

3. SDAE DESCRIPTION

SDAE tool provides the capability to build and execute dynamic operational, functional and behavioural models of a system, associating model parameters to mission or system requirements.
A high level architecture of the SDAE is provided in figure 2. Dotted lines in the figure show functionalities which are presently under development or test.

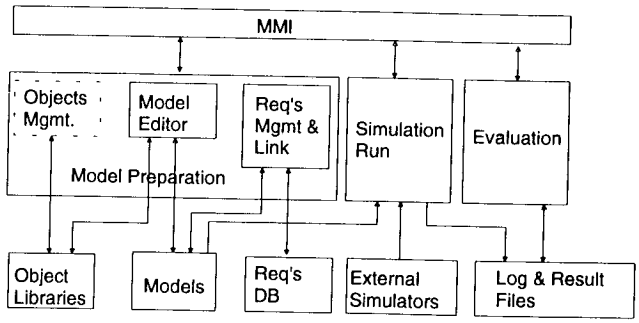

Figure 2: High Level SDAE Architecture.

The SDAE is constituted by three separate environments:
- Model Preparation;
- Simulation Run;
- Evaluation.

3.1 MODEL PREPARATION

Models are generated by means of:
- an object management facility (under development) for the static definition of basic model entities and their characterisation by means of a set of variables;
- a model editor facility for the end-to-end description of objects dynamic behaviour and relationships or interfaces;
- a requirements management and link facility for the models parametrisation with numeric parameters derived from mission or system requirements.

The model objects descriptions can be stored within object libraries and reused.
Models can also be interfaced at design time with external application specific simulation

libraries, with which they exchange data and status at run-time, providing in this way a realistic scenario for the simulation.

The **Model Editor** realises the core modelling functionality.

Such an editor is based on a Petri Nets-like synthax, and exploits a dedicated extension of Petri Nets methodology.

The editor enables the model dynamic specification through:

- a core state-transition network with deterministic and /or stochastic transitions;
- a predicates editor, which supports the definition of network predicates (conditions and actions) by means of a dedicated simulation language, and enables the model link with external simulation libraries.

The **Requirements Management and Link** facility enables the mission / system requirements handling, through:

- a requirements database editor;
- a linker between model variables and numeric requirements parameters, with possibility to specify input and output links, together with derivation rules for derived parameters;

The model preparation environment also enables the generation of ad-hoc panels for simulation monitor and control.

An example of SDAE preparation environment display output is provided in Figure 3.

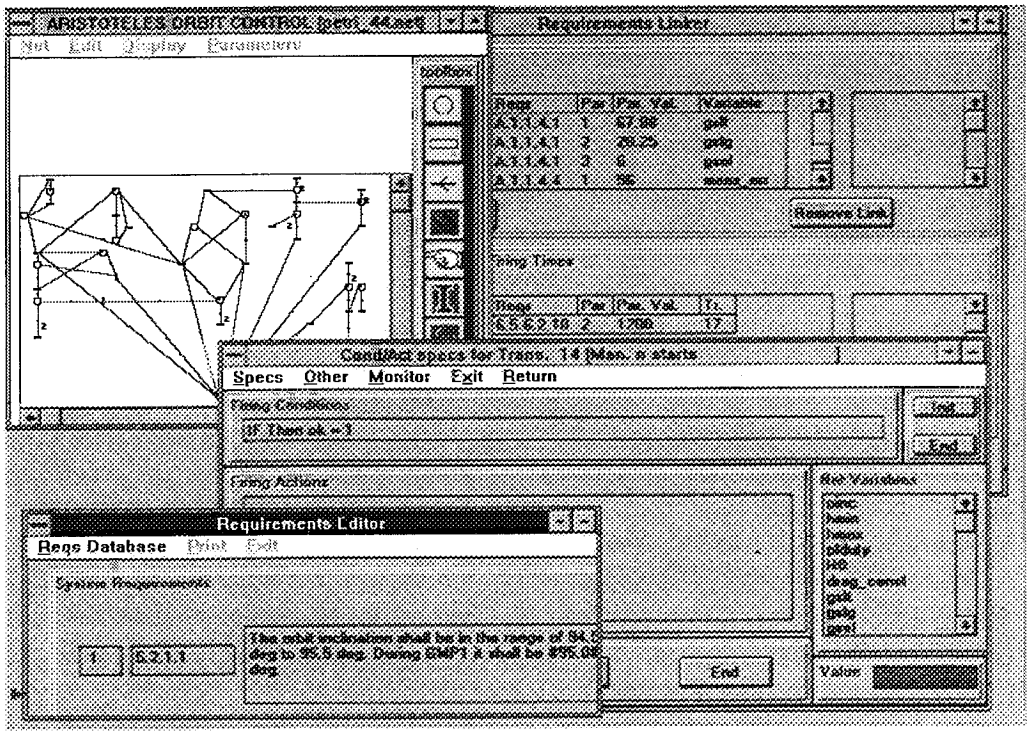

Figure 3: SDAE Model Preparation Environment.

3.2 SIMULATION RUN

Once the model has been generated, a simulation can be executed by means of the Simulation engine of the tool.

The simulation execution environment allows:

- initialisation of simulation parameters (e.g. duration, step) and variables;
- three different modes of simulation:
  - batch (the model works stand-alone with user interface);

- step by step (the model stops in case of firing conflicts in order to highlight decision branches in system behaviour);
- debugging (the user decides which transition shall fire, among those enabled, in order to experiment predefined behavioural paths);

- capability to stop, continue or restart a simulation with the same or different initial conditions;
- user interaction in batch mode, by means of monitoring and controlling the model through customised control panels defined at design time;
- simulation history log;
- on-line display of simulation statistics.

During the simulation, the run module executes the model syntax, interfacing with external simulation software.

The capability of defining firing conditions for the network transitions enables the implementation of priorities, in case the modelled process is fully deterministic, i.e. no resource conflict between concurrent functions is allowed.

The definition of transitions associated actions enables the parametrisation of network tokens, modelling in this way the availability of different kind of resources within the system.

Examples of simulation execution environment display outputs are shown in Figures 4 and 5.

The shown examples reflect different simulation and design objectives, as pertaining to different stages of system life cycle.

The application shown in Figure 4 has been developed within ESA/Dornier ARISTOTELES Phase A and Pre-B studies.

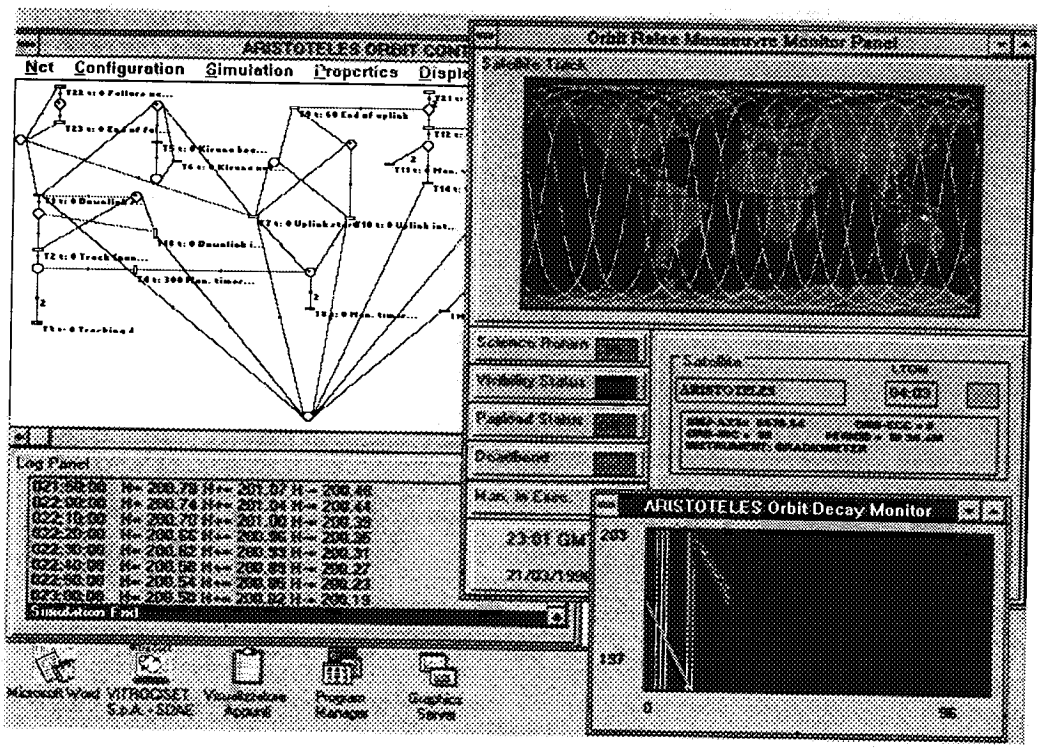

Figure 4: Simulation output example: ARISTOTELES ORM Analysis.

It constitutes the modelling of a spacecraft operational process, the Orbit Raise Manoeuvres (ORM) execution process, which involves ground, spacecraft and environmental functions.

The overall objective of the study was the definition of an optimum strategy for satellite tracking and ORM execution, identifying the impacts of the selected strategy onto the flight element and ground segment architecture.

In particular the following topics were addressed by the study:

- define the on board autonomy level, working on the flexibility of the mission;
- identify a safe orbit maintenance manoeuvre sequence;
- ensure required scientific return from the system operations viewpoint;
- identify the interrelationship of chosen coverage, link budget and memory budget with the selected operational strategy;
- validate the sequence of events in the operational scenario;
- analyse consequences of failure on the chosen design (e.g. redundancy philosophy).

Figure 4 shows:

- the model of ORM process within the Simulation Run Environment display screen;
- the ORM monitor panel, including an orbital propagator (external module) outputs and significant simulation variables monitoring;
- the Altitude display panel with an atmospheric drag model (external module) output;
- the log display of satellite contacts with Kiruna Ground Station, as computed by the orbital propagator.

The execution of the ORM process model for different initial conditions and environmental conditions (contact failures scenario) has enabled the selection and validation of an operations strategy, which satisfied all the system requirements in the defined worst case conditions.

The model has also been exploited as a breadboard of the process under study, deriving and verifying quantitative parameters determining the sensitivity of the strategy (and therefore strategy failure conditions) to the variation of any of the parameters of the model, like e.g. the spacecraft decay rate or the altitude determination errors, with respect to the reference values.

A wide number of statistical results about the process under study has been derived, as the time distribution of manoeuvres intervals and of manoeuvres size, the deadband utilisation figure, the scientific return distribution.

Finally, concrete impacts on the space and ground architecture have been identified on the basis of simulation results, especially with respect to On Board Data Handling System (in terms e.g. of definition of autonomous functions, sizing of mass memory required for manoeuvres parameters storage) and Ground Station architecture (e.g. need for a dedicated ground station, which has been derived as an "a posteriori" constraint for successful exploitation of ORM strategy).

The application shown in Figure 5 has been developed in the frame of ESA/SAT CONTROL Hermes Board Observability Breadboard (BOB) software project.

The BOB is a spacecraft simulator which models the generation and downlink of Hermes telemetry, with the scope limited to Guidance, Navigation and Piloting (GNP) functions.

The objective of the BOB is to provide a mean (breadboard) for the definition of an optimum telemetry strategy, and the verification of how this strategy copes with spacecraft observability requirements.

In this context, VITROCISET has been responsible for the definition and development of the on board Telemetry Generation Assembly simulator, which reproduces the generation of CCSDS telemetry packets on the basis of on board events and operator directives, and their delivery to Communications subsystem for downlink.

The Telemetry Generator Assembly (TGA) was designed with the SDAE simulation support.

A behavioural model of the assembly was generated and executed, in order to validate system behaviour w.r.t. specifications, to experiment different implementation solutions and to derive performance objectives for the software modules in order to cope with system requirements.

The model was able of fully reproducing the system behaviour, including partial modelling of hardware equipment (disk driver, buffers).

As an example, the model reproduced the following characteristics:

- packet generation directives acceptance and rejection policy (including input data format

and parameters check and consistency check with current packet generation status) and related timing;

- directives processing operations;
- directives scheduling policy (e.g. insertion/deletion/update of schedule items, schedule execution tasks "jumping" in case of critical delays) and related timing;
- internal synchronisation and priorities (e.g. enabling / disabling of packet playback on the basis of schedule status, blocking and non blocking operations, internal overrides);
- packet generation policy (e.g. handling of measurement variations occurred during the generation of a packet, generation policy of supercommutated packets).

The model accepted as an input a timeline of telemetry generation directives, and enabled the operator interaction by means of issuing at any time new directives for the model. The output of the model was a list of generated packets, with:

- packet generation and delivery times;
- list of included measurements and related values.

The time resolution of the simulation was chosen of 1 millisecond.

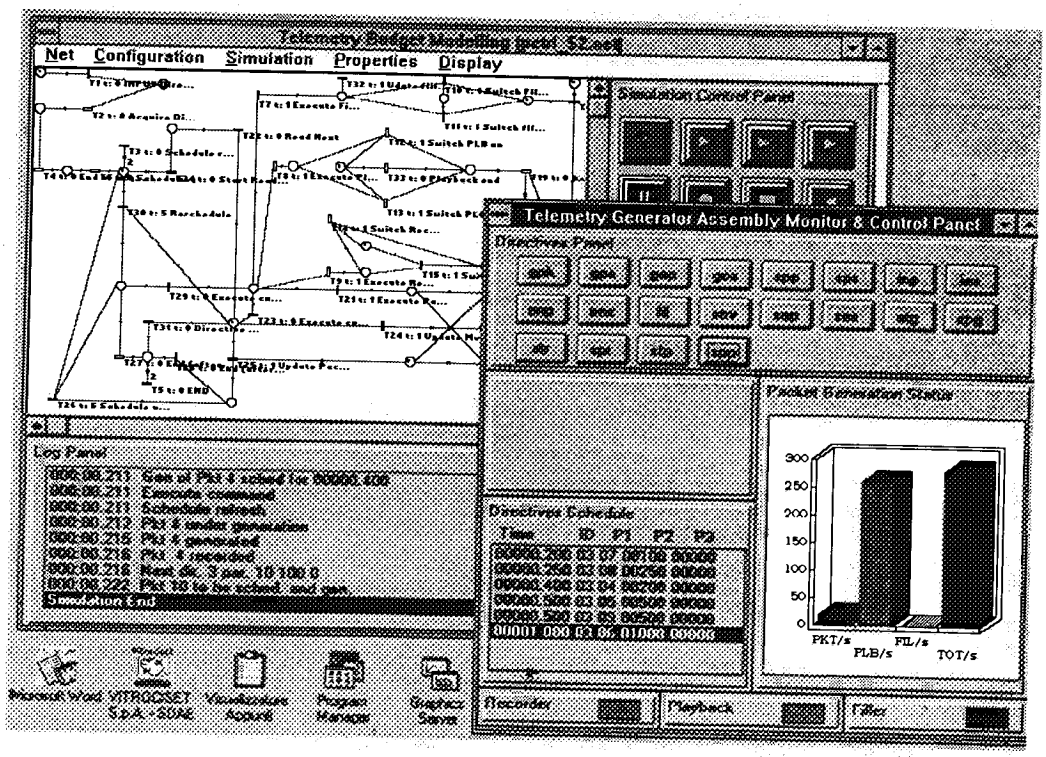

Figure 5: Simulation output example: BOB TGA Architectural Design.

Figure 5 shows:

- the process model within the Simulation Run Environment display screen;
- the test operator monitor and control panel, including:
  - directives panel for the generation of telemetry generation directives by the operator;
  - packet generation status monitoring panel;
  - current system schedule;
  - status of the main system functions.

The execution of the TGA behavioural model provided the designer with a lot of information on the system. In particular different scheduling policies and packet generation policies have been tested before selecting the one which optimised system functioning under nominal and peak load conditions.

Even though the model was at behavioural level, inferences on system performances have been

derived by setting and changing maximum allowed times for software tasks execution, and deriving in this way objectives to be pursued in single functions implementation in order to meet the overall system performances.

In depth analysis of deadlock conditions has been performed, by means of identifying and quantifying the relationship between the input data rate and the system response, which under critical conditions is characterised by a degradation in performances due to the skipping of packet generation tasks in order to avoid propagation of delay with respect to the schedule.

In addition, the system response under different modes of functioning (e.g. recorder, playback, filler activated / deactivated with a predefined rate) allowed the determination of packet generation rate achievable in the different modes, deriving in this way differentiated constraints for packet generation function.

Finally, the partial modelling of some significant time consuming hardware functions (access to disk, input/output operations) enabled the assessment of limits imposed by the hardware onto system performances.

3.3 EVALUATION.

After the simulation run, the log file is processed by an **Evaluation** environment, which computes and displays the main network statistics, i.e. for each transition:

- overall number of firings;
- minimum, average and maximum time between two successive firings.

The environment also supports the generation of customised graphical reports by means of interface with standard Windows facilities and the processing of the log file, providing statistical figures of predefined network parameters and variables (e.g. distribution of parameters values across the simulation).

An example of Evaluation Environment screen layout is provided in Figure 6, representing the ARISTOTELES ORM process model simulation statistics.

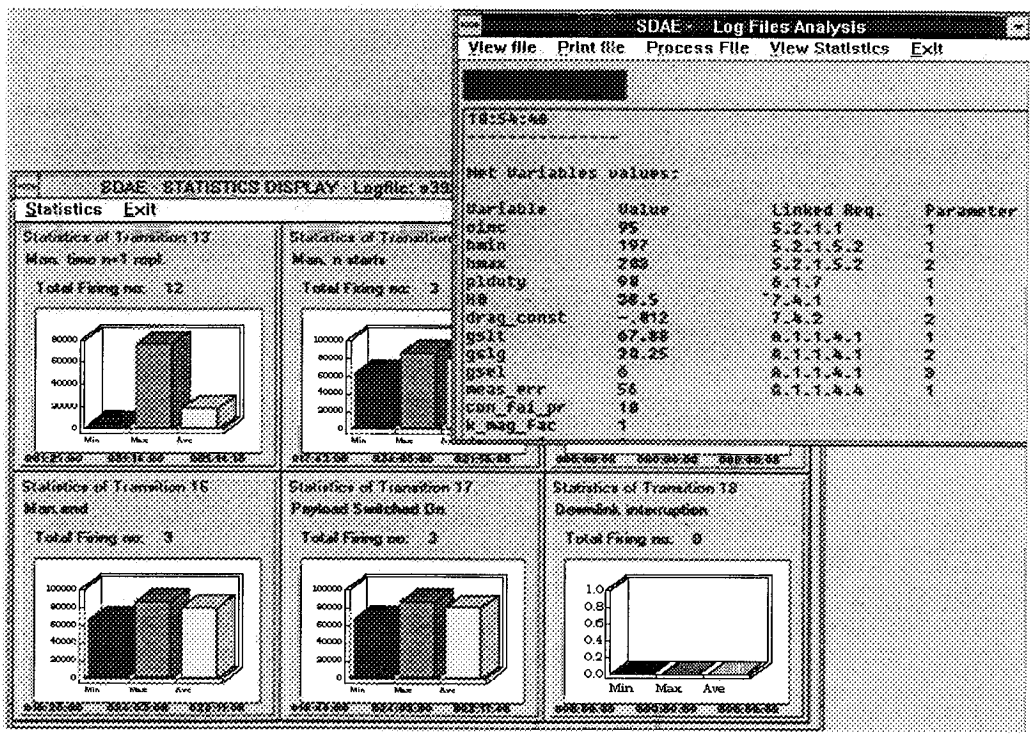

Figure 6: Evaluation Environment output example: ARISTOTELES ORM Analysis.

4. CONCLUSIONS

SDAE prototype implementation has been originated with the purpose of investigating the system engineering process of a modern space system in the first phases of its life cycle.
In particular, the ultimate objectives of the tool were:

1. to provide an efficient breadboard for testing "on the job", within limited implementation costs and effort, methodologies aimed at:
   - ensuring a harmonic and consistent growth of system information in this phase;
   - empowering system analysis and validation capabilities, especially for highly automated or non procedural systems.
2. to derive requirements for methodologies assessment and refinement, on the basis of concrete engineering needs outcoming from the tool application experience.

SDAE application has resulted to effectively support both system analysis and conceptual design, lowering the engineering effort for the execution of operations analyses and architectural trade-off's and providing, by means of simulation, significant support to operations and system concepts validation capabilities.
In particular the following characteristics of the prototype have been found of particular interest, especially in comparison with engineering tools available on the market:

- the flexibility of modelling methodology, which enables the easy generation and maintenance of "on purpose" models, without constraining the engineer to rigorous top-down approaches, but at the same time providing capabilities for system information consistency keeping;
- the adequacy of modelling and simulation tools to non procedural, event drive systems;
- the reusability of model objects and simulation modules;
- the "live dialog" capability of system models with mission and system requirements parameters through numeric data exchange and derivation rules, which highly enhance ability to manage, control and validate system information.

Those positive outcomes suggested the prosecution of the internally funded SDAE prototyping activity, which currently is being performed in the direction of both:

- improvement of tool modelling powerfulness and engineering support scope;
- increase of tool application experience, through the investigation of new application areas, such as communications and ground data control and distribution systems.

REFERENCES

1. Agerwala T. 1979. Putting Petri Nets to work. Computer, Dec. '89, 85-94
2. CISET. ARISTOTELES Phase Pre-B Study: Satellite Autonomy. Ref. ARI-TN-CI-001, Issue 1.1, April 3rd, 1992
3. E. Barro & F. Rossi. An Application of Timed Petri Nets to Operations Analysis: the ARISTOTELES Autonomy Concept. In *Proc. ESA Symp. "Ground Data Systems for Spacecraft Control"*, Darmstadt, FRG, ESA SP-308, 317-322.
4. SAT CONTROL. BOB Software and Hardware Architecture Description. H-NT-01210-0286-SATC, Issue V0, September 8th, 1992.
5. E. Barro, A. Del Bufalo & F. Rossi. Operational Characterisation of Requirements and Early Validation Environment for High Demanding Space Systems. In *Proc. NASA 2nd Symp. "Ground Data Systems for Space Mission Operations"*, Pasadena, California, USA, November 16-20 1992, 845-850.