111157 **N95-17572**

3/977
p. 6

Evaluating Modeling Tools for the EDOS

Gordon Knoble and Frederick McCaleb
NASA GSFC Code 560
Greenbelt, MD 20771

Tanweer Aslam and Paul Nester
Computer Sciences Corporation
7700 Hubble Space Dr. Lanham, MD 20706

ABSTRACT

The Earth Observing System (EOS) Data and Operations System (EDOS) Project is developing a functional, system performance model to support the system implementation phase of the EDOS which is being designed and built by the Goddard Space Flight Center (GSFC). The EDOS Project will use modeling to meet two key objectives:
(1) Manage system design impacts introduced by unplanned changes in mission requirements and (2) evaluate evolutionary technology insertions throughout the development of the EDOS. To select a suitable modeling tool, the EDOS modeling team developed an approach for evaluating modeling tools and languages by deriving evaluation criteria from both the EDOS modeling requirements and the development plan. Essential and optional features for an appropriate modeling tool were identified and compared with known capabilities of several modeling tools. Vendors were also provided the opportunity to model a representative EDOS processing function to demonstrate the applicability of their modeling tool to the EDOS modeling requirements.

This paper emphasizes the importance of using a well defined approach for evaluating tools to model complex systems like the EDOS. The results of this evaluation study do not in any way signify the superiority of any one modeling tool since the results will vary with the specific modeling requirements of each project.

INTRODUCTION

A set of criteria specific to EDOS modeling requirements was developed for evaluating and selecting the most suitable modeling tool. These criteria identified potential strengths and weaknesses of modeling tools which would affect the EDOS model development time, enabling the team to initially screen each product prior to evaluating its capabilities in detail. This approach ensured timely adjustments to the overall EDOS modeling plan based on manpower estimates for implementing a useful EDOS model with the chosen tool.

The EDOS modeling tool evaluation criteria were divided into two categories, essential and optional. Essential criteria (e.g., modeling of high data rates) identified the modeling tools which could satisfactorily support the development of the EDOS model. Optional criteria (e.g., model software configuration management support) were used to identify modeling tool features which could aid in developing and operating the EDOS model by its users. A ranking and weighting scheme enhanced the evaluation process further, ensuring that major differences between modeling tools were well understood by the modeling team. The evaluation approach was even further refined by requesting each prospective vendor to develop a sample model of a representative EDOS function and demonstrate the tool capabilities considered critical for developing the EDOS model. These demonstrations provided additional modeling tool discriminators, improving the team's understanding of the tool capabilities and enabling them to adjust the evaluation scores accordingly. A detailed matrix of evaluation results was developed on an EXCEL_{TM} spreadsheet.

Major categories of the evaluation criteria included: Simulation data collection and generation of results, ease of model development, architecture representations (e.g., hardware, software, and data), user interface, additional development effort (necessary to compensate for modeling tool limitations and meet satisfactory requirements), model execution control, tool reliability, model platform choices and execution speed, documentation and training, vendor support, and portability of developed models. Additional criteria included modeling tool licensing and training costs, annual maintenance fees, and inherent risks (e.g., tool immaturity). The modeling tool with the best combination of evaluation score, least additional manhours estimated, and least implementation risk was selected as the most suitable tool for modeling the EDOS. If the evaluation results in more than one technically compliant candidate, then cost may well become the major deciding factor in the selection process.

The NASA / CSC EDOS modeling team consisted of experienced system engineers, each with at least ten years of experience in developing functional system performance models on various projects. Because of their current knowledge in the modeling field, they were readily able to identify a number of potential candidates for modeling the EDOS. Seven modeling packages and two modeling languages were identified as potential candidates. These were either commercial-off-the-shelf (COTS) items or available through NASA GSFC. Table 1 lists the candidate modeling packages and languages, in alphabetical order.

Table 1: Candidate Modeling Tools for EDOS

Item	Candidate Modeling Tool Name	Type	Developed by
1	Block Oriented Network Simulator (BONeS)	Package	Comdisco, Inc.
2	COMNET III	Package	CACI, Inc.
3	Data System Dynamic Simulator (DSDS)	Package	GSFC/STEL, Inc.
4	Extendible Computer System Simulator (ECSS)	Language	NTIS
5	General Purpose Simulation System (GPSS V)	Language	IBM
6	L•NET and NETWORK II.5	Package	CACI, Inc.
7	OPNET	Package	MIL3, Inc.
8	Quantitative computer Assisted System Evaluator for Reliability and Timing (QASE RT)	Package	AST, Inc.
9	SLAMSYSTEM 2.0	Package	Pritsker Corp.

EVALUATION APPROACH

The EDOS modeling team developed a well defined, structured approach to evaluate modeling tools, consisting of the following activities:

- Defining evaluation criteria
- Identifying available modeling tools
- Screening modeling tools against essential criteria
- Evaluating modeling tools in detail
- Requesting vendors to model a sample processing function
- Selecting the most suitable modeling tool for EDOS

Defining Evaluation Criteria

The EDOS modeling requirements document and the EDOS modeling plan were used in identifying and defining a uniform set of evaluation criteria for modeling tool packages and languages. A total of 12 evaluation categories (EC) consisting of 101 essential features and 24 optional features were identified. The categories are listed in Table 2.

Identifying Available Modeling Tools

Identifying suitable modeling packages and languages as potential candidates for modeling the EDOS was the second step. The experience of the modeling team members, as well as a search of available literature, produced several candidates. This was not intended to be an exhaustive search and many packages were not identified simply due to the lack of available time.

Screening Modeling Tools against Essential Criteria

All candidate modeling tools were evaluated against the essential criteria. After an initial screening, several modeling packages and languages designed for specialized applications (such as packet switching) were clearly not suitable for modeling the EDOS and were rejected from further consideration.

Detailed Evaluation of Modeling Tools

The detailed evaluation assessed the capabilities of each modeling tool qualitatively. The following scoring scheme was used in the detailed evaluations.

Scoring Scheme

A scoring scheme, ranging from "0" to "5", was used to evaluate the modeling tools in detail:

0: The modeling tool has no capability (fail).

1: Only minimal (poor) capability is provided, requiring extensive work to overcome the problem. The additional effort was estimated and included in the detailed evaluation matrix.

2: The capability is less than satisfactory (fair), requiring some work compensate for the deficiency. The additional effort was estimated and included in the detailed evaluation matrix.

3: The tool provides a satisfactory (average) capability.

4: The tool provides more than a satisfactory (good) capability.

5: The tool provides an excellent capability.

Table 2: EDOS Modeling Tools Evaluation Criteria

EVALUATION CRITERIA	Evaluation Category / Weight	Examples of ESSENTIAL FEATURES	Examples of OPTIONAL FEATURES
Data Collection and Output Generation (e.g. flexibility of selecting measurement points and accuracy of results)	12	Self contained production of output results, can produce trace, snapshots, summary and periodic reports	Can transfer results to analytical tools
Ease of Development (e.g. built-in features, formulae and predefined elements)	11	Predefined elements for statistics collection and measurement	Able to automatically verify model completeness
Architecture Representation (e.g. representations of data, time, functions, HW and SW elements and/or resources)	10	Accepts several sources of stimuli	Support modeling of functions independent of system specific H/W and S/W design
User Interface	9	Display/print simulation configurations, input parameters, and results	Tool user's interface (ICONS, GUI, etc. are utilized)
Model Modification (e.g. modification of parameters and configuration of model resources)	8	Ability to propagate parameters thoughout model structure	*(None were identified)*
Development Effort/ Schedule	7	Support implementation in 6 months	*(None were identified)*
Execution Control (e.g. flexibility of simulation control, enabling and disabling of functions and resources)	6	Single functions, grouped functions, and message logging	*(None were identified)*
Tool Maturity (Trusted by users)	5	Tool is mature, field usage is greater than 1 yr	Number of copies sold/licensed to users
Platform and (Run Time e.g. flexibility of choosing and upgrading a platform)	4	*(None were identified)*	Flexibility of selecting modeling platforms (machines supported: PC(DOS), Macintosh, Windows, OS/2, UNIX, etc.)
Documentation and Training	3	User manuals and training courses	*(None were identified)*
Vendor Support (e.g. during sample problem modeling and future commitments)	2	Availability of product support personnel	*(None were identified)*
Developed Model Portability (e.g. due to model growth and platform upgrade)	1	*(None were identified)*	Developed Model Portability to support growth

Assessment of Additional Development Effort

Modeling tool capabilities earning a score of 1 or 2 were considered deficient. The EDOS modeling team carefully reviewed these deficiencies and assessed the feasibility of correcting them with additional development effort. Previous model development experience with similar modeling packages and languages aided in assessing the number of manhours required to compensate for any shortcomings. Consulting with modeling tool vendors also aided in arriving at the most conservative estimates for correcting the deficiencies, if possible.

Modeling of a Sample Processing Function

This step of the evaluation approach was invaluable in the selection process. The EDOS modeling team prepared a sample modeling problem, generic in nature, representing an aggregation of typical processing functions required for EDOS. Each modeling tool vendor was asked to use the sample processing function to prepare a sample model, without cost to the project, to demonstrate the capabilities of their tool in support of the evaluation. Four vendors chose to model the sample processing function free of charge to demonstrate the capabilities of their tools; two did not (three did not pass the initial screening). Models of the sample function were not developed with modeling languages because of the extensive effort required by CSC personnel. There were no disqualifications of modeling package or modeling language vendors if they chose not to develop and demonstrate the sample processing function model. However, the demonstrations of the sample model enabled the EDOS modeling team to accurate assess the capabilities of those vendors' modeling tools.

Selection of the Most Suitable Modeling Tool for EDOS

All modeling tools meeting all essential criteria participated in this final evaluation activity. The following steps were used to identify the most suitable modeling tool for EDOS:

a. The total score for each modeling tool was calculated by adding all scores for each evaluation category (a total of 12).

b. The total effective cost for each modeling tool was calculated by adding modeling tool software cost, training cost, and cost for maintaining the tool for four years.

c. The total additional development effort required to compensate for deficiencies of a modeling tool and to improve its performance to a satisfactory level was calculated.

d. A risk factor (low, medium and high) for each modeling tool was assessed based on the results of detailed evaluation and the amount of additional development effort (manhours) required to improve the tool performance to a satisfactory level.

e. The modeling tool with the best combination of detailed evaluation score, lowest manhours for additional development effort, and least implementation risk was selected as the most suitable tool for modeling the EDOS.

SUMMARY OF EVALUATION RESULTS

Of the nine candidate modeling tools, only six: BONeS, DSDS+, OPNET, QASE RT, ECSS II, and GPSS V were fully evaluated. The development manhour estimates for the two modeling languages, ECSS II and GPSS V were beyond the scope of the modeling schedule. Of the remaining four modeling packages, DSDS+ and QASE RT were chosen as the most cost effective modeling tools which meet or exceed the EDOS modeling evaluation criteria. The data stream feature of DSDS+ enables modeling of scenarios spanning several days and weeks. The separate HW and SW architecture components of QASE RT provide a more realistic, graphical representation of the EDOS.

LESSONS LEARNED

The following key lessons were learned while evaluating the modeling tools for EDOS:

1. The modeling tool criteria should be developed from the modeling requirements and objectives specified for a candidate system. Therefore, system requirements and plans describing the modeling objectives should be complete before defining the modeling tool evaluation criteria.

2. The modeling tool evaluation criteria should carefully distinguish the essential and optional features considered. Non-critical requirements having little impact on the system development must not be allowed to influence the modeling tool selection.

3. Predetermining optional features desired can prevent the evaluation process from being misled by a single interesting aspect of a modeling tool. Several tools had spectacular features which, while very impressive, were not applicable.

4. Vendor development of a sample model of a representative system function to demonstrate the real strengths and weaknesses of a modeling tool can ease the completion of the modeling tool evaluation work in a single demonstration session.

5. The best results are achieved by team evaluation of modeling tool capabilities, which aids in balancing any bias.

6. There is no perfect modeling tool for any system. Use of additional effort, if not major, should not be overlooked for overcoming minor deficiencies of an otherwise robust modeling tool before eliminating it from further consideration.

7. The number of discrete events required for modeling a function has an extremely detrimental effect on the runtime ratio between simulated time and real time, due mainly to the exceptionally high packet rates. While this risk is dependent upon the speed of the platform selected, ways should be investigated early on to minimize it by properly designing the model's structure.