AN EXPERIENCE WITH THE USE OF SYSTEMS ENGINEER CASE TOOL

MÁRIA BIELIKOVÁ and PAVOL NÁVRAT
Slovak Technical University, Dept. of Computer Science and Engineering,
Ilkovičova 3, 812 19 Bratislava, Slovakia
E-mail: {bielikova,navrat}@elf.stuba.sk

Systems Engineer is a client/server CASE tool for supporting software system development on PC machines under Windows. It compares fairly satisfactorily with other similar tools of the same category. Systems Engineer offers various supporting techniques which address the full software life cycle, from planning and design to code generation, reengineering and maintenance. It supports system engineers working in a team.

Keywords: software system analysis, software system design, Systems Engineer, CASE tool

1 Introduction

Systems Engineer [4] is one of the numerous CASE tools that are currently available for supporting software system development on PC machines under Windows. Our experience with using it comes from two sources:

- industrial applications, most notably system analysis and design of an integrated information system for Slovak Telecom, which is a large, nationwide geographically distributed company acting as the sole national telecommunications operator,
- educational process, where our Informatics majors specializing for Software Engineering are trained to be able to use the tool, and are assigned projects to be completed within one semester time.

In this article, we shall review the tool on the following criteria: (1) scope of system analysis and design methods that are directly supported, (2) ability in supporting a software developer in the intended application, (3) quality of user interfaces and user manuals, (4) ability to integrate with other tools. We conclude with some comments that generalize our experience. Also, we compare Systems Engineer with other tools with similar objectives.

2 Scope of support

Systems Engineer Release 6.1 is a comprehensive client/server CASE tool. It claims to address the full life cycle, from planning and design to code generation, re-engineering, and maintenance.

The crucial word is "to address", as it can be interpreted more loosely than "to support", not even speaking about "to automate". On the other hand, it is only fair to state that it indeed is capable of supporting the software engineers working in a team in developing an application during the system's life cycle.

Systems Engineer supports several analysis and design techniques with the graphical editors and many types of forms which allow users to associate information with symbols in diagrams. They are based on LBMS Systems Engineering methodology [3] which employs mainly structured techniques.
Techniques that are supported include:

- *process oriented* techniques with Data Flow diagrams, Entity Life History diagrams, forms for Functions, Events and Transactions;
- *data oriented* techniques with Data Model diagrams which represent the structure of the data in the business or application area. The tool can be used to generate the database schema from the components of the physical data model. One can create the design by self or by reversing an existing system;
- *user interface* design with User Object modelling, and Transaction Dialog Structure diagrams. They serve for specification how the users will interact with the on-line parts of the system together with Menu Control Structure diagrams and Screen design where prototyping is allowed;
- *general* techniques with forms for Problems, Requirements and Solutions, General Picture diagrams and General Forms.

We were using the tool during analysis and design of a large information system [1]. We were also using it to design user/system dialogues and to prototype them.

Systems Engineer can be complemented by other tools. Process Engineer is another tool of the same vendor. It is devoted to supporting monitoring, controlling, and improving the software development process. It comprises process library, process manager, project manager, and activity manager.

3 Ability to support

When considering a degree of control over the compliance of the product development process with a chosen methodology, Systems Engineer ranks among the more flexible CASE tools. It permits the developer to exercise full control over the development process [5]. Checking for hierarchical consistency occurs on request.

The tool supports data and process modelling by one chosen formalism, and no switching between alternative formalisms is available. In data modelling, one can use with advantage a possibility to split large models into smaller logical groupings known as data model subsets. Processes can be decomposed, but data stores cannot. Data flows decomposition is based on the names of particular data flows.

Its client/server architecture supports a development in workgroups. In evaluating workgroup development, we were inspired by Vessey's and Sravanapudi's model of coordination support [5]. The model is based on requirements needed for *control*, i.e. the ability to provide different levels of access rights to different levels of group members, *information sharing*, i.e. the principal means a workgroup uses to communicate when it is concerned with product development, and *monitoring*, i.e. tracking different aspects (e.g. changes) of both the product and the user.

Now, we briefly describe features which Systems Engineer supports together with the total number of observed features according to the above model (in brackets):

- **Control**

  Acces Control (7 features)

  - ac-2. One can specify different access rights to different teams
  - ac-3. One can specify different access rights to different team members
  - ac-6. One analyst can have read-only access to another analyst's work

- **Information Sharing**

  Concurrency Control (5 features)

  - cc-1. It is possible to access the data dictionary concurrently
  - cc-2. The tool provides a mechanism to control concurrent access to the data dictionary
  - cc-3. The tool provides diagram-level locking in the case of multiuser access to the dictionary
  - cc-5. One can find out who has locked the item of interest

  Data Sharing (5 features)

  - ds-1. Possibility to simultaneously display the same diagram on different workstations
  - ds-3. Possibility to share the data in the data dictionary across projects
  - ds-4. Possibility to make one's work available to others without validation

  Consistency Enforcement (7 features)

  - ce-6. The tool automatically notifies an analyst whose work might be affected due to a change in the data dictionary
  - ce-7. It is possible to freeze parts of the development work to protect it from changes

- **Monitoring**

  Product (6 features)

  - pm-1. Possibility to obtain information on who created any given object in the data dictionary and when
  - pm-3. For any given object in the data dictionary, it is possible obtain information on who made the last change and when
  - pm-6. The tool can generate reports on every reference to an object in the dictionary

  User (2 features) - no support

In comparison with CASE tools evaluated in [5], namely Deft 4.0, Iconix 4.0, System Architect 2 and Visible Analyst 3.0 where range of features per tool was from 9 (Deft) to 17 (Iconix) out of the total 32 features, Systems Engineer is fully competitive with them with its total of 15 supported features.

Communication between software developers and customers is also supported by high quality textual representations through tools for specification and formatting of data. Document preparation in Systems Engineer is an easy process with the "point & click" user interface. When a document is produced, current data dictionary information is extracted ensuring that documentation is always consistent with the development.

Moreover, a developer can use standard reports about objects in the repository, which include the object's properties and associations.

4 Quality of user support

Systems Engineer's user is supported by two principal means: written guides, and an on-line help. There is in fact just one basic guide aimed at the user of Systems Engineer i.e., the Concepts Guide. The material describes concepts in a fairly satisfactory way. However, the user would quite often appreciate or even find indespensable a more detailed and complete information on the modelling formalisms (languages) and techniques. The on-line help is available within practically all the menus.

User interface is influenced by Windows environment. It is comfortable but several "trifles" decrease quality of support and productivity of work e.g., there is no *undo*; scrollbars are not sizeable and sometimes one cannot see a whole item; several times more than one hot key has several interpretations within a single menu; there are different dialog boxes for the same actions accross the system.

5 Ability to integrate

The tool allows generating database schemes (SQL DDL statements) required by specific database products. It supports Visual Basic and Power Builder integration by extending the facilities of the standard Visual Basic and Power Builder graphical user interface tools.

A very useful feature is the possibility to manage all development objects, not just those from specifically supported partner tools such as Visual Basic and Power Builder. The facilities of data repository allow to store and control different development documents and application objects. One can access these facilities through Systems Engineer general forms.

6 Conclusions and Comparison with other tools

There are perhaps dozens of CASE tools falling into the same category as the one reviewed here. We cannot report on having experience with such a wide spectrum, so we prefer to refrain from making any statements at such a general level. On the other hand, we do have some experience with other tools, and should like to present their comparison by several selected features (see

Table 1 and 2).

We present CASE tools that we have (directly on indirectly) experience with. We have selected only those tools which we have used in the last two years. We have exercised every effort to have our review findings as accurate and as up to date as possible. However, some of the tools may have been improved recently in newly relelased versions.

Excelerator was the first tool that we used in training our software engineering students. For several years, our students were developing their course projects using Excelerator. We found very useful the possibility of data flows and data store decomposition which is not directly supported in Systems Engineer. We did not list Excelerator in our summary tables because our version is quite outdated and we know of rather substantial improvements in its recent version.

The features described in the tables were selected on the base of Guideline for the evaluation and selection of CASE tools [2]. It is not a complete list of features, but it is sufficient for observing differeneces and similarities between the compared tools. Among the similarities, recoverability (capability to re-establish level of performance and recover the data directly affected in case of failure) is becoming a standard property along with support for release and version management.

Acknowledgement

Assistance by Eduard Zelko in comparing some of the tools is gratefully appreciated.

References

1. M. Bieliková, M. Galbavý, I. Kapustík, Ľ. Molnár and P. Návrat, "Using a CASE tool in developing an information system for Slovak Telecom", In *Information Tools and Technologies*, Moscow (to appear), International Academy of Automation, 1996.
2. ISO/IEC 14102. "Information technology - Guideline for the evaluation and selection of CASE tools", International Standard, First edition, 1995.
3. LBMS ČR. "Systems Engineering CZ 1.0", Learmonth & Burchett Management Systems Plc, 1996.
4. Systems Engineer 6. "Concepts Guide", Learmonth & Burchett Management Systems Plc, 1996.
5. I. Vessey I. and A.P. Sravanapudi, "CASE tools as collaborative support technologies", *Communications of the ACM*, **38**(1) (1995) 83–95.

Characteristics	**Systems Engineer** LBMS	**Object Team** Cadre Technologies[1]	**CASE/4/0** microTOOL
Methodology support (set of methods or methodologies)	LBMS Systems Engineering Methodology – functional model – data model – dynamic model – user interface model	OMT (Object Modelling Technique) – object model – dynamic model – functional model – class communication model – case scenarios	– DeMacro/Yourdon structured analysis[4] – Ward/Mellor real-time extensions – Page-Jones structured systems design – data modelling (Codd, Chen)
Parts of development life cycle addressed	analysis, design, implementation (database generation), maintenance (reverse engineering)	OO analysis, design, implementation (database and code generation), maintenance (reverse engineering)	analysis, design, implementation (database and code generation)
Degree of methodology support	flexible	restrictive	restrictive
User defined entity characteristics	no	no	yes (dynamic forms)
Abstraction/ decomposition (ability to use abstraction principles with a CASE tool)	– process, transaction, menu decomposition – windows properties inheritance – data subsets	– process decomposition – multiple levels of details – object hierarchy	– function, data store, data structure, module decomposition – data flow substitution, data elements hierarchy
Security (ability to prevent unauthorized use or misuse of itself)	– three kinds of users – control access rights by creating groups of users	– role based access control (in the context of a particular project or level of model elaboration within a project)	– flexible user forms – authorised use of entities based on their status
Concurrent development (degree of multi-user support)	– concurrent access to the data dictionary – diagram locking	– interactive access by groups of users – locking system	– entity locking – secure of logical integrity
Release and version management (ability to maintain multiple versions of a system)	– versioning on diagram level – base-lines supported	– configurations, phases, systems, diagrams and classes are versioned	– versioning and configuration management of design entities
User friendliness (1 - poor, 5 - excellent)	4	1[2]	5
Ease of learning	two weeks	we could not estimate[3]	two weeks

[1]Cadre Technologies now merged with Bachman Information systems to Cayenne Software.

[2]Problems with modification of analysis, design and sharing information may have been solved by now.

[3]Partially, a consequence of poor user friendliness.

[4]ObjectiF environment enables transition from CASE/4/0 structured methods to object-oriented methods.

Table 1: CASE comparison

ERwin Logic Works	**BPwin** Logic Works	**StP** IDE	**Characteristics**
Data modelling[5] – logical and physical data models (Information Engineering and IDEF1X notations) – dimensional models	Process modelling (IDEF0, IDEF3 and Gane/ Sarson notations)	– DeMacro/Yourdon and Gane/Sarson structured analysis – Hatley/Pirbhai real time requirements spec – Jackson data modelling – E-R modelling (Chen) – Constantine/Yourdon structured design[9]	*Methodology support* (set of methods or methodologies)
analysis, design, implementation (database generation), maintenance (reverse engineering)	analysis, design	analysis, design, implementation (database generation, code generation), maintenance (C code reverse engineering)	*Parts of development life cycle addressed*
flexible	flexible	guided	*Degree of methodology support*
no	no	yes	*User defined entity characteristics*
– subject areas (division of data model into meaningfull parts)	– process decomposition	– decomposition of almost all entities in diagrams	*Abstraction/ decomposition* (ability to use abstraction principles with a CASE tool)
– the workgroup controlled access – flexible permission and security profiles[6]	no[8]	– locking with login name(s) of the people who are authorized to override the locking status	*Security* (ability to prevent unauthorized use or misuse of itself)
– identification of conflicts created by simultaneous updates[6]	no[8]	– concurrent access to the data dictionary – diagram locking	*Concurrent development* (degree of multi-user support)
– multiple versions of models – detailed change analysis[6]	no[8]	– SCCS (Source Code Control System) or – RCS (Revision Control System) – base-lines supported	*Release and version management* (ability to maintain multiple versions of a system)
3	3	4	*User friendliness* (1 - poor, 5 - excellent)
one week[7]	one week	three weeks	*Ease of learning*

[5]Now OOwin/CRC (object modelling) tool is available which is integrated with ERwin.

[6]These features are provided by ModelMart - model management system for the ERwin family modelling tools.

[7]Not including ModelMart

[8]BPwin is announced to be integrated with ModelMart soon.

[9]StP supports far more techniques than other tools which we have had possibility to work with.

Table 2: CASE comparison