A Knowledge-Based Decision Support System for Scraper Selection and Cost Estimation

Khalid Eldrandaly[1] and Neil Eldin[2]
[1]College of Computers, Zagazig University, Egypt
[2]Construction Technology Department, Purdue School of Engineering and Technology (IUPUI), USA

Abstract: *Scrapers are advantageous earthmoving construction equipments as they are independently capable of excavating, hauling, and placing soil. Determining the suitable and the most economical selection of the size, model, and number of scrapers and pushers is a complex process that depends on multiple factors such as the haul-road conditions, equipment performance, operation's travel time, etc. Equipment selection is the first and the most important decision made by a contracting firm. The right choice of equipment determines competitiveness as well as profitability. Therefore, this paper presents a prototype knowledge-based decision support system for scraper selection and cost estimation. The system is developed through employing Visual Rule Studio® (an object oriented COM-compliant expert system development environment for windows) as an ActiveX Designer under Microsoft Visual Basic 6.0 environment since it combines the advantages of both production rules and object-oriented programming technology. Also, using Component Object Model (COM) technology in designing and integrating the different components of the prototype system will assure software interoperability between these components. The architecture, the development and the implementation of the prototype system are discussed in details. A typical example is also presented to demonstrate the application of the prototype system.*

Keywords: *Scrapers selection, expert systems, decision support systems, COM.*

Received June 6, 2005; accepted August 25, 2005

1. Introduction

Scrapers are advantageous earthmoving machines as they are independently capable of excavating, hauling, and placing soil. Although neither as effective in excavating as the more specialized equipment such as hoes and shovels nor as efficient as trucks in hauling and placing soil, the fact that this one machine performs all three tasks makes it the equipment of choice when moving large quantities of soil. Scrapers are especially economical when hauling is done off-roads and for distances between 500 feet to approximately 3000 feet [7].

To estimate the time and cost for an earthmoving operation, one considers the performance characteristics of the scraper used, the soil properties, length and conditions of haul road, and the various cost components involved [6]. In such an operation, the haul road is divided into segments based on the variations in the road grade resistance. The scraper's maximum travel speed is determined for each segment from the equipment performance chart. The cycle time of the scraper under consideration is computed by adding the time required to travel each segment including the expected acceleration/deceleration when the equipment is approaching the cut/fill ends. Determining the suitable and the most economical selection of the size, model, and number of scrapers and pushers is a complex process that depends on multiple factors such as the haul-road conditions, equipment performance, operation's travel time, etc. Equipment selection is the first and the most important decision made by a contracting firm. The right choice of equipment determines competitiveness as well as profitability. The equipment selection has been the focus of a diverse group of past research. Researchers have developed expert/advisory systems targeting specific types of equipment. For example, in 1990, Touran [9] developed an expert system to aid in the selection of compactness machines. His system takes into account the scope, type of soil, degree of compactness required, and soil properties and recommends the best compactor for the condition specified. In 1994, Hanna [4] created a similar system for crane selection. In 1996, Christian and Xie [1] developed an expert system that takes in consideration the type and quantity of excavated material and recommends the type of equipment that would best accomplish the work. In 2005, Eldin and Mayfield [2] developed a Microsoft® Excel application that automates the calculations required to determine the production rate and the unit cost of all scrapers available for the site under consideration. However, not all scrapers available could be suitable for the job conditions. Such a system could be made more useful if the calculations are preceded by an advisory expert system to list only the scrapers viable for the conditions of a certain job and to perform the calculations for such a list. The purpose of this paper is to do so. This paper presents a prototype knowledge-based decision support system for scraper selection and production estimation. The system is developed through employing Visual Rule Studio® (an object oriented COM-compliant expert system development environment for windows) as an ActiveX Designer under Microsoft Visual Basic 6.0 environment, since it combines the advantages of both production rules and object-oriented programming technology. Also, using Component Object Model (COM) technology in designing and integrating the different components of the prototype system will assure software interoperability between these components. The architecture, the development and the implementation of the prototype system are discussed in details. A typical example is also presented to demonstrate the application of the prototype system.

2. Software Interoperability and COM Technology

Interoperability is the ability of two or more software components to directly cooperate/ communicate despite of their differences in programming language, inte face, and execution platform [3]. The development and deployment of successful interoperability strategies require standardization that provides the communication channels and format needed for direct exchange and integration of information [10]. COM is a standard that enhances software interoperability by allowing different software components, possibly written in different programming languages, to communicate directly [5]. COM specifies an object model and programming requirements that enable COM objects to interact with other COM objects. COM allows these objects to be reused at a binary level and thus third-party developers do not require access to source code, header files, or object libraries in order to extend the system [11].

3. Architecture of the Proposed Knowledge-Based DSS

The proposed system was developed as a three-tier architecture using three COM-compliant commercially available software packages: Microsoft®Visual Basic 6.0, Visual Rule Studio®, and Microsoft® Access 2003.

This software industry standard architecture provides a framework for logical components of the software to interact and enables flexibility in managing changes and updates in system components. This architecture consists of three layers as follows:

- *Presentation Layer*: (Client-tier) responsible for the presentation of data, receiving user events, and controlling the user interface.
- *Business Layer*: (Application-server-tier) contains a number of software modules that use the user interface data to perform required tasks.
- *Database Layer*: (Data-server-tier) responsible for data storage.

Figure 1 schematically shows the three-tier architecture of the proposed system.

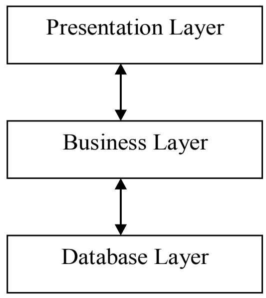

Figure 1. Three-tire architecture of the proposed system.

4. Development of the Proposed Knowledge-Based DSS

Visual Rule Studio® (an object-oriented COM-compliant expert system development environment for windows) was used to develop the prototype knowledge-based DSS. Visual Rule Studio solves the problem of software interoperability by allowing the developers to package rules into component reusable objects called RuleSets. By fully utilizing OLE and COM technologies, RuleSets act as COM automation servers, exposing RuleSet objects in a natural COM fashion to any COM compatible client. Visual Rule Studio installs as an integral part of MS Visual Basic 6.0, professional or enterprise editions, and appears within the visual Basic as an ActiveX Designer. RuleSets can be complied within Visual Basic .EXE, .OCX, or .DLL executables and used in any of the ways the developers normally use such executables [8].

The Visual Rule Studio's object-oriented rules technology is a new adaptation of rule-based expert system technology. It is based on the Production Rule Language (PRL) and Inference Engines of LEVEL5 Object®. Rules in a production system consists of a collection of If Condition-Then action statements. Each rule has a left-hand-side, or If part, and a right-hand side, or Then part. The If part of a rule comprises the conditions or antecedents of the rule. The Then part is the action part of a rule and is often called the rule's consequent or conclusion. A RuleSet may contain class declarations and methods, forward-chaining rules, backward-chaining demons, and an agenda. The grammar of the PRL uses an object-referencing notation that is the same as that of all popular language environments, such as, C++, Java, and Visual Basic. Visual Rule Studio objects are used to encapsulate knowledge structure, procedures, and values. An object's structure is defined by its class and attributes declarations within a RuleSet. Object behavior is tightly bound to attributes in the form of facets, methods, rules, and demons [8]. Figure 2 shows the breakdown of a Visual Rule Studio object's structure.

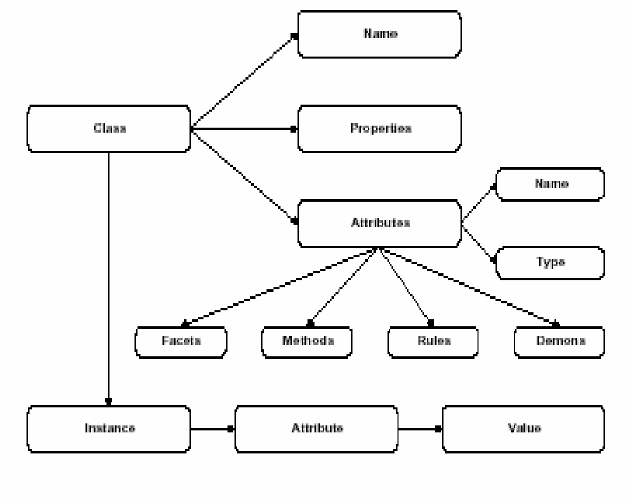

Figure 2. Structure of a visual rule studio object (adapted from RuleMachines [8]).

The RuleSet of the proposed system consists of 8 classes and 82 rules; the following gives a typical example of the classes and rules used in the RuleSet:

Class Decisions
With Decision COMPOUND Use Scrapers, Using Scrapers is not economical
With Haul_Distance STRING
With Haul_Road STRING
With Exavated_Soil STRING
With Expeted_Hauling_Speed STRING

Rule 1 Using Scrapers
If Decisions. Haul_Distance = "Between 300 Feet and 4000 Feet"
And Decisions. Haul_Road = "Mostly off Road"
And Decisions. Exavated_Soil = "Not In tacked Rock"
And Decisions. Expeted_Hauling_Speed = "Not Exceeding 35 MPH"
Then Decisions. Decision IS Use Scrapers
Else Decisions. Decision IS Using Scrapers is not economical

The inference engine of Visual Rule Studio's production system acts as the "*unseen hand*" or executor which causes processing to take place. Processing here is defined as the combining of supplied data with rules to create inferred data. It is the inferred data that is the desired end result of the production system processing. The Visual Rule Studio inference engine provides two primary problem-solving engines relevant to production systems: The forward chaining engine and the backward chaining engine. In the proposed expert system, forward chaining engine is used. Starting from an initial or current set of data, the forward chaining inference engine makes a chain of inferences until a goal is reached. In forward chaining, the data values of the context are matched against the If parts, or left-hand-sides, of rules. If a rule's If side matches the context, then the inference engine executes the Then part, or right-hand-side of the rule. If the execution of the Then part of a rule changes the data values of the context, then the inference engine repeats the entire match-execute cycle again using the new state of the context data values as a new initial set of data (Figure 3).

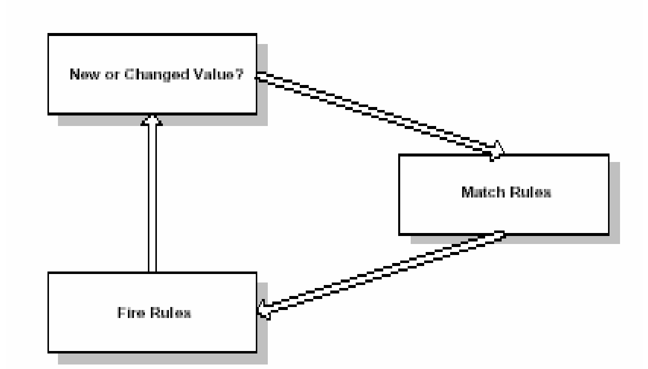

Figure 3. Forward chaining inference processing (adapted from RuleMachines [8]).

Microsoft® Access 2003 was used to develop the database module. This database maintains the necessary data such as standard soils' properties and scrapers' information including their performance charts. Although the data entry required for creating the database may seem time consuming, entries are only done one time when initiating the database.

Microsoft® ActiveX® Data Object (ADO) was used to read required information from the database. ADO provides consistent, high-performance access to data and supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects, using applications, tools, languages or Internet browsers. ADO was implemented using a set of COM-based interfaces that provide applications with uniform access to data stored in diverse information sources [5].

5. Example of Consultation Session

The knowledge-based DSS presented in this paper is designed specifically for earthmoving operations using scrapers. In order to demonstrate how the system can assist project engineers and estimators in selecting the most suitable scraper model and in calculating the expected costs for the job on hand, a sample run is demonstrated in this section.

Upon execution of the system, the system gives the decision maker the option of either checking the suitability of using scraper or entering the data of the project to calculate the expected costs as shown in Figure 4. It also, gives the decision maker the option of either using the program for choosing the suitable scraper or choosing the scraper by himself as shown in Figure 5. After choosing the suitable scraper as shown

Figure 6 and entering the project data (e. g., excavation quantities, soil type, haul road conditions, travel distance, etc.) as shown in Figure 7, the system determines the production rate, the time needed to complete the operation, and the expected costs for the job on hand as shown in Figures 8 and 9. Figures 4, 5, 6, 7, 8, and 9 are just only samples of the screenshots during a typical run.

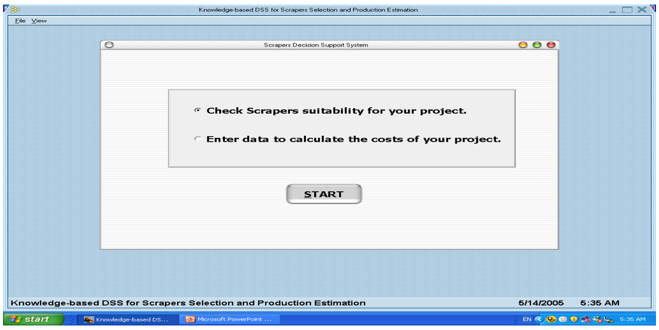

Figure 4. Checking the suitability of using scraper or entering the project data

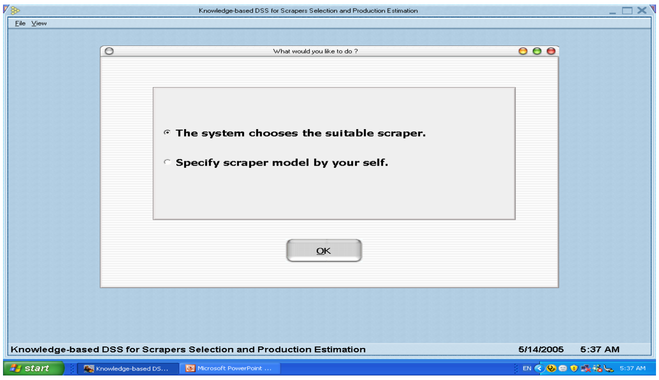

Figure 5. Choosing the suitable scraper.

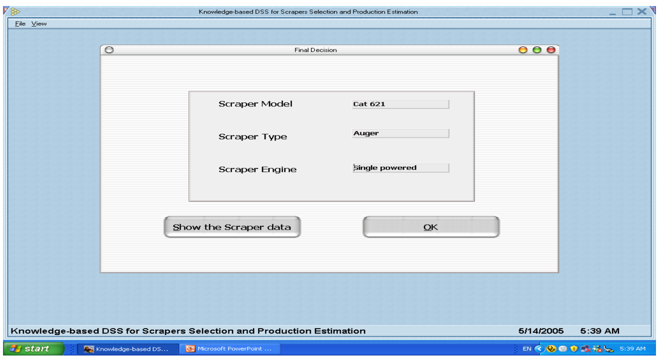

Figure 6. The selected scraper.

6. Conclusions

In this paper, the architecture, the development, and the implementation of a prototype knowledge-based DSS for scraper selection and cost estimation are discussed in details. The use of Visual Rule Studio® (an object-oriented COM-compliant expert system development environment for windows) which runs together with Microsoft Visual Basic 6.0 is found to be very effective in producing the system under windows environment. The advantages of both production rules and object-oriented programming paradigm are accomplished. Also, software interoperability between the different components of the system is achieved by adopting the COM technology in designing the system. This system will help project engineers and estimators in selecting the most suitable scraper model and in calculating the expected costs for the job on hand and allows them to examine alternatives by conducting what-if-scenarios whenever needed. The system could be extended also to allow the user to enter limitations on the time allowed for a certain job and request the system to provide the total number of scrapers and pushers required to complete the work within the time allotted.

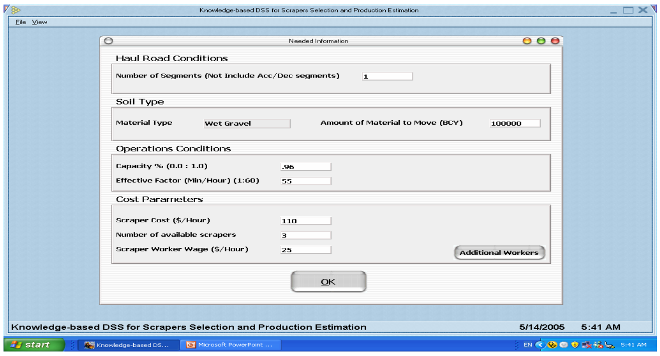

Figure 7. Entering the project data

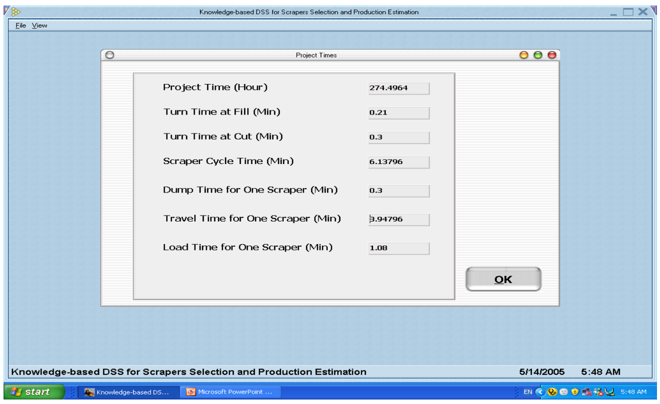

Figure 8. The expected times of the project.

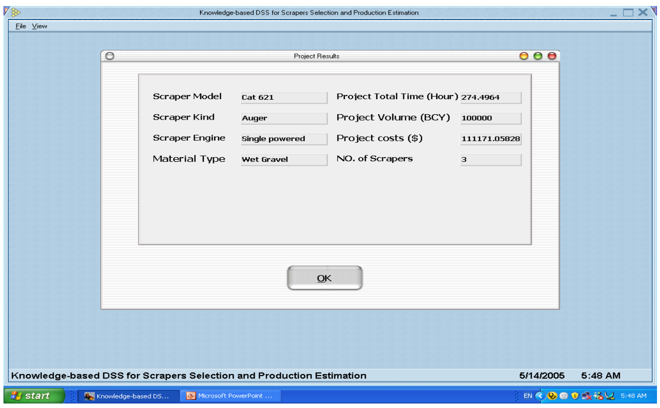

Figure 9. The expected costs of the project.

References

[1] Christian J. and Xie T. X., "Improving Earthmoving Estimating by More Realistic Knowledge," *Canadian Journal of Civil Engineering*, vol. 23, pp. 250-259, 1996.

[2] Eldin N., and Mayfield J. "Determination of Most Economical Scrapers Fleet," *Journal of Construction Engineering and Management*, ASCE, vol. 131, no. 10, 2005.

[3] Finkelstein A., "Interoperable Systems: An Introduction," in Kramer B., Papazoglou M., and Schmidt H. (Eds), *Information Systems Interoperability*, Research Studies Press, England, pp. 1-9, 1998.

[4] Hanna A., "SELECTCRANE: An Expert System for Optimum Crane Selection," *in Proceedings on the 1st Conference of Computing in Civil Engineering*, pp. 958-963, 1994.

[5] Microsoft, *Microsoft Developer Network Online Documentation*, MSDN Library, 2003.

[6] O'Brien J. J. and Zilly R. G., *Constructor's Management Handbook*, McGraw-Hill, New York, 1971.

[7] Peurifoy R. L. and Schexnayder C. J., *Construction Planning, Equipment, and Methods*, McGraw Hill, New York, 2002.

[8] RuleMachines, *Visual Rule Studio Developer's Guide*, Canada, OnDemand, 2002.

[9] Touran A., "Expert System for Compactor Selection," *Journal of Transportation Engineering*, vol. 116, no. 3, 1990.

[10] Vckovski A., *Interoperable and Distributed Pro essing in GIS*, Taylor and Francis, London, 1998.

[11] Zeiler M., *Exploring ArcObjects-ESRI*, Relands, CA, USA, 2001.

Khalid Eldrandaly is an assistant professor of computer information systems and interim head of Information Science and Technology Department, College of Computers and Informatics, Zagazig University, Egypt. He received his BS degree in civil engineering, his MS degree in systems engineering AI and expert systems, and his PhD degree in systems engineering (GIS). He was a visiting scholar at Texas A&M University, USA, for two years. His research interests include GIS, expert systems, SDSS, MCDM, and intelligent techniques in decision making. He is a member of the World Academy of Young Scientists (WAY), Arab Union of Scientists and Researchers (AUSR), and Texas A&M International Faculty Network.

Neil Eldin is a professor and head of Construction Technology Department at Purdue School of Engineering and Technology (IUPUI). He has over 30 years of professional experience involving teaching, research, professional service and industry positions in construction engineering and management. During his 15 years in academia, he taught the more challenging construction engineering and management courses. He received The-Teacher-of-The-Year Award in 1997, 1998, 1999, 2002, and 2004. He has an impressive research record of $1.5 million funded research and over 40 refereed publications. His recent research resulted in filing a US Patent No. 60/528,814. His research interest includes cost-schedule integration, construction projects management, environmental assessment of construction materials, and solid waste as construction materials, and robotic applications in construction. He is a licensed Professional Engineer (PE), CPC and OSHA Trainer.