

Open Architecture for Site Layout Modeling

Farnaz Sadeghpour, Osama Moselhi, Sabah Alkass

*Department of Building, Civil and Environmental Engineering, Concordia University
1455 de Maisonneuve West, Montreal, Quebec, Canada*

ABSTRACT: This paper presents an overview of a computer-based site layout model and focuses primarily on the project setup phase. The developed model has four modules: user interface, database, project module, and layout module. Setting up the project in the proposed model is carried out by the project module, utilizing open architecture concept. The main advantage in the open architecture is to allow for the incorporation of user-defined objects if they are not readily available in the model. The objects required to define a site layout problem are clustered into three tiers: 1) Site Objects, 2) Construction Objects, and 3) Constraint Objects. The model is implemented in a CAD environment using an object-based approach. The structure of each of the three tiers is described, and the mechanism of object selection/creation for a site layout project is explained. The paper describes the components required to implement an open architecture for site layout object selection along with their respective environments. The developed model can be easily extended to similar applications such as floor plan design.

KEYWORDS: CAD tools, Construction sites, Model design, Open architecture, Site layout planning.

1. INTRODUCTION

One of the problems in modeling construction site layout is the diversity of acceptable solutions and lack of exact rules or methods to follow (Cheng and O'Connor 1993). Previous surveys show site managers design the layout based on their experience, common sense and adaptation of past layouts (Rad and James 1983). Furthermore, a range of facilities can handle the same task, leaving even more options for site managers to choose from. Several factors are considered by designers in choosing a facility such as construction type, contract type, and project size and location (Hamiani 1987). Therefore, facility lists are justified by the uniqueness of the project.

Previous research works on site layout have mainly concentrated on the layout process and not the structure of the project setup and selection of temporary facilities. As a result the computer models have been programmed for a limited and/or fixed number of facilities, site conditions, or layout constraints (Elbetagi and Hegazy 2003, Mawdesley et al. 2002, Zhang et al. 2000, Harmanani et al 2000, Li and Love 1998, Philip 1997, Yeh 1995, Hamiani 1987). Consequently they can not readily be applicable to any site layout problem, but to the one they have been programmed for. On the other hand, it is

impossible to identify and embed every item that is used on construction sites. Construction industry has an open nature and new methods and products are continuously introduced. To comply with this diversity, an open architecture for site layout models is proposed in this paper. It focuses primarily on the project setup phase of a previously proposed computer-based site layout model (Sadeghpour et al. 2002). The model has four modules: user interface, database, project module, and layout module. The focus of this paper is to explain the functionality and mechanism of the project control module. This module assists in defining the requirements of a site layout and setting up a new project.

2. OPEN ARCHITECTURE FOR PROJECT SETUP

The developed module utilizes open architecture in order to allow the creation of objects that do not exist in the libraries of the model. Open architecture utilized in this paper calls for the formation of general categories that host a number of entities relevant to the problem being modeled. Selection of categories is based on the intuition of the model designer respecting two principles: 1) within each category, the entities share the same attributes; and 2) the selected categories are adequate to describe and present the project being

modeled. The project module of the proposed model applies these principles of open architecture to site layout problem.

Three tiers of objects identified for the proposed site layout models are: 1) site objects, 2) construction objects, and 3) constraint objects. Object-based concepts have been utilized to represent these three tiers. Object-based approach strongly promotes the formalism of data typing and encapsulation of information. Contrary to the object-oriented approach, no explicit inheritance scheme is imposed in the object-based approach (Zamanian 1992). As a result, the proposed tiers are implemented as object classes, encapsulating their relevant attributes. These attributes include the geometric and non-geometrical data and knowledge. The three aforementioned tiers of objects are described in more details below.

2.1 Site Objects

Site objects include the definition of site boundary and objects that reside on site before the commencement of construction, and hence have a known location. Site objects affect the location of construction objects, and consequently the final layout. Some examples of site objects are trees, existing buildings, specially marked areas on the site such as “unavailable”, “unsafe”, or “hazardous”, water ponds, or life lines such as sources of electricity, water, or phone lines. Site objects often exist on site permanently, however their duration on site can be specified if needed. In spite of their impact on site layout, site objects have often been overlooked in previous layout researches except in few (e.g. Lundberg et al. 1989). Site objects have two main roles in site layout: 1) they occupy space on site, so the area they occupy is deducted from the total site land; 2) their topological relations with construction objects define the constraining rules and hence affect the final layout of construction sites. Site objects contain geometrical attributes such as location on site, as well as non-geometrical attributes such as duration on site. A UML representation of site object is shown on Figure 1.

2.2 Construction Objects

Unlike site objects, construction objects are to be located on the site and hence do not have a predefined location. Construction objects address a range of items that are diverse in nature including equipment, material, temporary support

facilities, buildings, lay-down areas, working areas, and generally objects that have to be located on site. Similar to site objects, construction objects are represented by object class (see Figure 1). The structure of a construction object is similar to that of a site object with a set of geometric and non-geometric attributes. Specific characteristics of construction objects are modeled as additional attributes. As an example construction objects can be fixed or movable. Unlike movable objects, the location of fixed objects does not change during the course of construction once they are located. The main difference between construction objects and site objects is that the location of construction objects is to be determined. For this matter, construction objects have constraint attribute that points to a number of constraint objects.

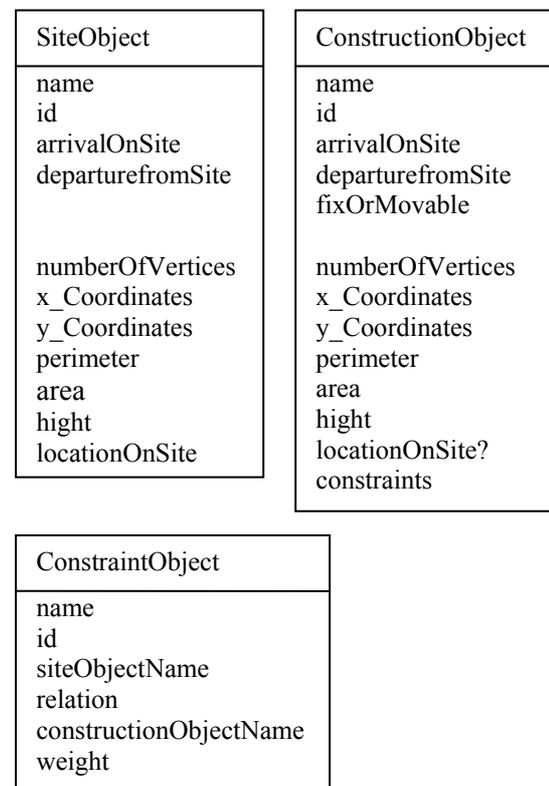


Figure 1. UML notation for three tiers of objects.

2.3 Constraint Objects

The process of finding optimum location for a construction object is carried out under a set of rules. These rules are mapped into a set of related objects referred to as constraint objects. Constraint objects are designed based on the objectives of the layout. As a result, the model is not limited to minimizing traveling distance, but can define other objectives such as safety and security.

Constraint objects are as well represented by object class. The main attributes of constraint objects are “siteObjectName”, “relation”, and “constructionObjectName” (Figure 1). These properties have been designed in order to facilitate the creation of new constraints when needed. Each constraint is structured of three main parts: 1) constrained element; 2) a topographic relationship; and 3) constraining element (Figure 2). The *constrained element* is always a construction object to which the constraint is assigned. *Constraining element* can be either another construction object, or a site object, which constrains the location of the constrained element. As an example, a safety constraint indicates that the explosives should not be kept close to offices. Offices are constraining the location of explosives with the topographic relation of “not close to”. If the constraining element is a construction object, like a trailer office, the constraint object is called (C-C). On the other hand if the constraining element is a site object, like an existing office building on site, the constraint object is classified as (C-S) (see Figure 2).

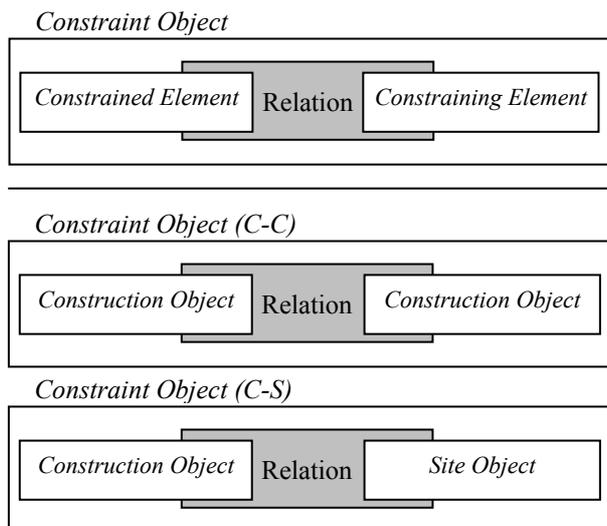


Figure 2. The structure of constraint object

3. DATAFLOW IN PROJECT MODULE

To achieve the flexibility of open architecture in the proposed model, the project module offers the users: 1) a list of the ready-to-use objects from each category to select objects that are pertinent to the project at hand; and 2) an environment that supports the definition of new objects and modification of existing ones if necessary. This process takes place via a graphical user interface. The model maintains libraries for the three tiers of objects described previously: “site library”,

“construction library”, and “constraint library”. Figure 3 demonstrates the functionality of the project module in relation to the libraries of the model when setting up a project.

Setting up a new project involves *defining* objects from the three tiers. The term *define* refers to selection from a library, modification of existing objects, or creation of new ones. Once an object is selected from a library, the object along with the knowledge and data associated with it are retrieved into the new project work place referred to as *project palette* (Figure 3). The user has the access to modify the object. However, if the required object is not found in the libraries, the open architecture of model allows the user to *create* a new one.

Object-based structure inherently facilitates regeneration of the objects. To create a new object, an instance of its class object is generated. The graphical user interface (GUI) assists in the creation of the geometry of the object in a CAD environment. The built-in databases of CAD model store the geometrical data of the construction and site objects. Examples of geometrical data are the area and the coordinates of object’s footprints. Upon the creation of the geometry, the user interface prompts for the complementary non-geometric data, such as date of arrival to site. To facilitate user data entry a fill-in-the-blanks questionnaire is provided (Figure 4). As part of construction objects’ non-geometrical data, at this point constraint objects are as well selected from the library and assigned to the construction objects. Similar to construction and site objects, the model allows the user to define new constraint objects. The design of the constraint objects described earlier, facilitates the process of their regeneration. The first element is known by the model since it is the construction object for which the constraint is being defined. The second element is selected from a list of topological relations provided by the model. The selection choice for the third element consists of a list of all the construction and site objects involved in the project. Figure 5 shows the user interface for constraint creation process.

The defined non-geometrical data is then merged with the geometric data of the object from the built-in database of CAD system to form a record in the relevant library. This record is associated with the geometry of the object via an attribute handling mechanism that provides a link between

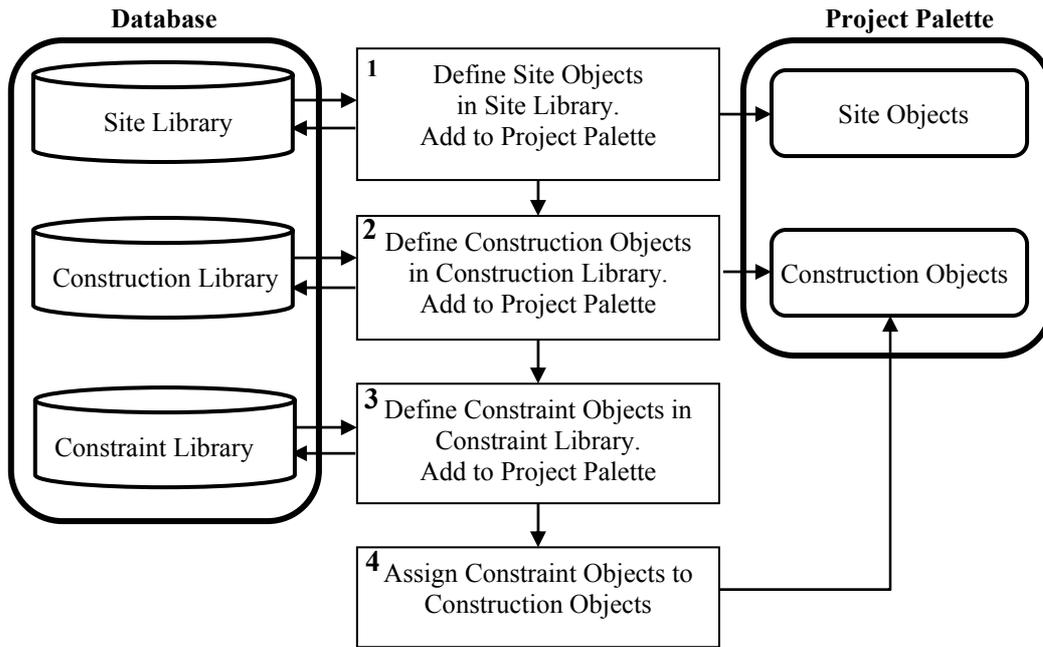


Figure 3. Workflow diagram for initiating a site layout project.

an object and user-defined attributes. This is a two-way link which facilitates retrieval of the attributes of a physical object, or conversely, finding the physical object by its record in the database. An instance of *defined* construction and site objects is sent to the *project palette*, representing the requirements for the project at hand.

Once a new object is created, it is automatically added to the corresponding library. This eliminates duplication and redefinition of objects for different projects. It also supports the expansion and enrichment of the model's libraries, and more importantly, gradually customizes the model according to design needs and preferences of its users.

The proposed model is designed to interact with users at both expert and novice levels. The first level provides the domain-knowledge expert with tools to enrich the model's libraries. This allows planners to apply their individual problem solving strategies, and thus, directly contribute to the knowledge base of the model. This feature eliminates the traditional need of a knowledge engineer for acquiring and structuring the extracted knowledge, and hence decreases the risk of misinterpretation and incomplete acquisition of relevant knowledge. Based on the model's status of knowledge, the project module provides less-experienced site planners with a decision support

for defining the requirements of a site layout project.

Figure 4. Data enquiry form for construction objects.

4. IMPLEMENTATION

A computer implementation has been developed, based on the open architecture concept described in this paper. Visual Basic for Applications (VBA) in AutoCAD® environment was used for programming. Microsoft Access® served as the external database to accommodate object libraries. VBA provides a seamless link between the model's user interface, AutoCAD, and the database. AutoCAD was chosen due to being one of the most commonly used CAD system, and its graphical capabilities, which facilitate data entry for objects.

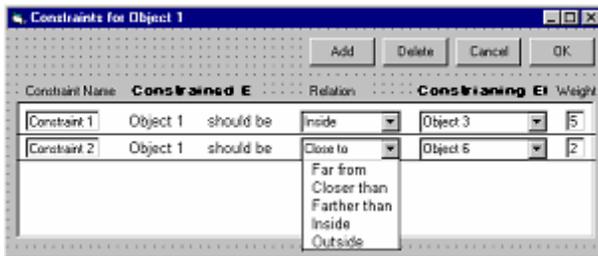


Figure 5. User interface for constraint object creation.

5. SUMMARY

This paper presented an open architecture computer site layout model and focused primarily on project setup module. The objects required to present a site layout project were introduced and their structure was described. The representation of objects in a CAD environment was briefly described along with the process of selection and creation of objects. The open architecture of the model allows for direct contribution of experts to the model's libraries, which can later provide a decision support for less-experienced planners. Using an open architecture in the project setup phase makes the site layout model flexible and not limited, as in previous models, to a single project configuration.

6. REFERENCES

- [Cheng and O'connor 1993] Cheng, M.Y., O'Connor, J.T. (1993). "Site Layout of Construction Temporary Facilities Using Enhanced-Geographic Information System (GIS)", *Proc. of 10th Inter. Symp. on Automation and Robotics in Construction*, Houston, TX, 399-406.
- [Hamiani 1987] Hamiani, A. (1987) "CONSITTE: A Knowledge-Based Expert System Framework for Construction Site Layout", *Ph.D. Thesis Dissertation*, University of Texas, Austin.
- [Harmanani et al 2000] Harmanani, H., Zouein, P., Hajar, A. (2000) "An Evolutionary Algorithm for Solving the Geometrically Constrained Site Layout Problem" *Proc. of the 8th ASCE International Conference on Computing in Civil Engrg.*, Stanford, CA, 1442-1449.
- [Elbetagi and Hegazy 2003] Elbetagi, E., Hegazy, T. (2003), "Optimum Layout Planning for Irregular Construction Sites", *Proc. of 5th Construction Specialty Conference of Canadian Society of Civil Engrg.*, CSCE, Moncton, NB, COG-197.
- [Li and Love 1998] Li, H., and Love, P.E.D. (1998), "Site-level Facilities Layout Using Genetic Algorithms", *J. of Computing in Civil Engrg.*, ASCE, 12(4), 227-231.
- [Lundberg et al. 1989] Lundberg, E. J., Beliveau, Y. J. (1989), "Automated Lay-Down Yard Control System -ALYC", *J. of Construction Engrg. and Management.*, ASCE, 115(4), 535-544.
- [Mawdesley et al. 2002] Mawdesley, M.J., Al-jibouri, S. H., and Yang, H. (2002). "Genetic Algorithm for Construction Site Layout in Project Planning" *J. of Construction Engrg. and Management*, ASCE, 128(5), 418-426.
- [Philip 1997] Philip, M., Mahadevan, N., and Varghese, K. (1997) "Optimization of Construction Site Layout – A Genetic Algorithm Approach" *Proc. of 4th ASCE Congress on Computing in Civil Engrg.*, Philadelphia, PA, 710-718.
- [Rad and James 1983] Rad, P.F. and James, B.M. (1983) "The Layout of Temporary Construction Facilities" *Cost Engineering*, AACE, 25(2), 19-27.

[Sadeghpour et al. 2002] Sadeghpour, F., Moselhi, O., and Alkass, S. (2002). "Dynamic Planning for Site Layout", *Proc. of 30th Annual Conference of Canadian Society of Civil Engr.*, CSCE, Montreal, QC, g73.

[Yeh 1995] Yeh, I-Cheng (1995) "Construction-Site Layout Using Annealed Neural Network" *J. of Computing in Civil Engrg.*, 9(3), 201-208.

[Zamanian 1992] Zamanian, M.K. (1992), "Modeling and Communicating Spatial and Functional Information about Constructed Facilities" Ph.D. Dissertation, Engineering Design

Research Center, Carnegie Mellon University, Pittsburgh, PA.

[Zhang et al. 2000] Zhang, J.P., Anson, M., Wang, Qian (2000) "A new 4D Management Approach to Construction Planning and Site Space Utilization", *Proceedings of the 8th International Conf. of Computing in Civil and Building Engr.*, ASCE, Stanford, CA, 15-22.