A GENERIC INCLUSION OF SPACE STRATEGIES WITH ACTIVITY EXECUTION PATTERNS IN 4D TOOLS

Zaki Mallasi and Nash Dawood
Centre for Construction Innovation Research, School of Science and Technology,
University of Teesside
z.mallasi@tees.ac.uk, n.n.dawood@tees.ac.uk

SUMMARY

In this paper we describe construction generic space strategies that affect the development of realistic 4D space visualisations. The simple and dynamic approach has been implemented in the PECASO model to allow a new insight into a project's space-time schedules. Our approach considers the activities execution patterns among the variables used for minimising space-time conflicts between site operations. The semantics of a construction activity execution patterns are illustrated in this work and they are: 1) progress of work direction, 2) execution of work direction, and 3) activity volume of work per week. The PECASO system applies a Simple Genetic Algorithm (SGA) to search for the most suitable execution pattern suitable for a construction activity. Among the included space strategies are the physical constraint such as activity-products Assembly Sequence Constraints (ASC) and the construction logic dependencies. The SGA has been proposed here to model the generic space strategies for the execution patterns. This research suggests that the definition of activity execution patterns semantics in 4D is an important element of interaction between site operations and could shape the site space usage in a different way. Other advantages are the benefits that can be generated from rehearsing different 'what-if' scenarios for coordinating site operations and to communicate the project plan in 4D. The paper presents an experimental execution patterns SGA runs with results, and shows how they are used to minimize space-time conflicts.

1. INTRODUCTION

Construction planners often communicate the coordination of the planned schedules based on highly generalised conceptual space terms such as North, South, East and West. Take an example of a construction planner when conveying the execution of *Ground Floor Steel Columns* activity to begin from the East and progressing towards the West by 100m^3 per week work rate. The execution plan of such activity is left to the workmen on the job, and it neither includes a detailed space strategy nor coordination with other activities execution patterns. With such conveyed statement, spatial interferences and work interruptions between site operations might occur on the site (Riley and Sanvido, 1997; Mallasi and Dawood, 2001; Guo, 2002). These spatial conceptual terms practised by industry are insufficient for coordinating the workmen on the job, especially in large complex construction projects where the site space involves a number of constrained site operations. Cheng and O'Connor (1996) claims that, in field practice, construction planners have to interpret spatial information into poor paper-based drawings and diagrams. The recent development that we present here contributes significantly in the construction industry toward increasing construction planners' awareness especially when coordinating and planning site operations inside the building boundary.

1.1 Background

Current progress in computerisation and Computer Aided Design (CAD) systems seems to offer a great opportunity for improving building construction information. For example, AutoCAD 2000 utilises drawing integration automated technique through the Open Database Connectivity (ODBC) and stores the CAD graphical information in the project database (Mallasi and Dawood, 2002). This successful automation in CAD is widely applied in many research, and lead into the development of next generations CAD systems. Kunigahalli et. al. (1995) generated the concrete placement process by extracting the topological relationships of floor slabs from CAD model of a given floor slab. Complete building geographical information can be retrieved from the CAD model like the components coordinates' values, the components 3D dimensions, geometrical adjacency relationships, volumes, and location data. Other models used GIS for dynamic site layout planning (Tommelein and Zouin, 1993; Elbaltagi, et. al., 2001). Similarly, Deb and Gulati (2001) have utilised

GIS software on top of CAD to acquire quantities of work takeoff and integrated cost estimates with material layout planning. These CAD systems are supportive in acquiring design building information (see Fig. 1-a) to databases, spreadsheets, and design data to generate technical reports (cost estimates, bill of quantities and materials, doors & windows schedules, and so on), but do not provide any evaluation criteria for visualising space and the on-site construction processes in time (McKinney et. al., 1998).

Further revolution in CAD systems also introduced the 4D CAD (3D + time) tools to assist project planners in visualising construction processes (see Fig. 1-b) and detect possible constructability problems before commencing work onsite. Akinci et. al. (2000) formalised an approach for space-time conflict analysis in 4D and defined construction workspace types, and taxonomy for classifying spatial conflicts during construction. Thabet and Belivau (1997) modelled the progress of construction processes by defining a hierarchy system of component blocks that in turn represents construction phases. Their model requires the planner to manually specify the components block and perhaps produce a detailed schedule of work. However, Akbas et. al. (2001) identified the need to improve the phasing approach to provide more effective 4D visualisations, i.e. ‘construction zone generation’. He proposed a product model where spaces are combined together to represent the production rate for an activity. However, detailed geometry is necessary for visualising smaller areas within the zones. Hierarchical product space models were utilised to represent the level of detail in the project schedule (Xu and AbouRizk, 1999; Mallasi and Dawood, 2002).

1.2 Paper Organisation

In this paper we make contributions to 4D space visualisations by studying the result of dynamically applying several execution patterns (refer to Fig. 1). In the second section, we provide a geographical analysis of what is called an activity *execution pattern*, building on Riley and Sanvido (1997) research in planning a multi-story building. Cartesian coordinates are used as a universal location definition when choosing the progress of work direction, execution of work direction, and access points. We show how, by explaining our efficient weekly work rate simulation algorithm, we can visualise the dynamic nature of site operations. Section three looks at the SGA method adopted here (Babu and Babu, 2001) for finding the best execution pattern for project activities. Here, the activity execution logic constraints are applied to match execution sequence in the new scenario. We implement the constraint detection algorithm as described by Anderl and Mendgen (1995). Before we conclude our paper, we present in section four an experimental SGA execution patterns 4D simulation showing that the changing of an activity execution pattern could reduce spatial interferences between site operations. The spatial constraints are applied to adjust the assembling relationships between supported products and non-supported ones. The change of an execution pattern results in reassembling the construction products in a different order and hence occupies different area on site.

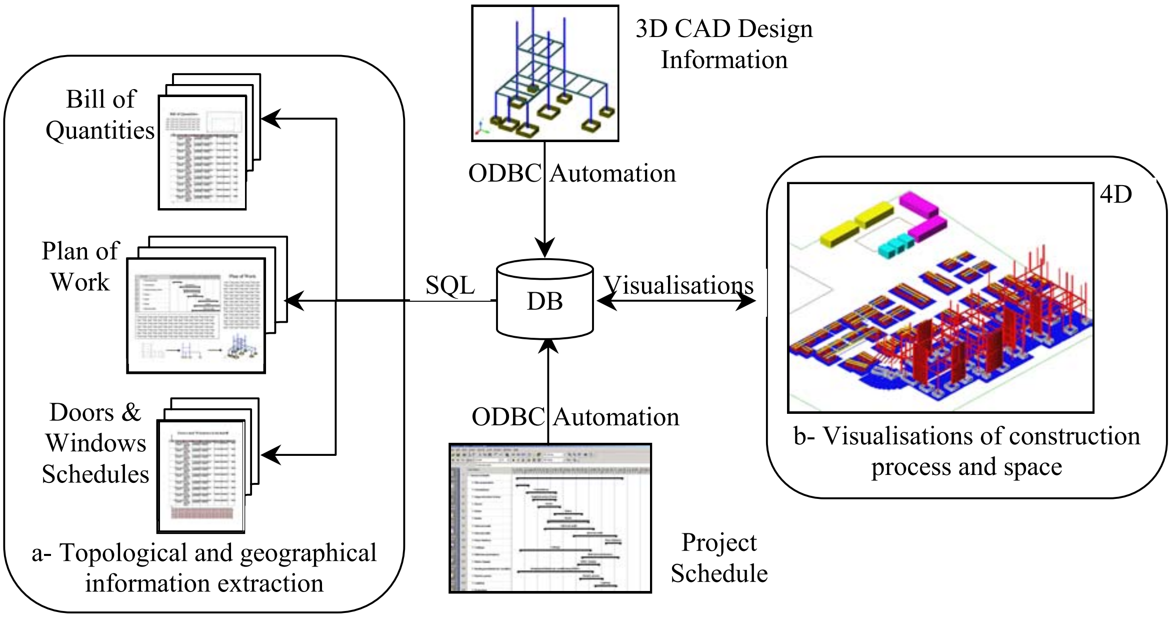
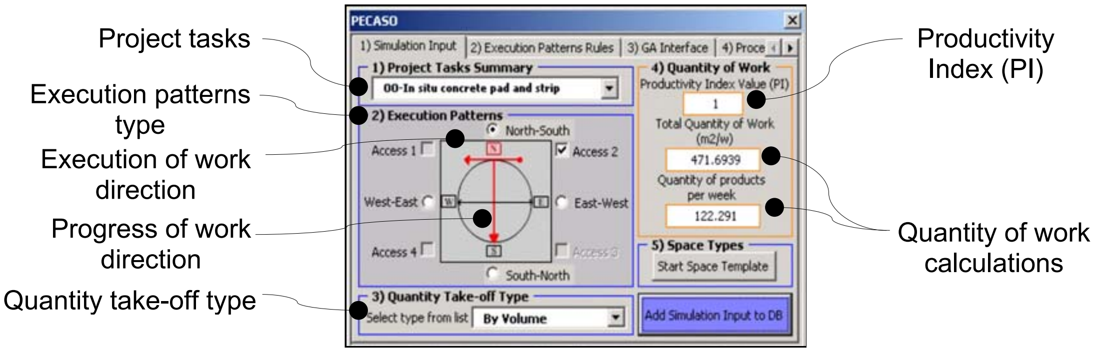


Figure 1: Current automation in CAD systems for integrating design and construction information

2. EXECUTION PATTERNS

2.1 Modelling Semantics of Execution Patterns

A prototype PECASO system was developed as an add-on AutoCAD 2000 in Visual Basic for Applications (VBA), to represent a universal methodology for modelling the activity execution patterns. As seen in Figure 2, the first semantic of an activity execution patterns is the *Progress of Work* (PW) direction and it is presented in the form of four cardinal directions such as North, South, East and West. The second semantic is the activity *Execution of Work Direction* (EWD) that is perpendicular to the *Progress of Work* direction. The effect EW on the PW produces the rest of the eight sub-cardinal directions. For example, the execution pattern North-South-Access2 forces the PW to commence from the North to the South, with priority access point for EW from the East. The spatial reasoning algorithm developed in PECASO generates a total of twelve execution patterns. It interprets geographically the location where activities are executed. The 3D geometrical components geodetic coordinates are classified approximately into longitude and latitude location (X and Y coordinates). Such classification algorithm is achieved by retrieving 'spatial indexing' (Goyal, 2000) values for X and Y points from the database paying attention to priorities for PW, EWD, and access point.



1) Project Tasks Summary: 00-In situ concrete pad and strip
2) Execution Patterns: ● North-South, ☑ Access 2
3) Quantity Take-off Type — Select type from list: By Volume
4) Quantity of Work
Productivity Index Value (PI): 1
Total Quantity of Work (m2/w): 471.6939
Quantity of products per week: 122.291

Figure 2: Illustration of the Execution Patterns semantics in PECASO user interface

The third semantic considers the dynamic calculation of the Quantity of Work (QW) in a weekly basis. This semantic is included to overcome the tedious effort from manually breaking the CAD model into building blocks or phases. More specifically, this approach graphically quantifies the amount of work required per week for an activity and visualises the appropriate number of building components in 4D. The unique feature in the system uses three types of quantity take-off they are: by area, by volume, and by unit. Figure 3 shows a possible scenario of the effect from combining the three semantics showing the dynamic space site usage generation. This semantic calculates the total QW for activities from the database and uses the QW per week formula as illustrated below in Equation (1) (Mawdsley, 1997). The QW semantic is useful for identifying the amount of finished work, progressing work, and the unfinished quantity of work, and hence visualises the occupied space graphically (refer to Eq. (2), (3), and (4)).

$$QW(pw) = QW(tot) / AD(tot) \qquad \ldots(1)$$

Where

$QW_{(pw)}$: is the quantity of work calculated per week
$QW_{(tot)}$: is the total quantity of work value obtained from the database
$AD_{(tot)}$: is the total activity calendar duration obtained from the schedule information

$$QW(fin) = QW(pw)(MonWeek - Week) \qquad \ldots(2)$$

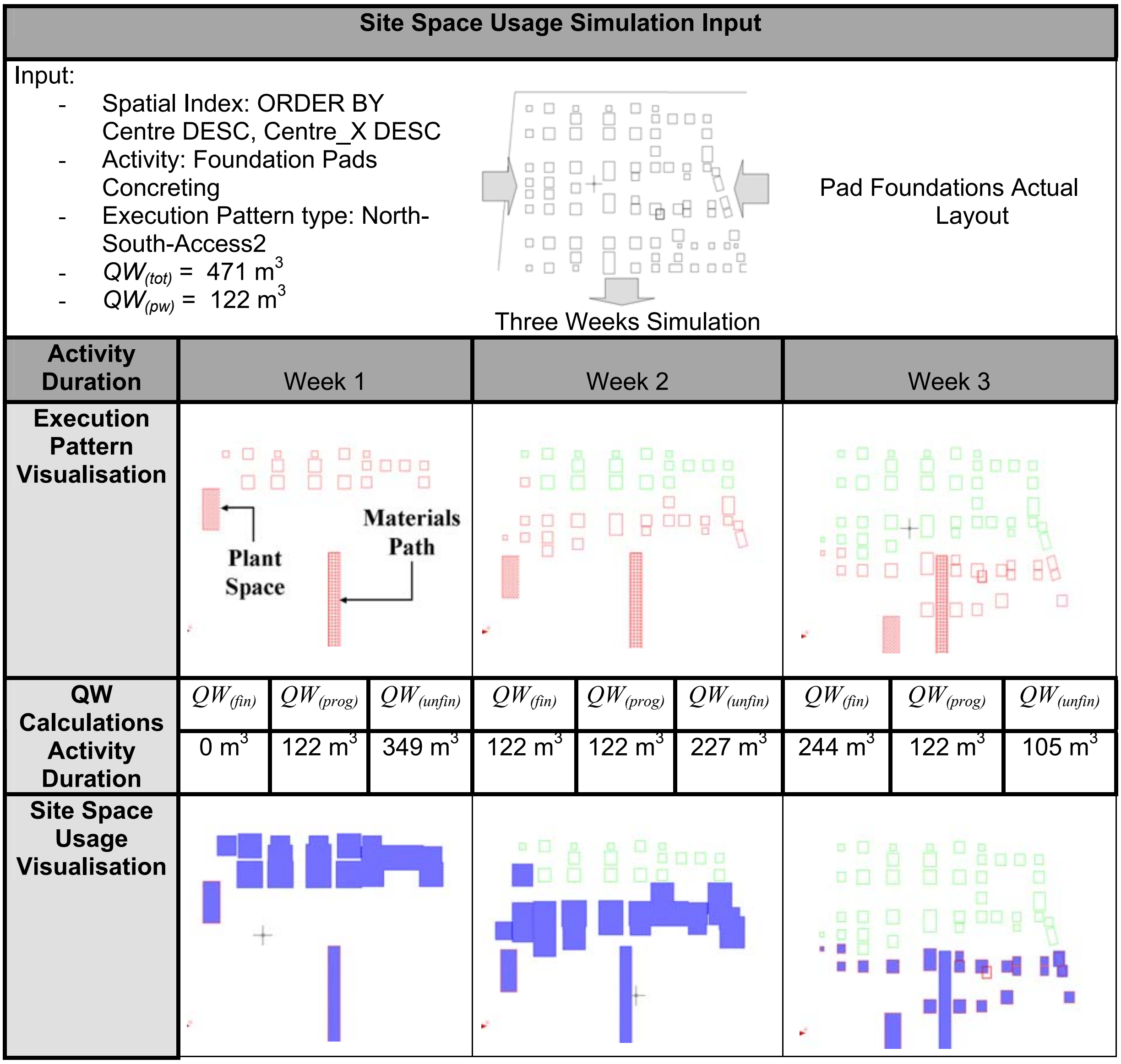
$$QW(prog) = QW(pw) \qquad \ldots(3)$$

$$QW(unfin) = QW(tot) - (QW(fin) + QW(prog)) \qquad \ldots(4)$$

Where

$QW_{(fin)}$: is the quantity of finished work calculated at monitoring week (MonWeek)
$QW_{(prog)}$: is the quantity of progressing work
$QW_{(unfin)}$: the quantity of unfinished work calculated at monitoring week (MonWeek)

An illustration example is shown in Figure 3 for the £8 million School of Health project, the University of Teesside representing execution patterns simulation for the Foundation Pads Concreting activity.



Site Space Usage Simulation Input									
Input: - Spatial Index: ORDER BY Centre DESC, Centre_X DESC - Activity: Foundation Pads Concreting - Execution Pattern type: North-South-Access2 - $QW_{(tot)} = 471\ m^3$ - $QW_{(pw)} = 122\ m^3$	Three Weeks Simulation						Pad Foundations Actual Layout		
Activity Duration	Week 1			Week 2			Week 3		
Execution Pattern Visualisation	Plant Space Materials Path								
QW Calculations Activity Duration	$QW_{(fin)}$	$QW_{(prog)}$	$QW_{(unfin)}$	$QW_{(fin)}$	$QW_{(prog)}$	$QW_{(unfin)}$	$QW_{(fin)}$	$QW_{(prog)}$	$QW_{(unfin)}$
	$0\ m^3$	$122\ m^3$	$349\ m^3$	$122\ m^3$	$122\ m^3$	$227\ m^3$	$244\ m^3$	$122\ m^3$	$105\ m^3$
Site Space Usage Visualisation									

Figure 3: Visualisations of Quantity of Works combined with an activity execution pattern

2.2 Activity-Products Assembly Sequence Constraints (ASC)

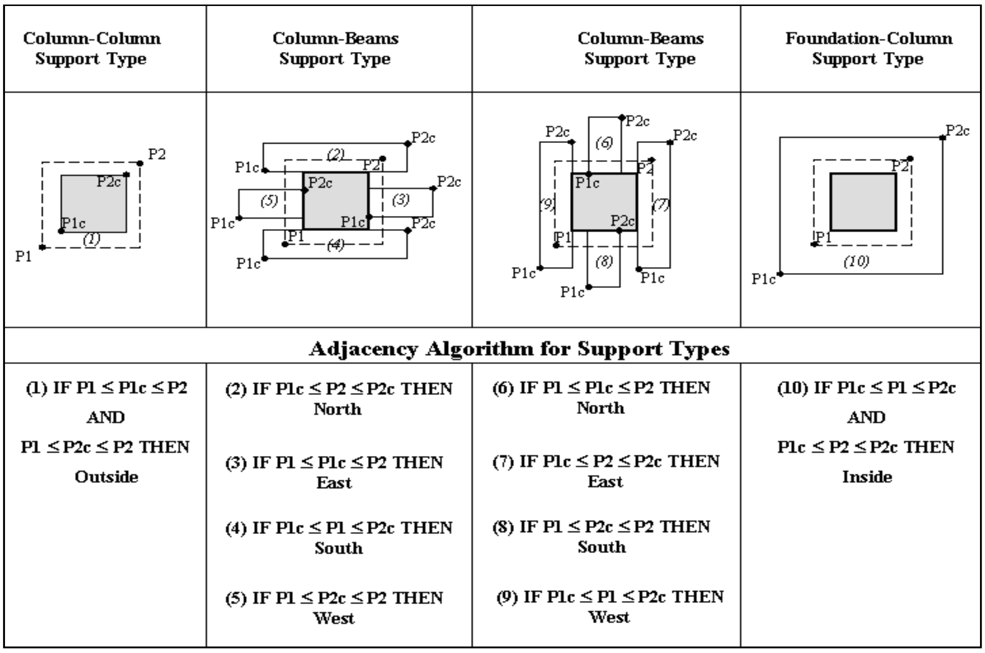
The assembling of the building products follows the generic supportability algorithm built-in PECASO system. For example, the algorithm searches for the proper 3D CAD components assembly sequence. Although previous research used geometrical constraints in AEC modelling and designs, our research here is utilising the constraints detection to verify adjacency relationships between building components (e.g. a column-beam or foundation-column). An illustration of the supportability types is shown in Figure 4. Anderl and Mendgen (1995) believe that the supportability detection can be problematic due to the following reasons:

1) The number of elements required for checking the adjacency relationships is vast and could reach to thousands.

2) If you take a 3D CAD model with tens of thousands components, then the possibilities for confusing the supportability detection is very high and might be not required.

3) The variation of the overlapping box detector (the dashed box in Fig. 4) depends on the extension factor around each component in the 3D CAD model, and could generate a number of undesirably supportability classifications.

In order to solve such supportability detection problems, the geometrical reasoning algorithm is simplified to find out only main and sub-supports. The implementation of this algorithm is illustrated below in Figure 4.



Column-Column Support Type	Column-Beams Support Type	Column-Beams Support Type	Foundation-Column Support Type
Adjacency Algorithm for Support Types			
(1) IF $P1 \leq P1c \leq P2$ AND $P1 \leq P2c \leq P2$ THEN Outside	(2) IF $P1c \leq P2 \leq P2c$ THEN North (3) IF $P1 \leq P1c \leq P2$ THEN East (4) IF $P1c \leq P1 \leq P2c$ THEN South (5) IF $P1 \leq P2c \leq P2$ THEN West	(6) IF $P1 \leq P1c \leq P2$ THEN North (7) IF $P1c \leq P2 \leq P2c$ THEN East (8) IF $P1 \leq P2c \leq P2$ THEN South (9) IF $P1c \leq P1 \leq P2c$ THEN West	(10) IF $P1c \leq P1 \leq P2c$ AND $P1c \leq P2 \leq P2c$ THEN Inside

Figure 4: Geometrical reasoning for adjacency relationships algorithm

2.3 Space Criticality Evaluation of an Execution Pattern

In this study we determine the space criticality of a specific execution pattern of activities *n* during monitoring period (weeks) D by evaluating the function $\underset{A}{f}(scr)$:

$$\underset{D}{f}(scr) = w1.\underset{D}{f}(co) + w2.\underset{D}{f}(r) + w3.\underset{D}{f}(no) + w4.\underset{D}{f}(st) + w5.\underset{D}{f}(cr) \ldots (5)$$

Where:

$f_A(scr)$: is the project space criticality.
$f(co)$: is the total conflicting space percentage.
$f(r)$: is the total space clashes ranking.
$f(no)$: is the total number of conflicting activities.
$f(st)$: is the total conflicting space types.
$f(cr)$: is the critical activities.
w: are the weights, and $w_1 + w_2 + w_3 + w_4 + w_5 = 1$

3. Simple Genetic Algorithm (SGA) Optimisation process

A simple genetic algorithm is developed based on the mechanics of natural selection and natural genetics evolution. A complete reference of Genetic Algorithms (GAs) heuristic random search techniques can be found in the work of Goldberg (1989). Applications of GAs in optimisation assume that the domain of problem can be presented by specific parameters within the genes of a chromosome. All individuals in a population compete for survival and have evolution analogy as in nature that is so called the Darwinian theory. SGA deals with the essence of natural selection process as follows:

- Start with an initial population using random number generator.
- Individuals who have better fitness values often survive and have great influence on new populations.
- Individuals generated in new populations are produced by crossover of parent's genes parameters (reproduction).
- Mutation comes after crossover with random change of genetic material.

3.1 Fitness Function

The SGA optimisation process evaluates the fitness function $f_{SGA}(scr)$ for each chromosome and performs a space criticality assessment of the specific execution pattern (see Eq. (6)). Furthermore, and through many generation runs, SGA should be able to find the best execution pattern for the assigned construction activities. Also, it should obtain the individual that possess the minimum conflicting space volumes (least space criticality) by applying minimisation to the following fitness function:

$$\text{Minimise } f_{SGA}(scr) = C - \max_{D} f(scr) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

Where:

C: is the maximum value of $f_A(scr)$ for this generation
$D = 1$ to number of monitoring dates (weeks)

3.2 Coding the Chromosome String

The chromosome structure for the best execution pattern for activities is represented in the string code. The chromosome parameters include each project activity and their assigned execution pattern in the string code. In Figure five step one, the chromosome shows five project activities with their execution patterns type (Babu and Babu, 2001). For instance, A1 represents the activity name and WE1 represents the execution pattern of type East-West-Access1 (Tsai et. al., 2001). Therefore, each chromosome (i.e. an individual) is programmed this way as a possible scenario for executing the site operations (i.e. best execution pattern).

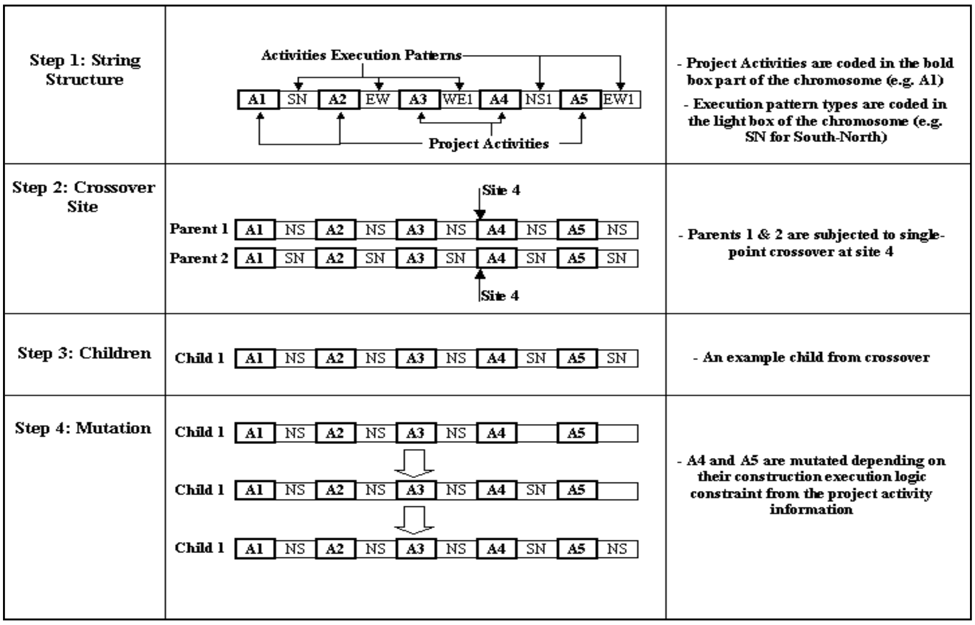


Figure 5: SGA Chromosome structure and operation

3.3 Initial Population Generation

Usually, SGA global search starts by randomly initialising a number of possible individuals in the initial generation. Based on previous research in SGA, the initial generation size ranges from 10 to 300. A small generation size of only ten individuals was used in this experiment in order to save computational time as it takes approximately one hour for each generation. The evolution process is stopped after predefined number of iterations and when a solution criterion is reached. The new generation is evaluated in the reproduction pool.

3.4 Reproduction of Individuals

A number of strings are selected from each generation for reproduction by using the Probability of Selection P_{select} mechanism for selecting the strings to the 'mating pool' (Goldberg, 1989).

3.5 Crossover Adjustment and Mutation

The SGA operates in correspondence with the natural life evolution scenario. A single-point crossover is applied on mates when a randomly selected probability value is set to 0.6 (see Fig. 5, Step 2). The result from the crossover operation produces a new execution patterns sequence for the construction project. The selection operator can be applied here to effectively create children solutions from parent ones (Fig.4, Step 3). For example, a single-point crossover is applied on parents 1 and 2 at site 4. It is assumed here that the invalid pairs selection resulting from crossover of the same parent are removed and considered as 'unhealthy ones' chromosomes. The solution children from crossover (strings) are mutated allowing only correct solution to remain in the reproduction pool. In other words, the mutation operator checks the activity execution pattern logic constraints. As illustrated in Step 4 of Figure 5, A4 is free of constraints and therefore accepts the new applied execution pattern of type SN. However, A5 is constrained (or follows) to A3's execution pattern of type NS; as a result from mutation, A5 is assigned similar execution pattern type as A3. It should be noticed here that the mutation of children only preserves the validity of the construction activities execution logic in whole.

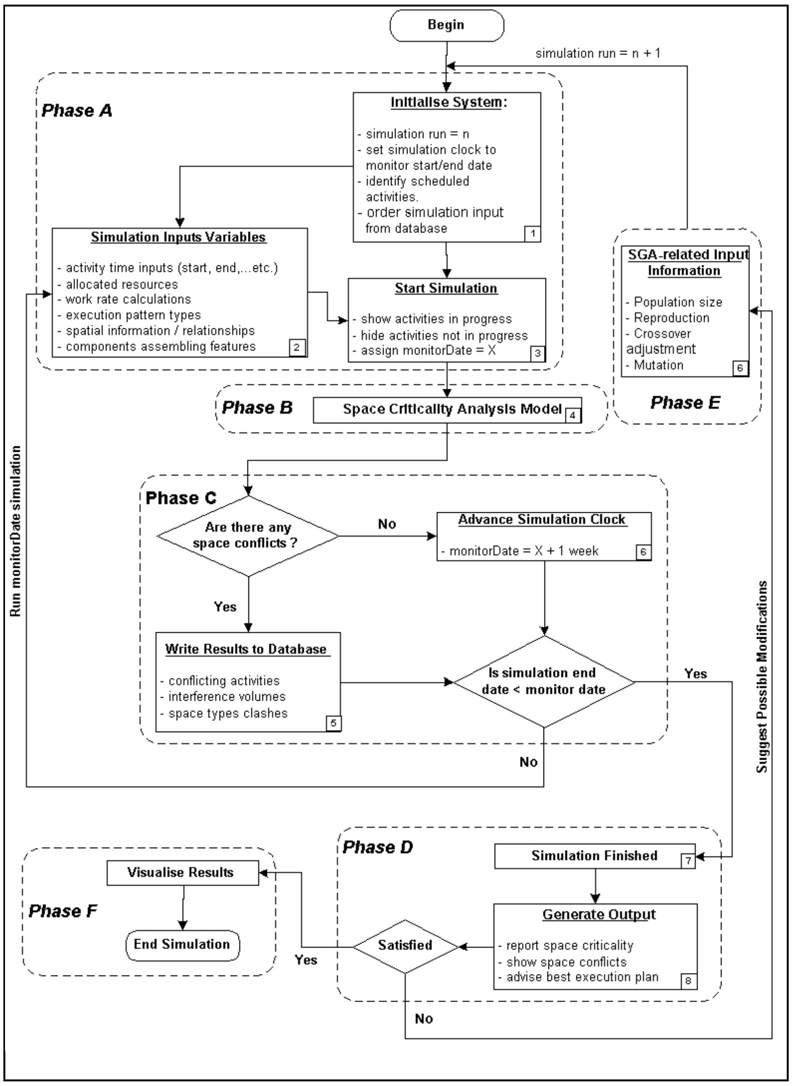


Figure 6: PECASO general simulation flowchart

4. Experimental Execution Patterns Simulation

The main objective in the PECASO system is to rehearse different 'what if' scenarios when given an actual project schedule information. The general simulation flow chart is shown above in Figure 6. One of the main simulation input variables is the activities start/end dates and duration coming from the MS ACCESS database 'tasks table' (Phase A). Furthermore, from the project planner's point of view, he/she utilises the PECASO system to solve the space-time conflicts by adding the execution logic rules as another variable for the simulation. When the 4D simulation is initialised, other variables are generically included during the simulation run time such as the quantity of work calculations. In this way, the amount of finished quantity of work and progressing work are estimated, and varies depending on the execution pattern type configuration.

In Phase B, the space criticality model detects and graphically maps the site space usage depending on the occupied areas by site operations. It is important to consider in the analysis the occupied spaces by the resources on site like plants, material paths, and storage areas. Followed by Phase C, the simulation results are written to the MS ACCESS database for future reference. The exported results during simulation monitoring dates include the total occupied space, the minimum conflicting volume, the number of conflicting activities, the conflicting space types (e.g. plant space conflict with activity process space) and the finished quantities of work. In Phase D, the system generates output report for the site planner in the form of charts to assess the project plan. SGA operation starts by experimenting with different execution patterns only when the output is not satisfied. A typical experimental illustration for minimising space conflict is shown bellow in Fig. 7 and applied on the School of Health project that we mentioned earlier. The simulation began with a max space criticality value of 108 representing the actual project schedule (refer to run No. (1)). The SGA optimisation method indicates a reduction of space criticality by %25 less than the original schedule (refer to run No. (5)). The reason for this minimisation is due to the alteration in execution pattern type for the Ground Flooring Concreting activity (North-South), while the rest of the activities were progressing from the West to the East. At the same time, the occupied space by the concreting plant moved to a space free of congestions and reduced the total number of conflicting space types *f(st).*

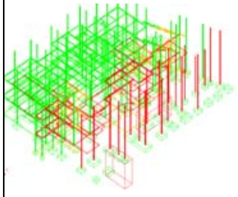
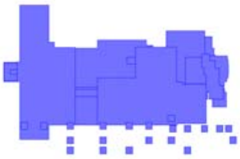
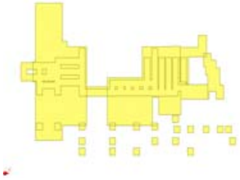
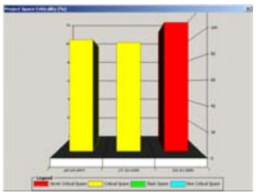
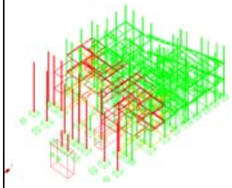
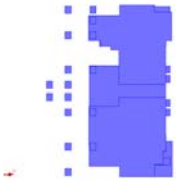
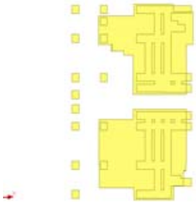
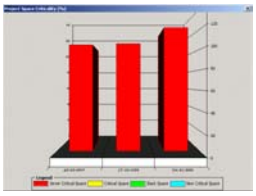
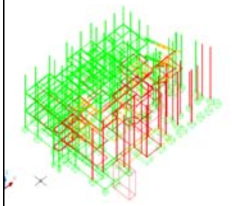
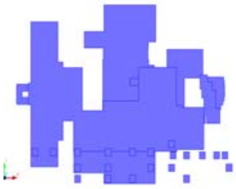
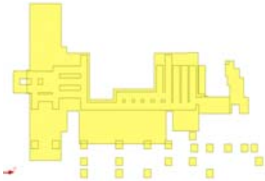
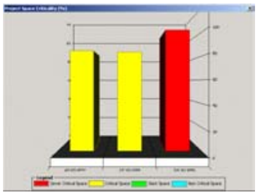
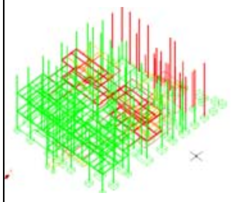
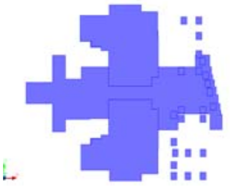
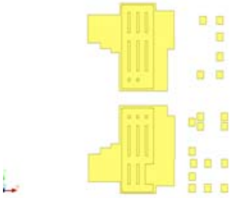
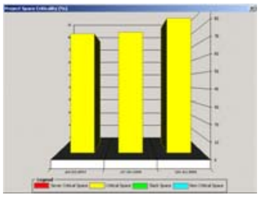
Run	**4D Visualisations**	**2D Site Space Usage**	**2D Site Space Conflicts**	**Critical Space Chart Report**
1				 Max. Criticality = 108
3				 Max. Criticality = 121
4				 Max. Criticality = 100
5				 Max. Criticality = 83

Figure 7: Experimental illustration of SGA space-time conflict minimisation

CONCLUSION AND FUTURE WORK

This has paper presented a methodology for modelling execution patterns of construction activities. The main semantics for an execution pattern have been explained and used in the 4D visualisation. One could argue that the advancements in 4D space-time conflict analysis relies on capturing the dynamic nature of construction site operations. Taking on this challenge, we identified a novel concept for space-time continuity in minimising space conflicts. To this, spatial strategies with generic spatial reasoning provided great flexibility for the SGA search when minimising the conflicted spaces. As shown in the experiment, the optimisation success depends on the alteration rules for the activities execution pattern. The SGA results suggest possible future use for the proposed optimisation technique in construction space planning, as the level of 4D realism is desired. The system can be extended to include random, top-down, spiral execution patterns that can be defined indirectly in the project schedule. It is anticipated that the inclusion of the developed generic spatial algorithms will increase the planner's strategic awareness for planning and they will becomes more confident when using 4D visualisation to communicate the construction programme of work with project team.

REFERENCES

Akbas, R. et. al. (2001). "Formalizing Domain Knowledge for Construction Zone Generation." *Proceedings of CIB w78 2001 Conference on Construction Information Technology*, 2001, South Africa, Vol. 1, pp 30-1/30-16

Akinci, B., Fischer, M., Levvitt, R. and Carlson, R. (2000). "Formalisation and Automation of Time-Space Conflict Analysis." *CIFE Working Paper No. 59*, June 2000, http://www.stanford.edu/group/4D/4D-home.htm

Anderl, R. and Mendgen, R. (1995). "Modelling with constraints: theoretical foundation and application." *Journal of Computer Aided Design*, 1996, Vol. 28, No. 3, pp 155-168

Babu, A. R. and Babu, N. R. (2001). "A generic approach for nesting of 2D parts in 2-D sheets using genetic and heuristic algorithms." Journal of *Computer Aided Design*, 2001, Vol. 33, pp 879-891

Cheng, M. Y. and O'Connor, J. T. (1996). "ArcSite: Enhanced GIS for Construction Site Layout." *Journal of Construction Engineering and Management*, December 1996, pp 329-336

Cheng, M. Y. and Yang, S. C. (2001). "GIS-Based Cost Estimates Integrating with Material Layout Planning." *Journal of Construction Engineering and Management*, January 2001, pp 291-299

Goldberg, D. E. (1989). *Genetic Algorithms in search, Optimisation and Machine Learning*. Addison-Wesley Publishing Company, Inc., New York

Goyal, R. K. (2000). *Similarity Assessment for Cardinal Directions Between Extended Spatial Objects*. PhD Thesis, The Graduate School, The University of Maine, Canada

Guo, S. (2002). "Identification and Resolution of Work Space Conflicts in Building Construction." *Journal of Construction Engineering and Management*, July/August 2002, pp 287-295

Kunigahalli, R., Russel, J. S. and Veeramani, D. (2002). "Extracting Topological Relationships From Wire-Frame CAD Model." *Journal of Computing in Civil Engineering*, January 2002, Vol. 9, pp 29-42

Mallasi, Z. and Dawood. N. (2001). "Assessing Space Criticality in Sequencing and Identifying Execution Patterns for Construction Activities Using VR Visualisations." *ARCOM Doctoral research workshop: Simulation and modelling in construction*, October 2001, Edinburgh University, UK, pp 22-27

Mallasi, Z. and Dawood. N. (2002). "Registering Space Requirements of Construction Operations Using Site-PECASO Model." *Proceedings of CIB w78 2002 Conference on Distributing Knowledge in Buildings*, 2002, Aarhus School of Architecture, Denmark, Vol. 2, pp 83-90

Mawdesley, M, Askew, W. and O'Reilly, M. (1997). *Planning and Controlling Construction Projects: the best laid plans*. Addison-Wesley Longman Publishing Company, Inc., England

McKinney, K., Fischer, M. and Kunz, J. (1998). "4D annotator: a visual decision support for construction planners." *Computing in Civil Engineering, Proceedings of International Computing Congress*, Boston, October 18-21, Kelvin C. P. Wang (Ed.), ASCE, pp 330-341

Riley, D. and Snvido, V. (1997). "Space Planning for Mechanical, Electrical, Plumbing and Fire Protection Trades in Multi-Story Building Construction." *5th ASCE Construction Congress*, Minneapolis, MN, pp 102-109

Riley, D. and Snvido, V. (1997). "Space planning method for multi-story building construction." *Construction Engineering and Management*, Vol. 123, No. 2, pp 171-180

Thabet, W. Y and Belivau, Y. (1997). "ScaRC: Space-Constrained Resource-Constrained scheduling system." *Journal of Computing in Civil Engineering*, Vol. 2, No. 1, pp 48-59

Tsai, Y., Wang, K. and Teng, H (2001). "Optimizing preventive maintenance for mechanical components using genetic algorithms." *Reliability Engineering and System Safety*, No. 74, pp 89-97

Xu, J and AbouRizk, S. (1999). "Product-Based Model Representation for Integrating 3D CAD With Computer Simulation." *Proceedings of the 1999 Winter Simulation Conference*, USA, pp 971-977