Grid Cell-based Algorithm for Workspace Overlapping Analysis Considering Multiple Allocations of Construction Resources

Hyounseok Moon[1], Vineet R. Kamat[2] and Leenseok Kang*[3]

[1] Senior Researcher, Ph.D., ICT Convergence and Integration Research Division, Korea Institute of Construction Technology, Korea
[2] Associate Professor, Ph.D., Department of Civil and Environmental Engineering, University of Michigan, USA
[3] Professor, Ph.D., Engineering Research Institute, Department of Civil Engineering, Gyeongsang National University, Korea

Abstract

When multiple workspaces for resources are allocated within a narrow work area, the constructability of work may deteriorate based on the conflicts that occur due to the mutual adjacency levels between the workspaces. In this situation, the identification of the workspace overlapping status is essential for establishing the optimized layout planning of workspaces. However, current 3D environments consider only physical collision and reaction by the cross verification of geometric properties to detect collisions between 3D objects. Thus, the configuration of methodology to visually check the overlapping level of workspaces with conflicting status is insufficient for a successful interference management. This study develops a new grid cell-based analysis algorithm and its 4D simulation system for visually verifying the workspace overlapping of dynamic resources to be allocated multiply within a confined area, such as a tunnel or underground facility. The proposed algorithm and system can assist in the intensive management of activities that have a high level of workspace overlap by visualizing their overlapping level.

Keywords: Bounding box; Workspace overlapping; Grid cell-based algorithm; 4D CAD; BIM (Building Information Modeling)

1. Introduction

A workspace is considered as a critical management factor for the safe execution and constructability of a project. In existing research, workspace conflicts are checked by time-space trade-offs with object space changes according to fixed periods of time. Besides, overlapping relationships with neighboring working areas are judged by their own mathematical approaches to estimate the overlapping relationships and their levels on a horizontal plane. However, there might be limitations in providing methodologies that analyze and visually identify congested statuses using the levels of overlapped areas in a 3D environment. In particular, because an individual activity requires the input of diverse resource types, a new allocation method for workspaces where many resources can be assigned to a single activity should be developed.

This study aims to build a workspace generation and allocation system for the resources required for each activity and to develop an algorithm and its simulation system to verify and manage overlapping amongst workspaces for resources by configuring a grid cell-based workspace overlapping analysis algorithm (G-SOAA) with a projection function. This study attempts to develop a workspace generation feature with a safe space, a workspace overlapping visualization algorithm by a grid cell concept using a projection algorithm, and a dynamic simulation system based on 4D CAD. The proposed system helps managers to obtain visual information to draw optimized design alternatives by considering the workspace layout for resources in the design stage. In addition, it is expected that the developed system will provide dynamic decision-making information to improve the constructability of a project by establishing quick layout strategies for the workspaces with a real-time analysis of the mutual overlapping relationships amongst workspaces in the construction stage.

2. Literature Review and Research Contributions

2.1 Literature Review

Recently, with the advancements in 3D and 4D CAD techniques (McKinney *et al.* 1996; Kang *et al.* 2010) with the real-time progress management (Kim *et al.* 2009), diverse methodologies for allocating workspaces into each activity and for managing the results have been attempted. By creating 3D workspace shapes, the execution status of the workspace has also been reviewed in a virtual environment. Guo (2002)

*Contact Author: Leenseok Kang, Professor,
Department of Civil Engineering, Engineering Research Institute, Gyeongsang National University, 900, Gajwa-dong, Jinju-si, Gyeongsangnamdo, South Korea
Tel: +82-55-753-1713 Fax: +82-55-753-1713
E-mail: Lskang@gnu.kr
(Received April 15, 2013 ; accepted February 26, 2014)

configured the workspace shape based on 2D plan-view envelops elements. Akinci (2002b, 2002c) generated a 3D workspace model required for the construction of elements and presented the layout methods in an available working area using the generated workspaces. In addition, Tantisevi and Akinci (2007) configured a 3D bounding box model as a moving space with the rotation area of a boom when hoisting materials with equipment, such as a crane.

With the workspace shape, a variety of collision detection techniques have been applied to identify physical conflicts between neighboring equipment and workspaces. Generally, diverse theories for the physical collision status by cross analysis between geometric properties and its reaction relationships have been introduced by AABB (axis-aligned bounding box), OBB (object-oriented bounding box), OBB-tree, k-DOP (discrete orientation polytope) (Held *et al.* 1998), and spatial division trees (Lengyel 2011), such as Octree and BSP (binary space partitioning) trees. The mutual collision between workspaces is based on geometrical parameters with the coordinates of objects (Lin and Gottschalk 1998). Kamat and Martinez (2005) utilized C-Collide and VITASCOPE (VIsualizaTion of Simulated Construction OPEration) software (Kamat 2003) using the collision detection method by OBB and OBB-tree (Gottschalk *et al.* 1996) to verify mutual physical collisions according to the movement of equipment. In addition, there are many physical detection systems, such as RAPID (Robust and Accurate Polygon Interference Detection) (Gottschalk *et al.* 1996), I-COLLIDE (Cohen *et al.* 1995; Gottschalk *et al.* 1996), and V-COLLIDE (Hudson *et al.* 1997). In the field of workspace planning, researchers have applied their own algorithms that generally estimate a volume and its degree of overlap to analyze overlapping and conflict amongst workspaces. Akinci *et al.* (2002a) checked the geometric cross status for the overlapped area between construction workspaces, which are rectangular prisms, based on the 3D shape and identified the conflicted status by checking the volume of their conflicted area. Dawood and Mallasi (2006) extracted the corresponding area with a cross check of the conflicted parts to check the mutual conflict level of the grouped workspace area and applied the CSA (Critical Space Analysis) approach to quantify and visually identify the conflict degrees of such areas.

2.2 Research Contribution

Existing researches have configured methodologies that verify a simple physical collision using the geometric properties of 3D objects. Due to difficulties of coding programs for the collision check algorithms, they mainly utilize collision detection libraries from a game physics engine that has already been developed.

The existing researches consider a workspace as a physical execution area of a single element rather than an operational task level. Because it is difficult to allocate multiple workspaces into a single activity at a time, there are limitations in performing the multiple layout management of resources and reviewing the overlapping range by the level allocated to each activity. Moreover, it is difficult to generate concurrently multiple workspaces with buffer spaces for all allocated resources because existing workspace shapes are mainly generated based on a single type with 2D drawings, a 3D approximate envelop, and a 3D solid model for a physical element of each activity. To address these issues, this study develops a new workspace overlapping visualization algorithm and its simulation system, which enables the managers to establish an efficient workspace layout by the unit resource type and a process planning by workspace overlapping based on a multiple allocation strategy for a single activity. The contributions of this study can be elaborated in the following cores.

1) A grid cell-based workspace overlapping analysis algorithm (G-SOAA): This study suggests a grid cell-based workspace overlapping analysis algorithm, which is a new concept that enables managers to visually identify the degree of overlapping levels according to the overlapped status between workspaces for resources. This algorithm assists managers not only in identifying entire overlapping levels of the workspaces of allocated resources for each activity but also in establishing reasonable allocation planning of the workspaces according to the workspace layout status compared to existing algorithms, which represent only the physical collision status between 3D elements during equipment operation.

2) BIM-based workspace overlapping visualization with 4D simulation: To visualize the workspace overlapping levels by each activity, this study attempts to develop a 4D simulation system that enables managers to dynamically review the overlapping status of the required workspaces for each activity. In particular, if multiple workspaces are required, the users can easily add workspaces and visually identify the summarized overlapping results for a single activity in a daily unit during the simulation.

3) A 3D bounding box model with buffer space for workspaces: As a representation model of such multiple workspaces, a workspace modeling method by a 3D bounding box with buffer space, which includes a safety area for work execution of resources, is configured.

The suggested algorithms enable managers to sum the overlapping level for many workspaces to the upper level of a unit activity. Therefore, because the conflict degree is visually categorized according to the overlapping level of workspaces, this study provides a visual decision-making model based on 4D CAD that helps to establish reasonable layout strategies for the workspaces.

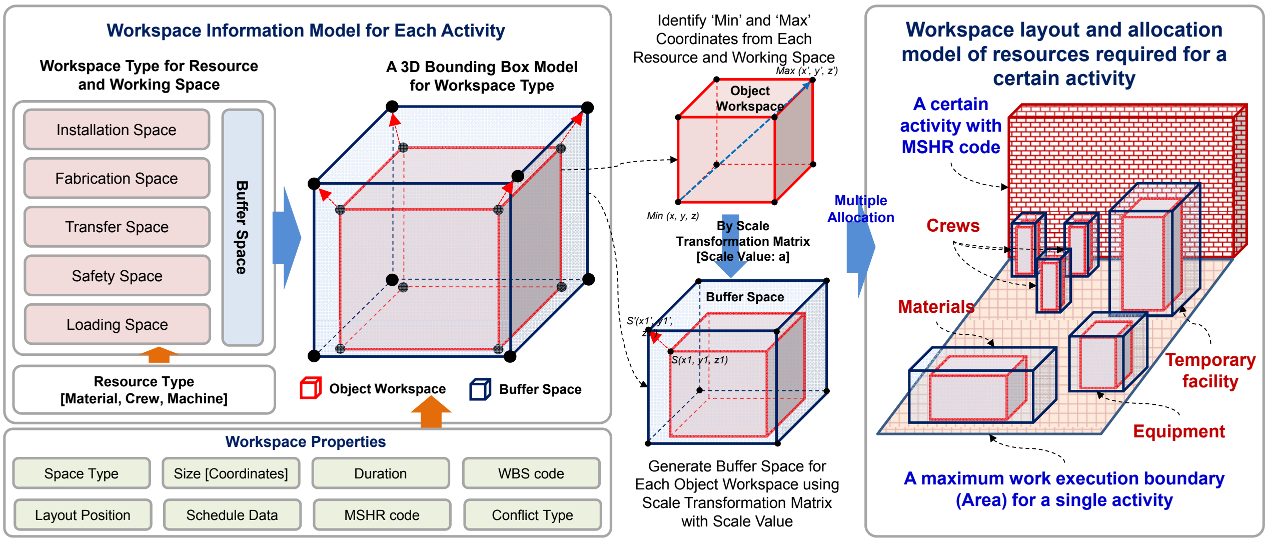


Fig.1. Workspace Information Model of Resource Data for Each Activity

3. Workspace Generation and Allocation Model for Resources

3.1 Information Model of a Workspace for a Resource Type

Generally, a workspace refers to an execution space for the physical work of tasks and resources for the construction process. The workspaces are divided into 5 types, installation, fabrication, safety, transfer, and loading space, by referring to common parts that are grouped according to the existing research (Moon *et al.* 2010). Fig.1. shows the information model of the workspace shape required for each activity.

A workspace shape with resource types utilizes an AABB model, which has been widely applied to check for the physical object collision in 3D game theory. The existing workspace shapes were based on 2D drawings for the space area of enveloped elements, the grouped single area of multiple spaces as an approximate envelope (AE), and a 3D solid model on a specific plane for manufactured elements with a construction method. Because such methods have limitations in generating workspace models in terms of an automatic extraction of the unit resource size and visual identification, many parameters from the existing methods are required to create an appropriate workspace shape. However, the workspace shape by the AABB model can be easily generated with simple parameters, the resource size and its type, and enables an intuitive verification of workspace overlapping in a simple manner.

Configuring the workspace shapes requires two types of information, such as the properties and geometric parameter data of the workspace. The workspace properties include the WBS (Work Breakdown Structure), MSHR (Multiple Space Hierarchy for Resources) code, schedule, and 3D element model for each activity by linking the schedule with the workspaces. The geometric parameter data consist of the center location of a workspace, the node coordinates (X, Y, Z) of object spaces for creating the workspaces and their buffer space, the scale value to determine the shape of the buffer space, the size of the workspace (width, length, height), the layout location and the directions, which are determined by a spatial topological relationship. Their adjacency level is judged by the base coordinates set along with the layout of the critical node points and the center coordinates.

In the suggested 'workspace properties', space type can be identified from a MSHR code. The size is critical for 3D workspace shapes and its buffers, and is identified within each 3D workspace shape. Duration is also checked in WBS. Layout positions are identified when the workspaces are located on a grid. All these data can be identified from a 4D model because the workspace properties are managed and controlled by the 4D model. Besides, these data are also shown on a database table by referring to mutual properties.

Space type is an important property in identifying what the workspace types are, and determining workspace conflict types. Besides, duration is an important factor in checking continuous status of the workspace and its conflict. The WBS code is especially crucial in controlling and managing all the properties of the workspace. Layout position is critical in analyzing the workspace conflict. Of note, schedule data is significant for 4D simulation. The MSHR code is important in recognizing what workspace type is with conflict type.

Basically, all workspace layout boundaries, which are a grid layer, are made for each WBS code. Therefore, workspace properties should be defined for each workspace layout boundary with multiple workspace types. If there are many activities, since their properties should be inputted separately for all the work packages, an automated input mechanism for their properties needs to be developed. Characteristics and requirements for defining eight 'workspace properties' values for each workspace type are different. However, even if the WBS work package is different, all the workspace properties should be basically included for each activity with WBS.

3.2 Generation of the Workspace Shape with a Buffer Space for Resources

The workspace shape and its size depend on the inputted resource type, size and its layout location. The size of inputted resources should be identified first, and then an initial workspace shape is generated. These values are determined by the eight nodes of the initial workspace by identifying the minimum and maximum points for the shapes of each resource. In addition, when these coordinates are connected mutually by geometric parameters in a 3D environment, a bounding box type of a workspace is generated to fit into the shape of the 3D object of the individual resource. For example, by multiplying any point '*Pn*' (*n*: 1~8) of 8 nodes by a scale a separately, a point P1 is calculated using the formula below. That is, this point represents $P' = a \times P$, and formula (1) below expresses a matrix for the scale transformation of $P_1(x_1, y_1, z_1)$.

$$S_v P = \begin{bmatrix} P_x' \\ P_y' \\ P_z' \end{bmatrix} = a \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} aP_x \\ aP_y \\ aP_z \end{bmatrix} = \begin{bmatrix} v_x = a & 0 & 0 \\ 0 & v_y = a & 0 \\ 0 & 0 & v_z = a \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (1)$$

$$where\ v_x = v_y = v_z = a \text{ (Uniform Scaling Value)}$$

This formula is used to apply the same scale value (a) in all x, y, z directions. Then, the new extended coordinates (x', y', z') are drawn by the scale value with each node coordinate, which is (x, y, z). For example, if the coordinate of P_1 (x, y, z) yields P_1 (4, 5, 4) and the scale value is 1.5, the new extended coordinate of $P_1'(x', y', z')$ becomes $P_1'(6, 8, 6)$ according to formula (1).

3.3 Layout Allocation of Multiple Workspaces for a Single Activity

The workspace types are configured according to the inputted resource type and the execution characteristics of the work. However, it is difficult to identify which workspaces are being shared for a given activity and what types of workspaces are required. This problem can be solved by specifying a MSHR code that configures an efficient allocation system and manages the shared data of workspaces for each activity. The MSHR refers to a grouped workspace hierarchy for allocating and managing the integrity of multiple resources in a limited working area. The MSHR is an essential management factor to configure what workspaces are being allocated and operated for a certain activity. The MSHR is configured based on WBS, which includes the 'Facility-Element-Activity' facet (Kang and Paulson 1997).

The single code can provide a flexible extension function that allows for the adding of the different space types required by each activity, which can also be utilized as an information centric basis for allocating multiple workspaces (N) of resource objects concurrently into a single activity (1) and for tracing and managing the generated workspaces in a virtual environment quickly. In addition, it is possible to sum the entire degree of overlap between allocated workspaces to the upper-level code of the MSHR. Therefore, it is expected that this method can be applied to establish a reasonable workspace planning that enables the users to cope with unexpected changes in the workspace number, size, and type during the construction stage.

4. Grid Cell-Based Workspace Overlapping Analysis Algorithm

4.1 Configuration of a Virtual Projection Grid Layer

The suggested algorithm is a new method for determining the physical overlapping status visually with only the add operation between overlapping cells projected onto grid layers without requiring a complex mathematical calculation. This algorithm was developed to minimize the complexity of the calculation by the complicated mathematical model and the difficulties of system development. This method is also configured by extending the cell grid concept (Zhang *et al.* 2007) of spatial items to the workspace overlapping check algorithm. The virtual projection grid layer is the target region where multiple workspaces are projected and is configured as a visual representation area of the overlapping status of workspaces with mutual grid cell matching between the divided cells.

Fig.2. represents an initialization status of cells and a configuration of a virtual grid layer, which is defined as an intermediate space area to check overlaps.

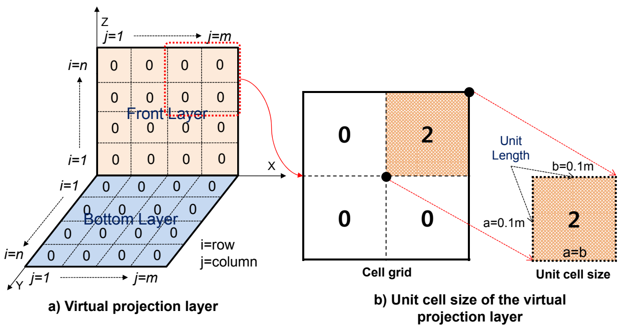


Fig.2. Definition of a Grid-based Virtual Projection Layer

Each array cell value in row i and column j in both the front and bottom grid layers is initialized with a '0' value to be recognized as an empty space. This value is generated in a virtual memory area and consists of a bottom and front projection grid layer on which workspace shapes are projected. In the case of projection onto the bottom virtual projection grid layer, the layer is a 2D-based virtual memory area of a target plane of projection where the Z coordinate value is '0'. In the case of projection of workspace models onto the front virtual projection grid layer, the layer also represents a 2D-based virtual area of a target plane of projection where the X or Y coordinate value becomes '0'. Therefore, by using an area projected onto each grid layer and the height value of the overlapping spaces, the overlapping volume to judge the overlapping level is calculated.

4.2 Configuration of the Orthographic Projection Algorithm

The bottom and front grid layers of the generated workspace must be divided into many cells with a quadrate unit to judge whether there are any overlaps by the cell operation after projecting the workspace and its buffer space. The unit cell with '0.1m × 0.1m' of the bottom and front grid layers for each workspace shape should be defined as the same cell size as the virtual projection grid layer.

To check the overlapping status of the bottom and front layers of the generated workspace on the projection plane, a proper projection algorithm should be applied. The projection algorithm utilizes an orthographic projection method where the angle to be projected is perpendicular to the projection plane when projecting in parallel. Fig.3. shows a status where the bottom and front grid regions of the workspace are projected onto the XY and X(Y)Z planes, respectively.

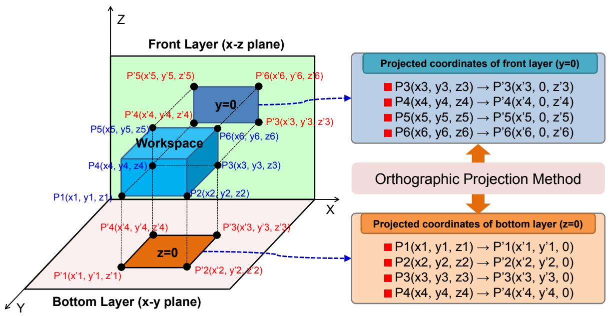


Fig.3. Orthographic Projection Method of the Bottom and Front Faces for Workspace Models

$Bottom\ Layer\ Projection(x - y\ plane)$:

$$Pv = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} v \qquad (2)$$

where $v = (v_x, v_y, v_z),\ v_z = 0$

For a projection algorithm for the bottom layer, an algorithm for projecting the node points of the bottom plane of the workspace with buffer space onto the XY plane (z = 0) is defined. As shown in Fig.3., the coordinate values of the bottom plane, which consist of 4 node points with P1, P2, P3, and P, are converted into P′1, P′2, P′3, and P′4 separately. If any points (v_x, v_y, v_z) on the bottom plane are projected orthogonally onto the XY plane, all Z values of the coordinates projected by formula (2) are converged with a '0' value and the final projected node points are converted into $(v_x, v_y, 0)$. In addition, for a projection algorithm for the front layer, an algorithm for projecting the node points of the front plane of the workspace with a buffer space onto the XZ plane (y = 0) is configured. If any points (v_x, v_y, v_z) on the front plane are projected orthogonally onto the XZ plane of a virtual projection grid layer, the final projected node points are converted into $(v_x, 0, v_z)$ with a '0' value for Y coordinates.

4.3 Cell Calculation and Judgment by Projection of the Overlapped Workspaces

Fig.4. shows overlapping types and its visual status of the bottom and front planes of two workspaces overlapped by the orthographic projection algorithm and grid matching method.

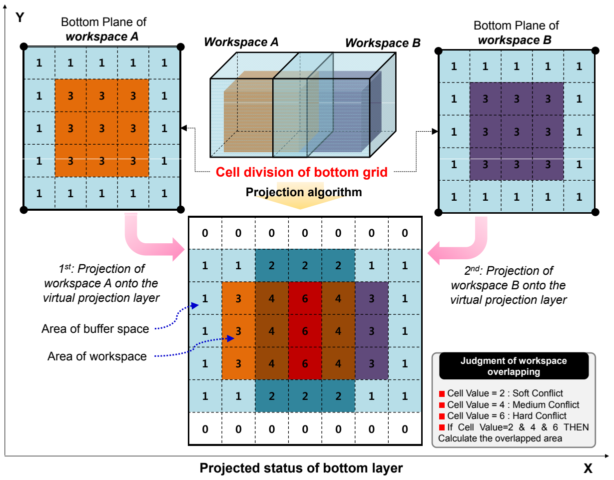


Fig.4. Overlapping Cell Calculation and its Judgment between the Workspaces A and B Projected onto the Virtual Projection Grid Layer

Here, in the case where a buffer space and the other buffer space are overlapped together, the grid cell unit of the bottom plane of each buffer space is set as '1' before projecting the workspaces. When the overlapping calculation is performed after projecting the buffer spaces of the two workspaces, the corresponding overlapped cell is estimated with '2' (= 1+1).

When the buffer space and a workspace object are projected onto a virtual projection grid layer, the cells for the buffer space are set to '1' and the cells for the workspace set to '3' on the grid layer. In addition, when a workspace and other workspace parts are concurrently projected on the virtual grid layer, all grid cells of the bottom plane for two workspaces are set to '3'. If the overlapping between two workspace areas also occurs, the cell areas projected onto the virtual grid layer are calculated with '6' (= 3+3). The total area of the overlapped section is estimated by the 'area of a unit cell of the bottom grid layer (m^2)×the number of the overlapped cells (N)'. For example, if the width and length of the unit cell are 1.0 m and 1.0 m, respectively, its area is 1.00m^2. Accordingly, if the number of overlapped cells is 6, its area is 6.00 m^2 (= 1.00 m^2 × 6). This figure is utilized as a base value for a bottom plane area to calculate the overlapping volume using the overlapping height of the workspaces from a front virtual grid layer.

In addition, the overlapping level of the workspaces conflicting with the base space is determined by the ratio of the area of the overlapped cells for the base space to the total area of cells for the base space. That is, if the total area of the bottom plane in the base space is 25 m^2 and the area of the overlapped cells is 15 m^2, the overlapping ratio of the bottom plane of the base space is 0.6 (= 15/25). This value is utilized as a standard for visually identifying the status where the overlapping is as high as 60% of the area of the base space.

5. Development of a 4D CAD-based Workspace Overlapping Analysis System and Case Study

5.1 System Architecture and Case Study

This study developed a workspace overlapping visualization system based on 4D CAD to dynamically verify workspace-overlapping levels based on the suggested methodologies. This system is called 4D-WCD (4D Workspace Conflict Detector). The system consists of a workspace generation module, a workspace allocation module, a workspace overlapping analysis module, and its 4D simulation module. Fig.5. shows a system architecture that includes the main algorithms to be embedded within the system, functional modules, and interaction system between the modules with data.

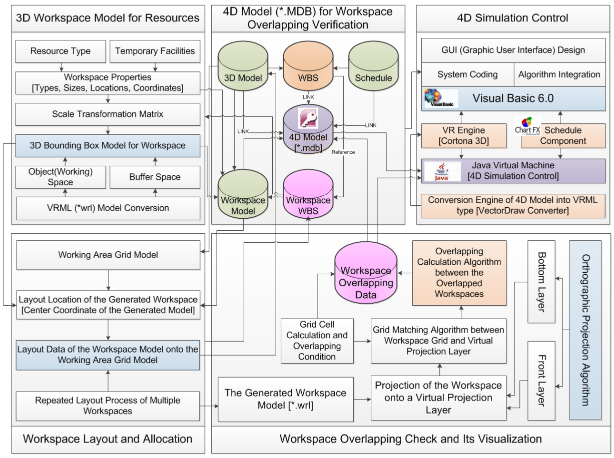


Fig.5. System Architecture of 4D-WCD

This chapter performs a case study for a 'Honam express railway' project in Korea, which includes a narrow working area, limited construction boundary located in a mountain area, and high frequency of workspace overlapping compared to the other projects.

First, a project WBS code is configured from a 4D model of the tunnel as a standard code for workspace generation. Then, a MSHR for the tunnel project is configured by recognizing what workspaces are created by the MSHR for each activity. Fig.6. represents a generation and allocation process of workspaces based on the MSHR.

1) Generation of the MSHR: A target activity in this tunnel project is 'Lower half section excavation' as a project WBS. After selecting this WBS code, the MSHR is generated through a pop-up menu. The MSHR, which includes 'Lower-half section excavation', is stored in the database, and the final MSHR tree is configured. Once the process is completed, the users can identify the MSHR code with 'SPA_100_120_122' as a '120_Lower-half section excavation_SPACE' item.

Here, the MSHR for the workspaces allocated to WBS code (100_120_122) of the 'Lower-half section excavation-2nd step' activity in a tunnel project is defined by automatically declaring the 'SPA' prefix name before the corresponding code. If 'Lower 2nd half section excavation_SPACE', which is 'SPA_100_120_122', requires installation (SPI) and loading space (SPL), the Workspace WBC codes, which consist of 'SPI_100_120_122' and 'SPL_100_120_122' respectively, are generated separately in the lower level of the 'SPA_100_120_122' code. The code name indicates that the top level includes all sub-workspace types for each available working area.

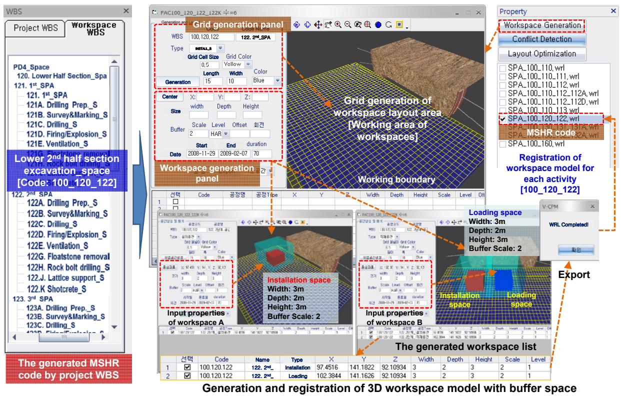
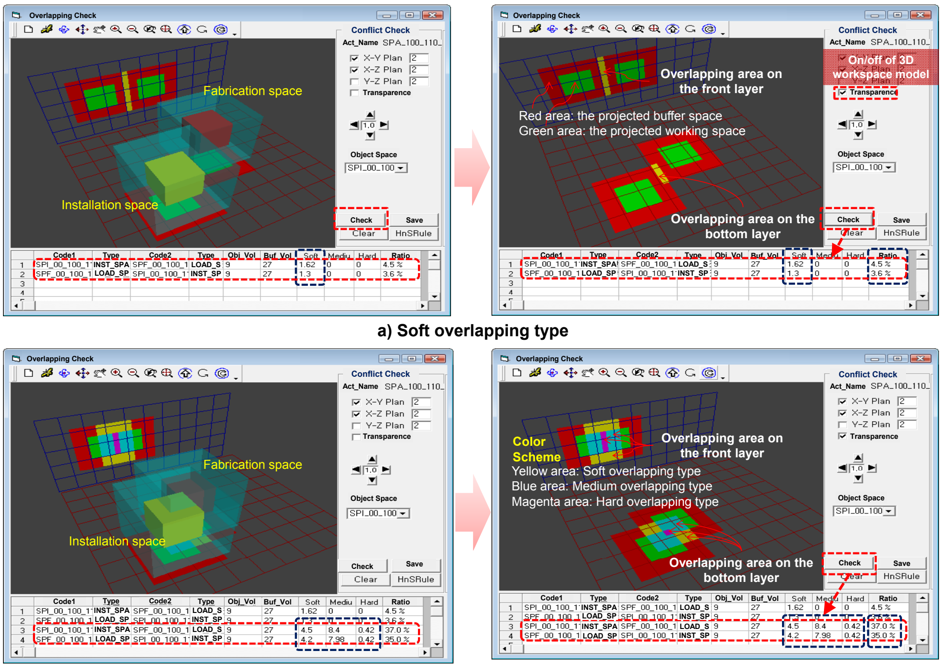


Fig.6. Generation and Registration of a 3D Workspace Model for the Activity

2) Generation and layout of the workspace: Once the MSHR is generated, a grid for the maximum work execution boundary in which the workspaces are arranged is generated. Next, the type, size and layout location of the workspace should be determined by users. The shape for required workspaces is modeled by defining the size and scale values of each workspace and its buffer space. Their values are inputted by users considering maximum size and safety region of resources to be located onto a grid layer. Subsequently, to assign the generated workspaces to the work execution boundary, the spatial center coordinate where the workspaces are located should be extracted in advance. Referring to these coordinates, the center coordinates with clicking points by a mouse are matched with the coordinates of the workspaces. Then, the layout process is performed, and its layout status is identified on the screen.

3) Selection and trace of the workspace by activity: The workspace created based on the MSHR is built by specifying the workspace types for a single work execution boundary at a time. Subsequently, the



a) Soft overlapping type

b) Hard overlapping type

Fig.7. Overlapping Visualization of Two Workspaces and its Overlapping Type

workspace type and its overlapping status are identified. In addition, the integrated management of multiple workspaces can be performed by generating the workspaces of diverse resources required for each activity and managing the workspaces as a spatial model. Fig.7.(a) represents a screen for visually identifying the overlapping status for the buffer space of the two workspaces. The overlapping regions are visually identified with a color scheme according to the overlapping level by performing a calculation process internally for the overlapping analysis algorithm.

Accordingly, the overlapping status and its level can be identified, and the overlapping level with numerical data is checked. After projecting, the workspace region is represented in green and its buffer space is shown in red. In addition, if the overlapped region is shown in yellow, there is a soft overlapping between the buffer spaces. That is, the cell value of an overlapped region is calculated with '2', and the value is inputted into an algorithm for visualizing the overlapped region in yellow. However, in Fig.7.(b), if the color of the overlapped region is magenta (color code=6), its overlapping type is 'Hard type', which occurs due to an overlap between workspace regions. If the overlapped region is shown in cyan (color code=4), its overlapping type is 'Medium type', which occurs due to an overlap between the workspace region and buffer space. For these reasons, the overlapping status is judged by defining the color codes according to the corresponding overlapping region.

5.2 4D Simulation of Workspace Overlapping Analysis

Fig.8. shows a screen of the 4D simulation for visually identifying the overlapping status of workspaces over time by integrating the overlapping results with the schedule.

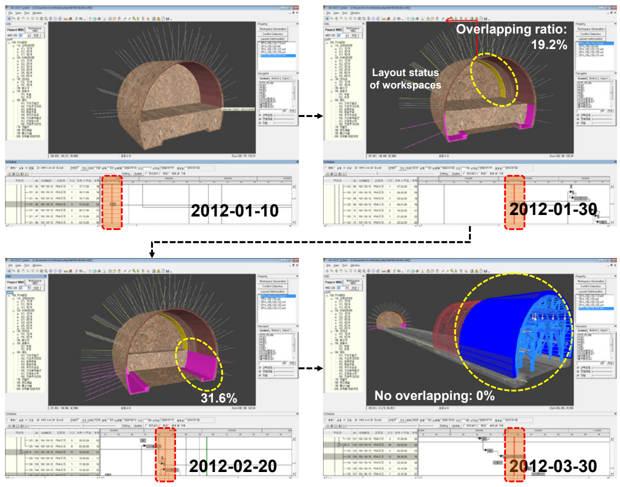


Fig.8. 4D Simulation of the Workspace Overlapping Status

In the upper part of the right side of Fig.8., the ratio value is inputted into the 3D object property of an element because the overlapping ratio of corresponding activity is calculated to be 19.2%. Because this value is less than 30%, its overlapping status of the activity is shown in yellow. In the lower part of the left side of Fig.8., the overlapping ratio of the 'SPA_100_120' code, which represents all workspaces assigned to the '100_120' activity, is 31.6%, and the color of the 3D element is shown in magenta. Additionally, if it is greater than 61%, the color of the activity is red. If there is no overlapping, a 3D element for corresponding activity is calculated to be 0%, and its color is blue. With this ratio, the overlapping level can be visually identified over time, and the target activities for critical control can be intensively reviewed according to the level of overlap.

5.3 Analysis Results of the Case Study

The generation process of a workspace model and visualization of workspace overlapping were verified through a case study, and its 4D objects were simulated with the case project. In this project, the generation methods for the diverse workspaces required for each activity in the tunnel were reviewed. The level with a change status of the workspace overlapping was reviewed dynamically over time by reflecting the level into 3D elements linked with the activities. Accordingly, an intensive management of workspaces for activities that have a high overlapping level was possible by tracing such activities quickly. It was also shown that a layout plan of the workspaces could be established efficiently through a workspace allocation module by the MSHR.

To further enhance the practical applicability of the developed system, the suggested algorithms should reflect the site conditions sufficiently. Besides, because the overlapping of the rotating workspaces could not be checked, the improved overlapping check algorithm should be presented.

6. Discussion and Conclusions

This study developed a grid cell-based workspace overlapping analysis algorithm with an allocation method with MSHR code and a 4D simulation to dynamically check the workspace overlapping levels between resources.

Unlike the existing method, the algorithm minimized the usage of geometric parameters. By using a projection algorithm and operation process between the overlapping cells by a cell grid-matching algorithm, a new algorithm to check workspace conflict and identify its level visually according to the overlapping status was configured. In addition, a workspace conflict verification system to review the overlapping status of the workspaces arranged on the work execution boundary by each activity was developed in the study. Besides, this process was verified by a tunnel project. Generally, considering that the tunnel project has a limited workspace and many conflicted activities, a project manager could easily check the workspace overlapping by the proposed projection algorithm.

The developed 4D CAD system was useful for visualizing workspace overlapping through a case study. Even if the system was verified with one project, the practical applicability could be expected through the case study because the tunnel project consists of the complicated workspaces in a limited work area. This system includes functions for the proposed searching algorithms for conflicted workspaces. Moreover, the system has also a simple schedule simulation function, which will be used as a decision making tool for efficiently establishing a workspace layout planning with a workspace overlapping detection.

Acknowledgements

This study was financially supported by the National Research Foundation of Korea (R&D No. 2011-0016064).

References

1) Akinci B., Fischer M., Levitt R., and Robert C. (2002a) Formalization and Automation of Time-space Conflict Analysis. Journal of Computing in Civil Engineering, 16 (2), pp.124-134.
2) Akinci B., Fischer M., Kunz J., and Levitt R. (2002b) Representing Work Spaces Generically in Construction Method Models. Journal of Construction Engineering and Management, 128 (4), pp.296-305.
3) Akinci B., Fischer M., and Kunz J. (2002c) Automated Generation of Work Spaces Required by Construction Activities. Journal of Construction Engineering and Management, 128 (4), pp.306-315.
4) Cohen J., Lin M., Manocha D., and Ponamgi K. (1995) I-COLLIDE: an interactive and exact collision detection system for large-scaled environments. Symposium on Interactive 3D Graphics, pp.189-196.
5) Dawood N. and Mallasi Z. (2006) Construction Workspace Planning: Assignment and Analysis Utilizing 4D Visualization Technologies. Computer-Aided Civil and Infrastructure Engineering, 121, pp.498-513.
6) Gottschalk S., Lin M.C., and Manocha D. (1996) OBB-tree: a hierarchical structure for rapid interference detection. The 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp.171-180.
7) Guo S. J. (2002) Identification and Resolution of Work Space Conflicts in Building Construction. Journal of Construction Engineering and Management, 128 (4), pp.287-295.
8) Held, M., Mitchell, J. S. B., Sowizral, H., and Zikan, K. (1998) Efficient collision detection using bounding volume hierarchies of k-DOPs. IEEE Transactions on Visualization and Computer Graphics, 4 (1), pp.21-36.
9) Hudson T.C., Lin M. C., Cohen J., Gottschalk S., and Manocha D. (1997) V-COLLIDE: Accelerated collision detection for VRML. The Second Symposium on Virtual Reality Modeling Language, pp.117-123.
10) Kamat V. R. (2003) VITASCOPE: Extensible and Scalable 3D Visualization of Simulated Construction Operations. Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
11) Kamat V. R., Martinez J. C. (2005) Efficient interference detection in 3D animations of simulated construction operations. The International Conference on Computing in Civil Engineering, ASCE, Cancun, Mexico 2005.
12) Kang L. S. and Paulson B. C. (1997) Adaptability of Information Classification Systems for Civil Works. Journal of Construction Engineering and Management, 123 (4), pp.419-426.
13) Kang, L. S., Moon, H. S., Park, S. Y., Kim, C. H., and Lee, T. S. (2010) Improved Link System between Schedule Data and 3D Object in 4D CAD System by Using WBS Code. KSCE Journal of Civil Engineering, 14 (6), pp.803-814.
14) Lengyel E. (2011) Mathematics for 3D game programming and computer graphics. Third edition, Course Technology PTR.
15) Lin M.C., Gottschalk S. (1998) Collision detection between geometric models: a survey. IMA Conference on Mathematics of Surfaces.
16) McKinney, K., Kim, J., Fischer, M., Howard, C. (1996) Interactive 4D-CAD. Proceedings of the Third Congress on Computing in Civil Engineering, Jorge Vanegas and Paul Chinowsky (Eds.), ASCE, Anaheim, CA, June 17-19, 1996, pp.383-389.
17) Moon, H. S., Kang, L. S., and Dawood, N. (2010) Development of a Workspace Conflict Verification Model for Temporary Facilities based on a VR Simulation. The International Conference on Computing in Civil and Building Engineering-2010. June 30-July 2, 2010, University of Nottingham, UK.
18) Tantisevi, K., and Akinci, B. (2007) Automated generation of workspace requirements of mobile crane operations to support conflict detection. Automation in Construction, 16 (3), pp.262-276.
19) Zhang C., Hammad A., Zayed T. M., Wainer G., and Pang H. (2007) Cell-based representation and analysis of spatial resources in construction simulation. Automation in Construction, 16, pp.436-448.
20) Kim, K. H., Kim, G. T., Kim, K. H., Lee, Y. S., and Kim, J. J. (2009) Real-time progress management system for steel structure construction. Journal of Asian Architecture and Building Engineering, 8(1), pp.111-118.