**FPSWizard**: A web-based CBR-RBR system for supporting the design of active fall protection systems

Yang Miang Goha, Brian H.W. Guo

### Keywords:
- Case-based reasoning
- Rule-based reasoning
- Fall from height
- Active fall protection system
- Construction safety

### Abstract
Fall from height is a perennial problem in the construction industry. Active fall protection system (AFPS) is frequently a must in situations where working conditions are difficult and other controls are not feasible or inadequate. However, the design and selection of AFPS are still problematic in the construction industry. This paper aims to develop an online knowledge-based system, FPSWizard, to support the design and selection of AFPS. The hybrid system adopts a combination of case-based reasoning (CBR) and rule-based reasoning (RBR) to improve retrieval performance. FPSWizard is meant to recommend suitable AFPS based on similar past design cases. Potential end users, such as professional engineers and safety professionals, can use the system as a decision support system when they are selecting and designing a solution to the work-at-height problem at hand. A total of fifty stored cases were created based on actual work scenarios and AFPS designs in the construction industry. A case structure was also created using the AFPS-Ontology. The system was assessed using a leave-one-out cross validation approach, where fifty cases in the case base were used to test the retrieval performance of the system. The hybrid CBR-RBR approach had an average positive predictive value (PPV) (or precision) of 90%. In comparison, a pure CBR approach had an average PPV of 76%. FPSWizard forms an important part of an intelligent system which provides comprehensive solutions to fall from height. This paper also made important strides towards intelligent safety engineering and management in the construction industry.

### 1. Introduction
Construction workers are at risk of falling from height across the lifecycle of a building, from construction to maintenance, renovation, and demolition [1]. Working at height is dangerous in nature. Falls from height are high risk occupational accidents in many countries, such as the U.S. [2], the U.K. [3], Australia [4], Singapore, and New Zealand [5]. For example, falls accounted for 35% of all workplace fatalities and 43% of major injuries in Singapore in 2015 and construction remains the top contributor of fall-related injuries [6]. These injuries caused huge suffering, loss, and economic costs. The Occupational Safety and Health Administration (OSHA) estimated that each fall result in claims between US$50,000 and US$106,000 [7], and this amount does not include indirect or intangible costs like work stoppages, morale issues, societal and personal costs.

In practice, preventing fall from height has always been one of the priorities of site safety management. Fall hazards are identified and communicated to workers through job safety analysis, task analysis, or safe work method statement. In order to protect workers working at height, control measures must be comprehensive and multifaceted. A combination of control measures is usually adopted, including elimination (e.g., prefabricating wall frames horizontally before standing them up), substitution (e.g., using mobile elevated work platform instead of ladders), engineering controls (e.g., guardrails), and administrative controls (e.g., work-at-height rules and procedures). Fall protection systems can be classified into two categories: passive and active fall protection systems. Passive fall protection system is “a means of providing fall protection that does not require workers to wear or otherwise use fall-protection equipment or to have any special knowledge or skills related to this system.” [8]. PFPS does not require any use of personal protective equipment (PPE) or active participation of workers. Typical examples include guardrail systems and safety nets. By contrast, active fall protection system (AFPS) is defined as “a means of providing fall protection that requires workers to take specific actions, including wearing (and otherwise using) personal fall-protection equipment and following prescribed procedures.” [8]. Common AFPS includes travel restraint and fall-arrest systems. A travel restraint system is used to prevent its users from reaching unprotected edge or
opening, while a fall arrest system is an assembly of components (e.g., full body harness, connectors, lanyard, energy absorber, and anchor) that will arrest a worker's fall. Fall arrest systems generally include horizontal lifeline systems (HLL) and vertical lifeline systems (VLL).

PPE is the least effective control measure in the hazard control hierarchy and often considered as a last resort. However, it is frequently a must in situations where working conditions are difficult and other controls are not applicable or inadequate. For example, installing a safety netting system may not be an applicable solution for old roof maintenance and repairing. When PFPS solutions and other controls are not applicable and inadequate, AFPS is the next best solution. In general, there are two problems with the use of AFPS: workers' misuse/non-use of AFPS and inappropriate design/selection of AFPS. The first problem is often explained and managed by investigating the effects of organizational, group, and human factors on workers' safety knowledge, motivation, and behavior [9,10]. By contrast, the second problem has received less attention. The design and selection of AFPS is a knowledge and experience intensive process [11]. It is highly dependent upon dynamic construction environments, tasks, location, and workers. It may be difficult for designers to select an appropriate type of AFPS because they work upstream of a construction project and thus face great risk of failing to anticipate job tasks, workers, and involved building elements. This is even more challenging for novice engineers who may not have adequate knowledge and competency to select and design a reliable solution to a given working at height problem. For example, AFPSs were often poorly designed by users (e.g., contractors) without proper endorsement from professional engineers (PEs) [12]. Even PEs often adopt wrong calculation methods with invalid assumptions when they design AFPS [13]. PEs may underestimate the maximum arrest force exerted on a worker when a fall arrest system stops a fall. Another common mistake is that inadequate "minimum clearance below the platform" is specified based on incorrect calculation approach. Consequences of these mistakes are that a worker who is arrested by the fall arrest system can be injured, or even killed, by the arrest force or an obstacle or the ground before the fall arrest system stops the fall. In practice, PEs must be knowledgeable in a wide range of fall protection systems and different ways to combine the equipment and systems in varying environmental and site conditions. However, human judgment is always subject to serious fallacies. Relying only on intuition and experience may lead to inappropriate AFPS designs and can cause serious consequences.

Considering the lack of competency in designing and selecting AFPS in the construction industry [12], a knowledge-based system can assist both PEs and contractors with decision making under uncertainty. With the advancement of artificial intelligence (AI), a number of established AI techniques such as case-based reasoning (CBR) and rule-based reasoning (RBR) can be used to provide reliable recommendations. Such a system can relieve cognitive burden of designers and save time. However, to the best of the authors' knowledge, there is rather limited application of these AI techniques to the design of AFPS. In order to fill the research gap and promote the use of AI in construction safety engineering and management, this paper aims to develop an online knowledge-based system, FPSWizard, to support the design of AFPS using a combination of CBR and RBR. FPSWizard is designed as a web-based decision support system which has the ability to identify solutions, i.e., the type of AFPS, to new work-at-height problems based on similar past design cases. Potential end users, including professional engineers, contractors and safety professionals, can use the system as an online assistant when they attempt to identify a solution to a work-at-height problem at hand.

2. Background

2.1. Fall from height in the construction industry

Fall from height has received significant attention from scholars. Early attention was mainly on identifying root causes and contributing factors of fall accidents. For example, Chi [14] developed a coding system to categorize fatal falls in terms of fall causes, fall location, individual factors, and company size. The coding system can be utilized to identify important contributing factors and control measures. Huang and Hinze [15] investigated the root causes of fall accidents based on the data from OSHA. A number of contributing factors were identified, including lack of safety training and human error. Similarly, Chan [16] identified 12 common contributing factors by analyzing 22 fatal industrial fall accidents in Hong Kong. Based on the findings, they proposed five strategies to reduce fall accidents, including (1) provide and maintain a safe system of work, (2) provide a suitable working platform, (3) provide safety information, training, instruction, and supervision, (4) provide suitable fall arresting system/anchorage, and (5) maintain safe workplace. On the other hand, Wong [17] adopted the Human Factor Analysis Classification System (HFACS) to identify and classify the root causes of fatal fall accidents. Within the area of human factors, Goh and Binte Sa’adon [10] adopted the theory of planned behavior to model the cognitive factors influencing the unsafe behavior of scaffolders.

Risk identification and assessment are important processes to control falls. In order to facilitate the process, Sa [18] compared the risk factors of falls between commercial and residential roofers. Results suggested that residential roofers are more likely to fall than commercial ones. Aneziris [19] developed sixty-four logical models to quantify fall risk. In addition, Nguyen [20] proposed a Bayesian network (BN) based approach to diagnose the accident risk of working at heights.

Information technologies have been used to help reduce fall accidents and injuries. For example, Navon and Kolton [21] developed an automated system that can identify dangerous work-at-height activities and areas. The schedule is integrated into the model which enables it to produce both textual and graphical reports that correspond to the schedule. The automated model was implemented in a prototype written in Visual Basic (VB), AutoCAD, and MS Project. More recently, Qi [22] developed a PTD (prevention through design) software tool to help designers implement best practices to prevent fall accidents. Using the PTD tool, automatic safety checking can be performed by using BIM technology and a knowledge base that was designed based on best practices. These efforts were aimed at reducing fall accidents by improving building design and optimizing production planning. Supported by these advancements, a part of fall hazards could be either eliminated, substituted, or managed by engineering and administrative controls. However, personal protection equipment, including AFPS, is still required as a last line of defense to protect workers in many situations [12]. Despite the importance, no research, to the best of the authors' knowledge, has been conducted using information technologies to support the design and selection of AFPS.

2.2. Design of AFPS

In order to support and facilitate the design of AFPS, standards were developed at the national level, such as ANSI/ASSE Z359.6:2009 by American National Standards Institute, American Society for Safety Engineers [23], Z259.16-15 by Canadian Standards Association [8], and SS607: 2015 by Singapore Standards Council [24]. These standards specify requirements for the design and performance of complete active fall-protection systems. Despite these standards and specifications, the design and selection of AFPS are still problematic in the construction industry. For example, Goh and Wang [12] pointed out that many AFPS are designed by PEs on an ad hoc basis and assembled by contractors who purchase the individual components. They evaluated eleven cases of HLL system design and concluded that all cases were inadequate and that many PEs were not familiar with these design standards. The lack of knowledge was reflected by the fact that PEs used wrong assumptions and calculation methods. For example, PEs tend to focus on static force rather than dynamic force on lifeline and users when a fall occurs [13].
Note that poor design of AFPS would provide a false sense of security, where workers wrongly assume that they are under protection. Two common injury scenarios caused by the use of ineffective AFPS are: (1) AFPS cannot stop the fall before workers hit the ground and (2) workers are injured by dynamic forces created during the fall arrest.

2.3. Previous applications of AI to construction safety

Artificial Intelligence (AI) is a broad discipline. John McCarthy, one of the founders of AI, first coined the term “AI” in the Dartmouth Conference which was devoted to the subject. He defined AI as “The science and engineering of making intelligent machines, especially intelligent computer programs.” [25]. In essence, AI aims to develop a machine or a computer that can think, learn, and behave in an intelligent manner. AI research covers a wide range of techniques and applications, including natural language processing, neural networks, machine learning, data mining, virtual reality, augmented reality, CBR, RBR, information retrieval, knowledge representation, intelligent robots, to name but a few. A detailed description of AI is beyond the scope of this paper. The main focus is placed on reviewing previous applications of AI to construction safety engineering and management.

The past two decades have witnessed a fast-growing number of applications of AI to construction safety engineering and management. Early attempts were made to integrate AI into virtual reality for intelligent safety training and education [26,27]. Computer vision technologies were utilized in safety management to detect and track objects and recognize action [28]. In addition, researchers used recent advancements in AI, machine learning and data mining, to predict injuries [29,30], identify safety incompatibilities [31], and predict safety behavior [10]. The main goal of these studies is to extract patterns and new knowledge from large amounts of data.

In addition, expert system methodologies (e.g., RBR and CBR) were used for different safety problems. For example, Goh and Chua [32,33] proposed a CBR approach to construction hazard identification that facilitates systematic feedback of past knowledge in the form of incident cases and hazard identification. Lu et al. [34] used a CBR approach for automated safety risk analysis on subway operation. Rule-based intelligence has been used to develop automated systems for construction safety management. For example, Boukamp [35] developed an automated system which supports job hazard analysis (JHA). A number of rules were designed to represent JHA knowledge. In addition, Zhang et al. [36] proposed an automated safety planning system for JHA. The Semantic Web Rule Language (SWRL) rules were developed to represent safety regulations and industry safety best practices. Similarly, Lu et al. [37] developed a model based on Protégé for automated construction safety checking, in which safety checking constraints were represented by SWRL rules. Knowledge is the core of most decision support systems. As such, efforts were made to engineer and represent safety knowledge by developing safety-related ontologies [11,36].

To summarize, the use of AI techniques is proliferating to a number of domains of construction engineering and management. These applications of AI techniques have facilitated smart and automated safety engineering and management processes in the construction industry. However, it is clear that before these applications are mapped and integrated into a comprehensive intelligent safety management system, more efforts are to be made to develop intelligent systems to discover and represent safety knowledge and solve problems in many other domains. For example, Zhang’s work [36] applies BIM technology and SWRL rules to visualize fall protection measures such as scaffold and guardrail systems and offer general recommendation about AFPS. However, it does not include any knowledge and rule bases on which the right type of AFPS can be derived.

2.4. Scope of the research

World Health Organization (WHO) defined a fall as “an event which results in a person coming to rest inadvertently on the ground or floor or other lower level.” [38]. Fall hazards are present at almost every workplace. To limit the scope, this study focused on falls from the construction industry during construction and maintenance phases. In addition, work at heights is classified as hazardous, i.e. when a person is liable to fall more than 3 m or more [39]. This study, as well as FPSWizard, focuses on hazardous work at heights which are more likely to require AFPS.

3. Design and development of FPSWizard

FPSWizard is an online knowledge-based system to support the design of AFPS. It supports AFPS designers by providing past similar cases as positive examples of how a particular type of AFPS was deployed in a similar scenario and at the same time make past calculations available.

3.1. Methodology

FPSWizard uses a combination of CBR and RBR to identify which type of AFPS should be used in a given work-at-height scenario and retrieve the most similar cases from a case library.

Case-based reasoning (CBR) is an approach to solving problems by learning from similar past experiences and cases. It originated from the work of Roger Schank [40,41], in which he suggested that when a person faces a similar situation, the past experience is re-collected and the person can follow the same steps to find a solution. CBR has often been used in problem-solving and learning [42]. Unlike other problem solving approaches in artificial intelligence (e.g., rule-based expert systems), CBR is memory based, stimulating human-thinking process and problem-solving strategies [43,44]. As a lazy problem-solving method, it has a number of benefits, such as ease of knowledge elicitation [45], incremental learning [46], ease of explanation [47], and ease of maintenance.

According to Aamodt and Plaza [48], the CBR cycle consists of four “REs”: retrieve, reuse, revision and retention, as shown in Fig. 1. The process of the four REs is as follows:

1. **Retrieve**: the system compares the query case with past cases and retrieves the most similar case(s) from the case library. Indexing schemes and similarity metrics are used for the retrieval process.
2. **Reuse**: the solution of the most similar retrieved cases is used as proposed solution to the query case.
3. **Revise**: The proposed solution is then assessed and revised, if necessary.

![Fig. 1. CBR cycle (based on Aamodt and Plaza [48]).](image-url)
(4) Retain: When the solution is adopted, the new problem-solving experience can be retained in database.

Rule-based reasoning (RBR) is another popular reasoning approach used in AI communities [49]. Unlike cases that encompass knowledge accumulated from specific situations, symbolic rules represent general knowledge of a specific domain [50]. The basic form of a rule follows IF-THEN structure. As such, rules are patterns and rule engine looks for these patterns in the data [51]. The first rule-based system was DEN-DRAL which was developed in the field of organic chemistry [52].

The combination of different problem-solving and knowledge representation approaches has received significant attention in the literature [50]. One of the most popular combination is a CBR-RBR integration. The rationale behind the combination is that symbolic rules and cases complement each other and therefore yield better performance [50]. Prentzas et al. [50] suggested that rules represent general knowledge of a domain whereas cases represent specific knowledge and that the integration is useful. Some disadvantages of RBR, such as brittleness of rules, difficulty in maintenance of large rule bases, and inference efficiency problems, can be compensated by CBR.

Four different approaches were used to combine CBR and RBR, including (1) RBR-first and CBR-last [53], (2) CBR-first and RBR-last [51], (3) CBR and RBR in parallel [54], and (4) RBR and CBR combined [55,56]. In practice, a CBR-RBR integration often serves the purpose of improving case retrieval performance. For example, Saraiva et al. [51] used a CBR-first and RBR-last approach to early diagnosis of gastrointestinal cancer. A set of rules were integrated into the CBR process to dynamically adjust the attribute weights. Evaluation results indicated that the accuracy of the diagnosis significantly increased when compared to a pure CBR approach. Similarly, Rossille et al. [53] adopted a RBR-first and CBR-last approach to improve the retrieval performance by comparing a patient’s case to the corresponding guideline (represented by rules), then to past cases. In the construction industry, Dzeng and Lee [56] developed a Schedule Coach system which finds potential schedule errors based on rules, and suggests corrections using CBR. A rules library, together with a cases library, was developed to improve suggestions for revisions of a project schedule.

This study adopts a CBR-first and RBR-last approach to retrieve similar cases for learning. In specific, CBR was used to retrieve an initial list of similar cases by computing similarity on both local (i.e., attributes) and global (i.e., cases) level. A set of rules were integrated into the CBR process to improve retrieval performance by dynamically adjust the attribute weights.

3.2. Workflow of FPSWizard

Fig. 2 illustrates the workflow of FPSWizard. A designer (e.g., a PE) uses the case query interface when s/he intends to identify an appropriate AFPS to a work-at-height problem at hand. The designer defines the problem by assigning values to a list of attributes defined by the AFPS-Ontology [11]. The ontology is partly utilized to define the query vocabulary and case structure. The information about the query case is processed by the search engine which consists of a combination of CBR and RBR. Similarity between the query case and stored case in the case library is calculated on both local (i.e., attributes) and global (i.e., cases) level. In order to improve retrieval performance, a set of rules are used to dynamically adjust the attribute weights. Based on the similarity, the most similar cases are retrieved from the case library. FPSWizard recommends the solution of the most similar case(s) to the work-at-height problem.

3.3. Case representation and case library

A case can be defined as a contextualized piece of knowledge representing a previous experience [57]. In order to develop a case library for FPSWizard from which similar cases can be retrieved, fifty-nine real work-at-height scenarios and PE-endorsed design cases were collected from the construction industry. Based on the real work-at-height scenarios, the research team designed solutions in accordance with the requirements of the design standard: SS607:2015 [24]. Original designs endorsed by PE were converted into formal cases with a detailed description of problem and solution. During the case library development, nine cases were removed because of inadequate information. Fifty cases were developed and compiled based on the following case representation structure (see Table 1):

Despite the comprehensiveness, potential industry users suggested that the cases were difficult to use as there was too much information; they prefer a more concise format. For this reason, the AFPS-Ontology [11] was used to define the structure of case reports. The ontology was adopted based on the following two considerations. Firstly, “deep attributes” or key attributes that influence the type of AFPS selected, have to be identified so as to improve the ability to retrieve suitable cases. Examples of deep attributes include relative position between anchorages and platform and direction of movement, which are captured in the AFPS-Ontology as it was designed to represent the domain knowledge of AFPS design. Secondly, CBR systems often perform specific tasks and generality is often compromised for system performance [43]. This issue can be reduced through the application of AFPS-Ontology, which is designed as a part of an overall ontology for construction hazard management, i.e. the case information can potentially be related to other construction hazard management information.

The final case library (L) is composed of fifty design cases. Being consistent with the AFPS-Ontology, each case was represented by three parts (Cp, Ci, Cc), where Cp is a problem description of a case C in the case library, Ci is the AFPS solution, and Cc is the context or design standard which the AFPS solution is based on (see Fig. 3). Because of the fact that all design cases were developed based on the same standard (i.e., SS607:2015), Cc is not able to distinguish between cases, Ci is not considered in the case retrieval process described herein.

In this study, a ‘problem’ is defined by ten attributes (a), namely task type, direction of movement, platform type, shape of platform, platform slope, leading edge of platform, possible anchorage, shape of anchorage, relative position between anchorage and platform, and IPC building element. A ‘solution’ is defined by the type of active fall protection system. There are four main types of AFPS, namely FAS-HLL (a horizontal lifeline (HLL) used as a fall arrest system (FAS)), FAS-(R)L, (a FAS with a lanyard (L) or a self-retracting lanyard (R)), FAS-VLL (a vertical lifeline (VLL) as a FAS), and travel restraint system (TRS). Values of all attributes used in the case retrieval process are presented in Table 2.

Fig. 4 illustrates the number of design cases in terms of AFPS type. Among all fifty cases, 18% use TRS, 40% use FAS-HLL, 20% FAS-VLL, and 22% FAS-(R)L.

3.4. Retrieval strategy and similarity measures

The purpose of similarity measures is to quantify the degree of resemblance between a query case and each of the past cases in the case library [58]. In this study, similarity between the query case and past cases is calculated on two levels: local level and the global level [32]. At the local level, similarity scores were computed for each attribute. At the global level, the similarity between cases was computed by combining the similarity scores of all attributes [42]. The value of similarity is between 0 (not similar) and 1 (most similar).

Local similarity functions measure the degree of similarity between attribute i of the stored case j and the query case. Hamming distance function was adopted to calculate similarity for symbolic values [58], such that
3.5. Rules of attribute weights

In conventional CBR systems, a static weighting approach was used to assign fixed attribute weights for all case attributes throughout the whole retrieval process. This approach is less effective in certain cases where the relevance of attributes changes in different scenarios. A close examination of design cases reveals that some cases are almost identical, where the cases have the same values for most attributes, but have different AFPS solutions. This problem would decrease the ability of a CBR system to select a suitable case for the query case [59]. In order to minimize this problem, rules were often integrated into the process to improve its effectiveness and efficiency by adjusting attribute selection, similarity measures [60], and/or attribute weights [51,61]. Designing rules to adjust attribute weights is an important part of attribute relevance learning, which is defined as the process of determining the attribute relevance embedded in similarity measures.

This study used a dynamic weighting approach, which allows the system to dynamically adjust the weight of attributes according to the attributes of each case. Six rules (see Table 3) were developed to improve case retrieval performance and increase the accuracy of recommendation. These rules were developed based on the information from AFPS design standards (e.g., Z259.16 and Z359.6), PE-endorsed AFPS design cases, and work-at-height best practices.

3.6. Case retrieval algorithm

The case retrieval algorithm (see Appendix A) works as follows: when a query case is input into the system, the search engine calculates similarity scores between the query case and all stored cases in the case library by applying the rules whenever necessary (Table 3). The most similar case(s) with the greatest value of similarity score is retrieved from the case library. The solution to the similar cases will be displayed as a recommended solution to the query case.

4. System implementation

The FPSWizard system implementation employs the MVC (Model-
(1) View module

A view handles the outputs. It handles the UI of the application. The View was developed using the bootstrap framework.

(2) Controller:

The controller component processes the inputs from a user, manipulates data using the Model component, selects views, and interacts with views to generate outputs.

**Table 2** Problem and solution attributes.

<table>
<thead>
<tr>
<th>Category</th>
<th>Attribute</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem attributes</td>
<td>a1: Task type</td>
<td>Symbolic</td>
<td>Attendance; construction; demolition; dismantle; disposal; installation; logistic; maintenance; move; operation; removal; renovation</td>
</tr>
<tr>
<td></td>
<td>a2: Direction of movement</td>
<td>Symbolic</td>
<td>Horizontal line; horizontal plane; inclined line; inclined plane; point; space; vertical line; vertical plane</td>
</tr>
<tr>
<td></td>
<td>a3: Platform type</td>
<td>Symbolic</td>
<td>Cable net; floor; formwork; gondola; ladder; roof; scaffold; slab; staircase; steel beam; steel structure; strut</td>
</tr>
<tr>
<td></td>
<td>a4: Shape of platform</td>
<td>Symbolic</td>
<td>Line; plane; point</td>
</tr>
<tr>
<td></td>
<td>a5: Platform slope</td>
<td>Symbolic</td>
<td>0 Degree; 1-5 degrees; 6-15 degrees; greater than 15 degrees</td>
</tr>
<tr>
<td></td>
<td>a6: Leading edge of platform</td>
<td>Symbolic</td>
<td>Fully open, three-side open; adjacent half open; opposite half open; one-side open; fully closed</td>
</tr>
<tr>
<td></td>
<td>a7: Possible anchorage</td>
<td>Symbolic</td>
<td>Barricade; beam; ceiling; catwalk; column; formwork; prop; purlin; rebar; roof; scaffold; strut; wall;</td>
</tr>
<tr>
<td></td>
<td>a8: Shape of anchorage</td>
<td>Symbolic</td>
<td>Line; plane</td>
</tr>
<tr>
<td></td>
<td>a9: Relative position between anchorage and platform</td>
<td>Symbolic</td>
<td>Anchorage above platform; anchorage adjacent to platform; anchorage below platform; anchorage on platform; anchorage separate from platform;</td>
</tr>
<tr>
<td></td>
<td>a10: IFC building element</td>
<td>Symbolic</td>
<td>HVAC; balcony; barricade; beam; ceiling; chimney; column; covering; curtain wall; door; floor; footing; formwork; pile; plate; railing; roof; scaffold; slab; staircase; steel structure; wall</td>
</tr>
<tr>
<td>Solution attribute</td>
<td>s1: Type of AFPS</td>
<td>Symbolic</td>
<td>FAS-HLL; FAS-(R)L; FAS-VLL; TRS</td>
</tr>
</tbody>
</table>

![Fig. 3. Case representation.](image3)

![Fig. 4. Number of cases in terms of type of AFPS.](image4)
The model component manages the behavior and data of the application domain, provides access to the case library, retrieval algorithm, and rule base. It corresponds to all the data related logic that the user works with. In FPSWizard, the model component is where all the calculation parameters are encapsulated into classes.

(4) Interactions among View-Controller-Model

The Controller sends commands from a user to the model to update the model’s state. For example, when a user types in the value of “Task type” in the interface, the controller sends the command to the model and updates the state of “Task type”. It can also send commands to its associated view to change the view (e.g., clicking the “download” button to download the retrieved cases). The model stores all required data, rules, algorithms that are retrieved according to commands from the controller. The view generates new outputs (e.g., a list of retrieved cases) to a user based on changes in the model. Based on the MVC design pattern, the users can gain access to the system on the Internet through browsers (e.g., IE and Chrome). Programming language Java was used to bridge the database server, web server, and GUI. Main development tools include Java JDK1.5, JavaScript 1.5, JSP2.0, ASP·NET, Tomcat5.5 and Oracle10 were developed in a Windows 2000 Server environment.

5. A case study

A case study was conducted as follows in order to demonstrate the retrieval process. Assume that a user has a work-at-height problem as follows:

“Two workers are hoisting glass panels near the edge of a building.
6. Evaluation

The purpose of the evaluation is to test the system's ability to retrieve a suitable case from the case library. The evaluation includes qualitative assessment of the suitability of the stored cases and quantitative analyses of the retrieval performance.

6.1. Evaluation of the case library

Considering the fact that retrieval performance is likely to be adversely affected if there are errors in the existing cases [59], the first step of evaluation is to ensure that all design cases have suitable solutions. This is an important prerequisite for the success of FPSWizard. Thus, six work-at-height experts were invited to participate in the evaluation. Details of the experts are presented in Table 4. An evaluation form and fifty design cases were sent to the experts via email. Experts were requested to evaluate whether the proposed solution in each design case is appropriate and effective. In addition, experts were allowed to provide further comments about the solutions. All six experts completed the evaluation and returned the evaluation form to the authors. Results indicated that experts deemed all solutions to be appropriate and effective.

6.2. Evaluation methodology and retrieval performance measures

This study adopted a leave-one-out cross validation approach [64], to evaluate the retrieval performance of FPSWizard. During the validation, each stored case is used as a test or query case and the similarity scores for the remaining 49 stored cases were calculated and the solution in the most similar case was used as the system's recommendation. This study conducted two experiments to examine the retrieval performance of FPSWizard. The first experiment was to investigate the effect of the number of attributes on retrieval performance for a pure CBR approach. This study developed a prototype based on MyCBR Workbench [65], which is an open-source similarity-based retrieval tool. MyCBR Workbench allows preliminary modeling of similarity measures and simulation of the retrieval process. Three models with various numbers of attributes were developed (Table 5). Model 1 includes three basic attributes, namely (a1) task type, (a3) platform type, and (a10) IFC building element. Model 2 computes similarity on five attributes, including (a1) task type, (a3) platform type, (a5) platform slope, (a7) possible anchorage, and (a10) IFC building element. Model 3 covers all ten attributes. As shown, three models used the same similarity measures and attribute weights in the case retrieval process. The second experiment was aimed at comparing the retrieval performance of Model 3 (a pure CBR approach) and a hybrid approach (i.e., CBR-RBR).

Table 5 Three retrieval models using a pure CBR approach.

<table>
<thead>
<tr>
<th>Model</th>
<th>Attributes</th>
<th>Similarity measure</th>
<th>Attribute weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Three attributes: a1, a3, a10</td>
<td>Eqs. (1) and (2)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Five attributes: a1, a3, a5, a7, a10</td>
<td>Eqs. (1) and (2)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Ten attributes: a1, a3, a5, a7, a10</td>
<td>Eqs. (1) and (2)</td>
<td>1</td>
</tr>
</tbody>
</table>

The workers need to move horizontally on along the edge. The columns in the area can possibly be used as anchorages. The columns are adjacent to the work platform.

Table 4 Profile of experts.

<table>
<thead>
<tr>
<th>Expert</th>
<th>Position</th>
<th>Experience</th>
<th>Number of cases evaluated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Professional engineer</td>
<td>10 years</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Safety professional</td>
<td>15 years</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Safety professional</td>
<td>15 years</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Safety professional</td>
<td>12 years</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>Fall protection products supplier</td>
<td>13 years</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Construction specialist</td>
<td>12 years</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 7. Retrieved cases.
are defined as

$$PPV = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}}$$  \hspace{1cm} (3)$$

$$TPR = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$  \hspace{1cm} (4)$$

where “true positive” refers to an event where the test makes an accurate prediction, i.e. the most similar stored case provides a suitable solution for the query case; “false positive” happens when the test makes a wrong prediction, i.e. the most similar stored case provides an inappropriate solution for the query case, and “false negative” is the event that a stored case with a suitable solution for the query case is not retrieved from the case library.

While both PPV and TPR are useful measures of retrieval performance in CBR, in the context of this study, PPV is more important than TPR. PPV is a fraction of retrieved cases that were deemed by the system to provide suitable solution for the query cases, while TPR is a fraction of all cases with suitable solution, including those not retrieved. For FPSWizard, avoiding provision of erroneous cases (high PPV) is more important than retrieving all the cases with suitable solution (high TPR). Unwanted stored cases with wrong solutions may adversely affect a user’s decision and therefore lead to ineffective or even unsafe AFPS. Since high PPV usually comes with low TPR, only PPV was selected to measure the retrieval performance. Thus, for each round of test (based on Models 1, 2, 3, and hybrid approach) has fifty individual test or query cases and 49 stored cases. PPV was computed for each individual test using Eq. (3). An average PPV was then calculated for each round of test.

### 6.3. Experiment results

Retrieval performance of three models and the CBR-RBR approach is illustrated in Fig. 8. As suggested, precision scores of Model 1 and Model 2 are low, with 54% and 72%, respectively. This is because these two models fail to capture attributes with discriminating power. When the number of attributes increased to 10 (i.e., Model 3), Results had a gain of 5% in precision, compared to Model 2. However, the precision of a pure CBR approach is still low, especially when the FPSWizard is meant to provide safety solutions. The low precision of Model 3 is a result of failing to recognize the relevance of some attributes in different situations. For example, in a situation where ‘platform type’ is ‘strut’ and ‘possible anchorage is ‘prop’, the attribute ‘relative position between anchorage and platform’ plays a more discriminating role to determine solutions. If ‘relative position between anchorage and platform’ is ‘anchorage above platform’, the solution is FAS-L, whereas if the value is ‘anchorage adjacent to platform’, the solution is FAS-HLL. The precision of Model 3 is a result of failing to recognize the relevance of some attributes in different situations. For example, in a situation where ‘platform type’ is ‘strut’ and ‘possible anchorage is ‘prop’, the attribute ‘relative position between anchorage and platform’ plays a more discriminating role to determine solutions.

The average PPVs, by type of AFPS, are shown in Fig. 9. It suggests that a pure CBR approach (Model 3) produces relatively low PPVs by the type of AFPS, with 86% for FAS-HLL, 80% for FAS-(R)L, 79% for TRS, and only 60% for FAS-VLL. The low PPVs were a result of the inability of a pure CBR approach to dynamically adjust the relevance of attributes in different scenarios. Therefore, the approach is unable to separate cases with different solutions based on attributes that are applied with the same weight. By using a combination of CBR and RBR, the system’s ability to retrieve correct cases was significantly improved. A hybrid approach has the highest PPV (100%) when it deals with travel restraint systems. With CBR-RBR, FPSWizard had a gain of 9%, 5%, and 20% for FAS-HLL, FAS-(R)L, and FAS-VLL, respectively. As a whole, the hybrid CBR-RBR approach had an average positive predictive value (PPV) (or precision) of 90% (averaged across the four types of AFPS). In comparison, a pure CBR approach had an average PPV of 76%.

### 7. Conclusions, limitations, and future work

This study developed a web-based CBR-RBR system, FPSWizard, to support the design of active fall protection system. The development of FPSWizard can be justified by the fact that problematic design of AFPS is still common in the construction industry and that such an expert system could ease cognitive burden of designers, as well as the conflicts between production pressure and safety as it is time-saving. This study developed a case library for the system which consists of fifty stored cases. The AFPS-Ontology was utilized to define case structure and query vocabulary. With FPSWizard, professional engineers, contractors and safety professionals can define a work-at-height problem using ten attributes, namely task type, direction of movement, platform type, shape of platform, platform slope, leading edge of platform, possible anchorage, relative position between anchorage and platform, and IFC building element. The search engine of FPSWizard conducts reasoning with rules and calculating similarity at both attribute and case level. By adopting a combination of CBR and RBR approach, FPSWizard simulates the knowledge and behavior of expert designers of AFPS. CBR is used to solve new problems by adapting previously successful solutions to similar problems, while RBR is aimed at improving retrieval performance by addressing changing relevance of attributes in different scenarios. Such a combination improves the ability of FPSWizard to retrieve the most similar case and recommend a correct solution to a query problem. This study conducted two experiments to test the retrieval performance by measuring positive predictive value (PPV) of a pure CBR approach (Model 1, 2, and 3) and a hybrid approach. Experiment results suggested that PPV increases as the number of attributes increases and that compared to using a pure CBR (Model 3), a hybrid approach has a gain of 14% in PPV.

FPSWizard is designed to be a web-based system, supplementing users’ AFPS knowledge and facilitates knowledge sharing among users in the domain of AFPS design. This is important considering current knowledge level in the domain is generally low. However, FPSWizard is still a prototype with a number of limitations. First and foremost, the
FPSWizard only provides AFPS-related solutions to working at height problems. It is important to note that the fundamental issues contributing to construction accidents are frequently organizational, managerial, and human factors, such as lack of management commitment, lack of motivation, and lack of safety awareness. More often than not, workers fall from height not because companies do not know how to manage the fall hazards, but because of apathy, ignorance and a lack of management commitment to safety. These fundamental issues are beyond the scope of FPSWizard, which is not developed to motivate organizations to manage fall hazards. Instead, it acts as a knowledge-based system which assists PEs, contractors, and safety professionals, with the design of AFPS. In addition, the case library contains only AFPS cases, FPSWizard should be used based on an assumption that an appropriate AFPS is needed for a specific working at height problem. As mentioned before, the assumption is not always valid as AFPS is not the only nor the best solution and other controls may be more effective to address the problem. Future efforts should be made to develop a comprehensive knowledge-based system which can provide reliable solutions to working at height ranging from elimination, isolation, engineering, administrative controls, to PPE.

Secondly, current case library is relatively small. Future work can be done to expand the case library. In doing so, the efficacy of FPSWizard can be further tested and the knowledge base can be updated. Because of the limitation, current cases are in a flat organization: cases are organized in a single table, with rows being cases, and columns being attributes. However, the retrieval would be expensive and inefficient as the case library gets large, as retrieval in a flat organization processes all cases in the library for each query. In future, other case organizations, such as structured, semi-structured, and attribute taxonomies organizations [42], can be adopted.

The third limitation is that current system focuses only on retrieval. Case retention is only performed manually by experts. Future effort can be made to expand FPSWizard by developing functions of ‘reuse’, ‘revise’, and ‘retain’.

8. Acknowledgement

The authors gratefully acknowledge the funding and support provided by the Workplace Safety and Health Institute, Singapore, and the Ministry of Manpower, Singapore, for this study (Grant number—MOMOSDETT12000007). Technical advice from Pallavi Singh and Sachin Parthasarathy is also appreciated.

Appendix A. Case retrieval algorithm

Input: A query case Q is described by n attributes, $\beta = \{\beta_1, \beta_2, ..., \beta_n\}$.
Output: Similarity score between a query case and all stored cases in the case library
Local variables: stored case C is described by a set of attributes as follows:
$\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$, where n is the number of case attributes,
Data type = {Symbolic} ,
Weight (W) is the weight factor of the attribute $\beta$ so that $W = \{w_1, w_2, ..., w_n\}$, default value of W is 1,
W is determined by rules (R) = $\{R_1, R_2, ..., R_m\}$, where m is the number of rules
Begin select the query case Q.
Begin match the rules $R_0\ do\ (0 \leq j \leq m)$
if attribute values match the conditions, then adopt the conclusions.
End
Begin For each $\alpha_j$ of C do (0 ≤ j ≤ n)
Begin
Compare attribute value of $\alpha$ and $\beta$ :
Case of (attribute-function = 'Equal') do
Compute Eqs. (1)
End
Calculate the similarity scores between the query case Q and all past cases by Eqs. (2)
End
Rank cases according to the similarity scores from highest to lowest.

References

Labor, Occupational Safety and Health Administration, Washington, DC. 2023.


