Abstract. The site equipment (SE) planning is a fundamental part of the construction preparation. A suitable site equipment supports the timely, cost-efficient and qualitative execution of the construction progress. The use of scientifically based planning tools can both speed up the process of construction site planning and lead to better results. This paper proposes a rule-based knowledge inference system to support SE planners in a semi- or fully automated manner using input data from building information models and working schedules. The knowledge based system is built using the ASL 2 licensed business rule management system Drools. Exemplary rules for four types of site equipment have been developed and implemented. Using sample projects, the feasibility of the proposed approach has been proven by showing that site equipment can be generated instantly.

1. Introduction

A reasonable and appropriate site equipment planning supports the timely, cost-effective and high-quality implementation of a construction project. The purpose of the site equipment (SE) is to ensure an orderly, productive and safe execution of all tasks necessary during the construction, reconstruction or demolition of a building or construction. Poorly designed or inappropriately executed site layouts can slow down the construction process, generate unnecessary costs and even constitute actual safety risks, making the process of the construction site layout planning (CSLP) an indispensable step of the execution planning. The first step to create reasonable site layout plans is to identify all needed SE, and to determine the necessary dimensions of each element of the SE. To generate a site layout plan, the SE has to be placed on the available areas. The necessary site equipment varies widely depending on the conditions of the specific construction project. Due to the large deviations of the circumstances and specific requirements in different construction projects, the site equipment has to be planned individually for each project. However, despite the large impact of the SE on the on-site overheads and productivity of the construction, the site planning process has been formalized very little. Usually, planners conduct the CSLP manually, without technological support. The dimensioning of the individual elements of the SE is mostly realized based on experience of the planners and rules of thumb, without qualitative or quantitative reviews. The results of the manual CSLP planning thus depend solely on the expert knowledge and practical experience of the executing planner.

A large set of information, which is traditionally acquired in very late planning stages, has to be considered during the CSLP process, and changes in the construction design and construction methods usually require the adjustment or re-planning of the SE. To reduce the planning efforts and prevent from repetitive re-planning phases, the CSLP is usually conducted only after decisions on the design and construction are final, depriving the possibility to include information about the necessary SE in the process. In this way, potentially expensive and inconvenient solutions might be condoned because the SE was not taken into account during the construction planning.

To be able to include aspects of the site layout in the planning considerations, a fast and easy way support planners in their decisions by partly or even completely automating the planning of the construction site. To support the planners during the generation of individual site facilities, knowledge based systems (KBS) form a very suitable basis. These systems are computer programs that formalize human knowledge in a strict logical and computable manner, allowing them to infer conclusions from given facts. They are used to assist humans in solving complex problems and tasks.
This paper presents an adaptable approach to efficiently automate the generation of construction site facilities, concentrating on the identification and dimensioning of individual site equipment. In the first part of the paper, an overview of the state of the art in construction site planning is given. In the second part of the paper, different types of knowledge based systems are presented. Furthermore, a rule-based KIS is defined as most applicable for the use case presented here. In the third part of the paper, an implementation of a rule-based KIS for automated construction site facility generation is introduced.

2. State of the Art

Up to now, the construction site equipment is generally planned by hand. However, there has been effort by several research groups to automate the planning process. In the next sections, a short overview over the traditionally site equipment planning and the state of the art in digitally aided site equipment planning is given.

2.1 Fundamentals of Site Equipment Planning

The site equipment on construction sites is used to prepare and conduct all individual construction processes in the best possible way in order to enable a fluent and continuous construction progress. The construction site equipment includes all producing and non-producing facilities required on site for the construction or renovation of a structure (Meyran, 1973). Depending on the state in which a construction project is to be carried out, different laws and regulations have to be applied. This work is primarily concerned with legislation in Germany. Construction-related regulations can be found, for example, in the workplace ordinance (BAuA, 2016) and various DIN standards. The site equipment can be classified in seven basic groups (see Schach and Otto (2011) and Albert and Schneider (2016)):

- Construction machinery (e.g. hoists and concrete pumps)
- Social and office facilities (e.g. office and sanitary containers)
- Storage areas (e.g. tool sheds, outside and inside storage)
- Traffic areas and transport routes (construction roads, entrances and exits)
- Media supply and disposal (e.g. power and water supply and waste disposal)
- Site security (e.g. fences, illumination, scaffolding)
- Excavation support

With ongoing progress, the requirements and conditions on the construction site change. The construction process is typically divided into several construction stages, where some facilities may not be required in each construction stage. Therefore, a dynamic construction site plan is required. At the beginning of each construction stage, items which are no longer needed are disassembled and replaced by other facilities (Kumar and Cheng, 2015).

During the planning process, all needed site equipment is identified, dimensioned, and finally placed on the construction site. In this paper, we concentrate on the identification and dimensioning of SE. Dimensioning—the determination of necessary and economic dimensions according to the specific conditions and requirements of a construction project—is especially crucial for producing, transporting and storing facilities. Under dimensioned elements can lead to a delay in the construction progress (e.g. non-sufficient storage areas). Individual work steps could become impractical (for example if a crane’s reach is too small). Over dimensioned elements increase the costs (for example if the crane is higher than needed) and the travel times (if storage areas are too large and need to be crossed frequently).
2.2 Related Work

While digital assistance is widely available in other construction phases, the CSLP is supported little. Especially the automatic generation and dimensioning of site facilities are widely neglected. While various research groups have dealt with the optimization of construction site facility planning during the last decades, they generally use predefined site equipment and consider solely the positioning on site. Up to now, there is no comprehensive implementation of the construction site setup problem, requiring extensive manual inputs for all approaches.

First approaches on using knowledge based systems for construction related tasks have been published by Hendrickson et al. (1987) and Moselhi and Nicholas (1990). Both groups are using expert systems to create working schedules. Hamiani (1987) presented CONSITE, a knowledge-based expert system framework to solve the construction site layout problem. Tommelein et al. (1992) used an expert system to designing construction site layout. In both approaches, construction site equipment objects are represented by rectangular surfaces and arranged on a 2D building site. Limits of the program lie in the need of high amounts of manual inputs, as well as the lack of transparency and the possibility to influence the generation.

Newer approaches use the advantages of using information obtained from Building Information Models, but again, are neglecting the automated generation of the site equipment. Researchers concentrate on heuristic optimization of the site layout. Huang and Wong (2015) use a binary-mixed-integer-linear algorithm to optimize the site layout towards reduced travel times as well as little set-up costs. Shawki et al. (2010), Elgendi et al. (2014), and Kumar and Cheng (2015) propose the use of genetic algorithms to create construction site plans. The algorithms begin with random layouts, which are varied by crossing and mutation. The research also includes the use of meta-heuristic algorithms, such as swarm intelligence: Yahya and Saka (2014) use an algorithm based on the behavior within a beekeeper (artificial bee colony algorithm), Ning et al. (2010) use an ant-algorithm (max-min ant system). Wang et al. (2015) plan optimal crane positions on large-scale sites using firearm algorithm. Schwabe et al. (2016) use interactive rule checking to evaluate site layouts during the manual generation. An extensive literature review on prior approaches to optimize site layouts is also provided by Huang and Wong (2015).

3. Knowledge Based Systems

Knowledge based systems are computer programs using methods from the field of artificial intelligence. They are used to assist humans in solving complex problems and tasks, that are usually conducted by specialized decision makers. To that end, the algorithms mirror human thought processes and attempt to draw intelligent conclusions and action recommendations from given information (Lunze, 2016).

3.1 Characteristics and Architecture of Knowledge Based Systems

Knowledge based systems are characterized by the strict separation of knowledge (stored in a knowledge base) and techniques to retrieve information from that knowledge (called inference engine). Further components are needed to fill the knowledge base (expert interface and knowledge acquisition component) as well as to retrieve solutions for a specific problem (user interface, working memory and explanation facility) (Beierle and Kern-Isberner, 2008). The typical structure of a KBS is shown in Figure 1.

The core functionality of a knowledge based systems lies in the knowledge base and the inference engine. The knowledge base contains the permanent knowledge, which can be structured in rules and facts. While the knowledge usually consists of factual and certain knowledge,
heuristic knowledge can be added. However, heuristic knowledge is less rigorous and may lead to unexpected results, and is therefore rarely used (Shen et al., 2010). The inference engine is the knowledge processing and reasoning component of the KBS. In analogy to the functionality of the human brain, it is able to generate answers, predictions or suggestions for a specific problem by the use of artificial intelligence. There are two types of inference that can be implemented in inference engines: forward chaining and backward chaining (Lunze, 2016). Forward chaining is the classical data-driven approach: the input data is known and a previously unknown solution is searched. Backward chaining is used, when the input data is not known, i.e. it generates possible input data to obtain a desired solution.

For the creation and expansion of the knowledge base with expert knowledge, a knowledge acquisition component and an expert interface are needed. The expert interface operates as input device for entering knowledge about the specific field of the KBS. The knowledge acquisition component inserts the data into the knowledge base. Knowledge can be entered and altered by human experts as well as by the KBS itself (Raza, 2009).

To generate and retrieve solutions for specific problems, a user interface, and an explanation facility are used. The user interface provides the communication between the user and the KBS. It may contain a graphical interface and different amenities to facilitate the interaction. The specific problems have to be converted to facts that can be processed by the inference engine. Temporary, case-specific data is stored in the working memory. This data includes both data inserted into the user interface as well as the determined solutions for this data. The explanation facility is used to control the generated solutions. An explanation of how the inference is drawn is substantial to better understand how the system generates a certain solution, and to supervise the results in case of unexpected outcomes (Beierle and Kern-Isberner, 2008).

3.2 Types of Knowledge Inference Systems

During the last decades, several different methods of setting up knowledge inference systems have been developed. In the following, a selection of these methods are briefly described. Finally, a well-suited method for the application in automated construction site layout planning is identified.

Logic Based Systems (LBS): In logic based systems, the knowledge is formulated as assertions in logic. They are well suited for problem domains that can be divided in small, logically solvable entities. LBS are usually represented using logic-based languages, such as Prolog. A disadvantage of using hard-coded LBS lies in the low flexibility and in the need to recalculate the whole program for each fact (Lunze, 2016).
Rule Based Systems (RBS): Rule based knowledge systems are related to logic based system. They are well suited for problem domains that can be represented in modular rules. Rules are used to precisely describe elementary circumstances. They are typically in the general form of WHEN <situation or condition> THEN <action> (Kostem, 1986), contrast to the first predicate logic of other logic based systems. More complex systems can be reduced to a set of elementary entities, with each entity being represented by one rule. To process the set of rules through inference, it is insufficient to merely execute one rule after the other. Several rules may refer to the same fact, which might lead to a loop – one rule changing the fact, and the other rule changing it back. This is why a rule engine is used. Rule engines are able to execute a set of rules in an intelligent manner by using algorithms such as Rete (Forgy, 1982). They are able to support additional features, such as priorities, preconditions, or mutual exclusion. Therefore, even complex, cross-linked sets of rules can be evaluated efficiently. An advantage of RBS is the human readable and comprehensible formulation of the rules, allowing for easy inspection and rule acquisition. However, more complex relations between pieces of knowledge may be impossible to represent accurately. Also, reasoning under uncertainty is a challenge to be considered.

Case Based Reasoning (CBR): In case based reasoning, the knowledge base is structured as a set of cases, where previously encountered experiences are stored. New problems are solved by adapting the solution of the most relevant known case. Subsequently, the problem and adapted solution are saved, gradually increasing the knowledge base (“learning by doing”). Case based reasoning is therefore not logic based, but rather a memory based process. CBR are well suited for problems with complex cases, that means large scale patterns that cannot be divided into smaller, logically solvable entities. Difficulties in CBR lie in the knowledge acquisition and in the adaptation of known solutions to new cases. Decisions based on imitation may lead to devastating consequences through incorrect adaption.

Network Based Reasoning: In network based reasoning, the knowledge is organized in decision trees or networks, where several attributes are used to describe one object. The knowledge trees or networks can be generated from known cases through machine learning techniques. However, for each case, the values of each attribute should be known. Each attribute is assigned to one node of the tree, whereby each node represents the query for the value of the attribute. The edges originating from a node represent the possible values of the attribute. Each end node (leaf node) is assigned a truth value or class representing the outcome of the decision tree for this end node. Network based systems are well suited and primarily used for classification problems (Beierle and Kern-Isberner, 2008).

Regulations and instructions concerning site equipment are usually formulated in clear and precise rules that can be broken down to elementary entities. Usually, the correlations between requirements and solutions are concise and definite (e.g. the number of washing facilities per worker or minimal required crane capacities). Therefore, a rule based knowledge system is a good choice for this application. The human-readability and comprehensibility of the rules helps to lower the acceptance threshold of engineers. Additionally, it facilitates the knowledge acquisition and control of results. In the past, rule based systems faced difficulties in user-friendliness and applicability. Limited resources in hard disk space and system memory restricted the scope and inference speed, impeding the broad implementation of rule based knowledge systems. With modern hardware and improved inference algorithms, new opportunities arise for user-friendly and comprehensive applications.

3.3 Rule Based Knowledge Inference Systems in Drools

For this work, the ASL 2 licensed business rule management system Drools is used. It contains an inference engine used to process data on specific projects (facts) depending on the knowledge
base (rules). As most rule based systems, Drools uses an advanced version of the Rete algorithm (Forgy, 1982), which is able to scale to a large number of rules and facts by increasing the efficiency of the rule processing. The Rete algorithm generates a discrimination network using all conditions given in the different rules. The network contains different types of nodes: root, object, 1-input, 2-input and terminal. All objects enter through the root node. Each condition is represented by a 1-input node. If a rule contains more than one condition, a 2-input node is used to combine each two conditions in one 1-input node. The output of the last input node is used as input for a terminal node. The terminal node represents the action of a condition. During the first propagation of the fact, all conditions are checked and actions are performed when needed. The output of each node is stored. From the second propagation onward, the stored outputs are reused when the facts remain the same. Only conditions for altered facts must be checked, reducing the calculation effort significantly. However, as the interim results have to be stored, the memory usage can increase drastically (Forgy, 1982).

To better represent object oriented data and to reduce the calculation effort further, the Rete algorithm was adapted. Adding further node types, the developers of Drools implemented the ReteOO (Rete-Object-Oriented) algorithm (Sottara et al., 2010). The new node types are entry point nodes, object type nodes, alpha nodes, join nodes, and left input adapter nodes. Entry point nodes are located behind the root node. If several entry point nodes exist, the network can be split into several networks. For each object type used in the rules, an object type node is created. Object type nodes act as barrier—they only propagate facts that apply to the following nodes. This way, facts are only checked against rules whose conditions demand the same object type. Alpha nodes expand the function of 1-input nodes to evaluate literal conditions. Left input adapter nodes are used to convert objects to tuples for following joins. Join nodes, or beta nodes, expand the function of 2-input nodes to find matches for rules with several conditions. Additionally, ReteOO adds enhancements such as node-sharing, where nodes can be shared when rules follow similar patterns.

After further development, the PHREAK algorithm was introduced (JBoss Community, 2016). PHREAK is characterized as lazy and goal oriented, whereas Rete is characterized to be eager and data oriented. Instead of instantly firing all rules, rules with all inputs satisfied are queued. Rules are then fired depending on their salience. According to JBoss Community (2016), the PHREAK algorithm is additionally enhanced by adding three layers of contextual memory, stack based evaluations, isolated rule evaluation as well as rule, segment and node base linking.

4. Proof of Concept

To demonstrate and test the proposed approach, a prototype has been implemented using JavaFX. The prototype provides different features to support engineers during the planning of the construction process: generation of work schedules, generation of site equipment and generation of site layout plans.

4.1 Development of the knowledge base

The knowledge base consists of a set of rules defining the required specifications of the site equipment. In the business rule management system Drools, rules can be formulated using both the MVFLEX Expression Language, which allows simplified syntax, as well as in plain Java when additional commands are needed. The rules follow the typical form of \textit{WHEN} ... \textit{THEN} .... In this implementation, Java statements have been used in the \textit{THEN} statements.

During knowledge acquisition, rules have to be identified and transformed in the rule format. Rules on how to create site equipment are available in technical documents such as standards,
guidelines and drawings. Required dimensions and performance of the site equipment can often be derived from the demands of the construction and logistical considerations. Additionally, there are non-written rules and engineering knowledge arising from experience of the planners. Up to now, rules have been implemented for four main types of site equipment: tower cranes, concrete pumps, storage areas and social facilities. For better maintainability, an individual rule file has been created for each type of equipment. The characteristics implemented per site equipment type are depicted in Table 1. Several rules need to be implemented per characteristic, as they can be influenced by several conditions. To form the knowledge base, the rules have been formulated using a development environment. The knowledge base can easily be expanded with further information on the existing site equipment or with rules for the planning of further site equipment types.

<table>
<thead>
<tr>
<th>Site equipment type</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tower crane</td>
<td>height, jib length, capacity</td>
</tr>
<tr>
<td>Concrete pump</td>
<td>capacity, reach</td>
</tr>
<tr>
<td>Storage spaces</td>
<td>minimum width, minimum length or minimum area</td>
</tr>
<tr>
<td>Sanitary and office facilities</td>
<td>number of offices, break rooms, first aid rooms, restrooms</td>
</tr>
</tbody>
</table>

An example rule for creating the needed size of a storage space is depicted in Figure 2. In a first step, a storage space is created for each material. It can be read as: when material of type rebar is needed, and a storage space of type rebar is planned, then calculate the minimum side lengths. For rebar, the minimum storage side length can be assumed as the length of the longest bar. The minimum side width can be depicted by the number of bundles, where a bundle is assumed to have the width of one meter. To get the required width in meter, the number of bars is divided by the number of bars per bundle. The number of bars per bundle is requested as user input and stored in the HashMap map.

```rule
"Storage size rebar"
when
  $s : StorageSpace(type == "Rebar")
  $m : Material(type == $s.type)
then
double number = $m.getQuantity();
double minWidth = number / (double)map.get("amountInBundle");
double minLength = (double)$m.properties.get("Length");
modify($s){setMinWidth(minWidth), setMinLength(minLength)}
end
```

Fig. 2: Rule for rebar storage size calculation

4.2 Prototype Implementation

A process scheme for the proposed concept is depicted in Figure 3. As input for the site equipment generation, information from the BIM model and the working schedule is used (see Table 2). The working schedule provides additional information on the operation sequences of the individual construction processes. If certain site equipment elements are required only for a short time, the demolition and reconstruction of the site equipment may be considered. Additionally, the
knowledge of the time operation is necessary for the determination of reserve quantities and the reasonable design of storage areas. For the generation of working schedules, the tool “ultimatum” has been integrated (Abdelaleim et al., 2014). Each construction component from the BIM-model is linked with a process pattern to retrieve all processes necessary to conduct the construction (cf. Marx and König (2011)). The processes are then sorted by an optimization algorithm to form a working schedule. Additionally, each process has a list of needed resources, such as tower cranes, workers or concrete pumps. The working schedule, the correlating building components and the needed equipment (according to the processes) are stored in an SQLite database for data exchange. Using this input, the site equipment is generated using the knowledge based system presented in this paper. Additional information needed for the generation can be entered in an input prompt, for example if the workers are accommodated on site. The generation of site layout plans by placing the site equipment on the construction field is currently conducted by the user via keyboard input or drag and drop. An optimization algorithm to automate the positioning of site equipment is currently under development. After finishing the site layout planning, the generated site equipment is stored in a database (type, characteristics and location of each equipment) and used to simulate the construction again, this time considering the site equipment. For example, process times could speed up when two tower cranes are placed instead of one. If the working schedule has to be updated, the site equipment might change as well. If no changes are necessary, the process is terminated. In a next step, the generated site equipment could be integrated and visualized together with the BIM model using a VR system (Ebner et al., 2012). In this system, a 3D representation can be created for each site element using the generated dimensions.

![Fig. 3: Process scheme for a semi-automated site equipment planning system](image-url)

**Table 2: Information to be extracted from the BIM model**

<table>
<thead>
<tr>
<th>Category</th>
<th>Used Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions on site</td>
<td>property lines, approach and exit roads</td>
</tr>
<tr>
<td>Planned construction</td>
<td>floor area, coordinates of the highest point, building components</td>
</tr>
<tr>
<td>Material for prefabricated construction elements</td>
<td>number, weight and format of the parts</td>
</tr>
<tr>
<td>Material for in-situ concrete constructions</td>
<td>number and format of formwork, rebar, concrete volumes</td>
</tr>
</tbody>
</table>
The site layout planning is conducted in 2D using a graphical user interface (Figure 4). The main steps scheduling, dimensioning and positioning (yet to be implemented) can be accessed the buttons on the upper right corner. On the right hand site, a 2D representation of the construction field (light gray) with the planned construction (gray) is depicted. The site equipment is represented by their footprint. For a better overview, the positions of the elements can be shown as text. For construction machinery (tower cranes and concrete pumps), the reach is depicted, additionally. On the left hand side, all generated site elements are listed. If one element is selected in the list or on the 2D map, further information is depicted in the lower left corner. All site equipment can be altered towards special user requirements, and additional equipment can be easily generated by hand by using the buttons on the lower left corner.

![Fig. 4: Graphical user interface with plan presentation and interaction features](image)

5. Conclusions

In this paper, a knowledge based system to support engineers during site equipment planning is presented after evaluating different types of KBS. With new technologies, that allow for larger databases and faster rule execution, KBS offer new opportunities. The KBS has been implemented in Java, using the business rule management system Drools. Dimensioning rules have been implemented for several site equipment types, allowing the automatic generation of site equipment for given working schedules and BIM models. However, the prototype is still open for extensions: further types of site equipment, such as supply facilities or traffic areas, can be added. Up to now, only rectangular storage areas are generated. If fuzzy input and uncertainty should be considered, another system might be necessary. However, Drools already announced to implement functionalities on fuzzy reasoning in further versions, so an upgrade could be sufficient (JBOSS Community, 2016). Another idea currently investigated is the addition of a machine learning component, that is able to draw conclusion from a large amount of existing construction projects and site plans. This way, the generation is not reduced to rules, but also relies on experience. If needed, the site could be visualized in 3D using an external program. Additionally, the prototype is to be expanded by a site facility placing algorithm. The prototype will be tested and validated on several real-world pilot projects in the future.
Bibliography


