Case-based Reasoning for Construction Tasks

Abstract: An effective, transparent, and failure-free execution of construction projects depends on many conditions and can be controlled to a high degree by the responsible project manager. To support the manager’s activity and speed up the planning process, previous successfully executed projects can be reused in new situations with similar characteristics. Case-based Reasoning is used within this paper to reuse scheduling experiences from former projects. Cases are modeled by the generic language Feature Logic and defined following the Industry Foundation Classes in content. Subsequently, storing cases in a case base as well as reusing stored solutions becomes feasible. Evaluation methods are provided to retrieve cases from case base that are similar to a current problem. Their execution sequence is presented to the project manager as a possible solution and can be adapted for new conditions.

Key words: construction scheduling; case-based reasoning, Feature Logic, construction task similarity

Introduction

Scheduling in the construction domain is mostly a manual process that is very tedious and time-consuming due to frequent changes in project objectives and construction conditions at construction sites. Working schedules are not updated sufficiently during the execution, which often causes impediments in the execution order. More efficient tools to support the project manager are required to eliminate these drawbacks and to facilitate the planning process. An approach for the automated generation of schedules has been introduced in recently published work.[1] Case-based Reasoning (CBR) can be used to learn from successfully executed projects. This paper introduces a concept for modeling and storing construction projects, including the correspondent execution schedules, as cases. Furthermore, the determination of similar construction cases is presented in order to model construction processes for new building projects. Based on the CBR approach an appropriate case-base structure was specified following the Industry Foundation Classes (IFC) to support the storing of projects. This concept is lab-tested and validated within a case study for constructing with pre-cast components.

1 CBR Concept

Assuming that similar problems have similar solutions,[2] reasoning from previously executed projects can be applied to current projects. CBR is a well-known approach to reduce the planning effort in recurring situations. A CBR system usually contains a case base and reasoning methods to reuse stored cases in similar situations.[3] Cases consist of a problem and a solution part. To solve a new problem the following cyclic process is carried out: first, cases with the most similar problems are retrieved from the case base. The solutions of the retrieved cases can be adapted for new problem specifications. Afterwards, the problem with its confirmed solution can be retained in the case base as a new case.[3] The case base size increases with each retained case and therefore offers more precise assistance for further evaluations.

Over the past two decades several CBR systems have been developed in the construction domain, dealing with cost estimation,[4] contractor prequalification,[5] and building[6-8] and architectural[9-10] design processes. Some CBR research deals with case-based scheduling.[11]
2 Case Representation

In order to reuse experiences from former projects, they have to be described consistently and in general manner. According to our approach a case represents a construction objective within the execution of a building project, and it includes the accompanying execution solution. The construction objective specifies the case problem part and describes the building elements in two states: an existent as-is state and a final target state, which is achieved through the execution of the process. One construction task or a set of construction tasks have to be executed, in order to transfer building elements from their as-is states into the requested target states. This sequence of execution tasks defines the case solution part. Building elements that are in an as-is state are called task prerequisites. Analogically, task results are presented as building elements in a target state.

In Fig. 1, the construction task building up a column needs at least the constraint footing is finished. The result of the task building up a column leads to the target state column is built up. Additionally, this constraint represents one of the required prerequisites for the following beam erection task. If the equality of task constraints can be defined, tasks can be brought to an execution order, i.e., an appropriate execution schedule can be generated.[1] A case is sufficiently defined if the case problem part (task prerequisites and results) is specified and the case solution part (task execution sequence) can be generated. However, a task with its constraints represents the simplest case form.

Construction case problems often describe a group of building elements that have to be constructed, transported, composed, decomposed, or modified. For example, several columns in conjunction with a beam can form a frame or a certain amount of bricks can be joined to bring up a wall. Subsequently, it is eminently important to find appropriate instruments for describing construction objectives. In our research the generic language Feature Logic[12] is used to represent the as-is and target states of building components as object structures, which are called feature graphs. Herein, elements can be defined by means of their properties, or in other words by their features. Detailed information about the application of Feature Logic in this context is given in previous work.[1][13] Subsequently a uniform definition base is required to specify building components in content and to enable consistent case-based reasoning processes. Industry Foundation Classes (IFC)[14] provide a wide range of application possibilities and are well suited to the construction domain. Currently, information about the element type, geometry, material, and element-type specification are captured to define the case problem part. Further information about the implementation of IFC into the introduced approach is shown in.[13]

In Fig. 2, feature graphs are used to model the as-is state of the construction process building up a column. Herein, building elements are assigned either to a repository if delivered and available at the construction site or to a structure if they are already constructed. Two building elements (footing and column) are needed for the execution process. They are defined in as-is and target states. The footing is an element of the structure in the as-is state and also remains built-in in the target state. The column changes its position during the execution, and is added to the structure in the target state and removed from the repository.
Within feature graphs, different detail levels are considered. For example, the elements footing and column are located at the second level, whereas the elements boundingBox and reinforced concrete are located at the third level. This information is used to implement different feature importance while calculating structural similarity measures.

3 Case retrieval and concepts for adaptation

The reuse of executed project solutions depends on the ability to retrieve stored cases that are similar to a current execution problem. The case problem part helps to find similar cases within the case base, whereas the case solution part is used to adapt the retrieved solution for the new problem situation (e.g., in the form of an executed schedule). Consequently, for each new execution problem, stored case problem parts are compared to the current problem to estimate the similarity measures between them. A certain amount of cases having the highest similarity measures is then selected to solve the new execution problem. Within our approach the case problem comparison corresponds to the calculation of the matching degree between the feature graphs of the compared cases. Two kinds of similarity measures are evaluated. The structural similarity measure is determined by comparing the atom values of equal features (cf. Fig. 2) between two problems according to the Hamming distance calculation.\[16\] Afterwards, an overall similarity measure is calculated for each retrieved case by combining the structural and content similarity measures. Cases with the best similarity measures are selected in the retrieve phase. Consequently, the most similar solution is adapted to the current problem. Further explanations concerning the similarity evaluation procedure are given in previous associated publications.\[1\],\[13\] The suggested similarity determination approach allows evaluation on different levels of detail due to the hierarchical structures of the feature graphs. The importance of each element in the feature graph that counts within the evaluation process is a remaining focus of interest. The weighting of the appropriate elements and their contents is currently decided by the project manager according to a certain construction problem as well as his or her subjective preferences and experiences.

After the retrieve phase the execution sequence tasks of the retrieved cases are considered in detail. Subsequently, the execution sequence of the most similar case is presented as a constraint-task sequence. This sequence can be used to generate a schedule. However, in most cases the retrieved schedule does not completely satisfy the current constraints in the new problem situation. Therefore, the project manager will have to identify all non-matching tasks in the presented schedule and remove them. Afterwards, the resulting
gaps have to be closed by the appropriate tasks or task sequences.

4 Pre-Cast Components Study

Within this section, the functionality of the suggested approach is shown, and the similarity evaluation process is demonstrated on an example for constructing an industrial building with pre-cast components. For this purpose, the desired project objectives are specified first. Next, the case base is scanned to find cases with similar problems. Finally, the similarity measures of all appropriate construction projects are calculated.

The industrial building consists of several simple and two-bay frames, which consist of footings, columns, and beams. Hence, the construction of the industrial building can be split into the erection of two different frame constructions as well as erection of beams lying in between the frames. Each of these project objectives is expressed by its construction components and is modeled by Feature Logic following the IFC in content. In the as-is state, all building elements are delivered on-site and modeled by references to the element repository while the element structure has no references.

![Feature Graph](image)

Fig. 4 Feature graph of a stored and a new construction problem
All elements change their position during the execution into the element structure, to achieve the desired objectives, which are: constructing a simple frame, constructing a two-bay frame, and constructing a beam. In the next step, corresponding cases with appropriate objectives are retrieved from the case base and the similarity measures are calculated. Due to the shortage of space, only one part of the industry building, namely the simple frame erection, is considered in this example (see Fig. 4). All hatched elements exist on the displayed positions, belonging to the repository in the as-is state. In the target state, these elements are built in and hence removed from the repository. Gray shaded elements do not belong to the structure until the target state is reached. Certain deviations exist between the stored and the new simple frame due to structural and content differences of their feature graphs. The most conspicuous distinction results from the footing type (strip footing) in the stored problem. This fact leads to a missing footing element within the feature graph and hence to the mentioned structural divergence. Furthermore, the dimension values between the two problems differ for several elements and cause lower content similarity measures. The combined overall similarity measure is calculated with a homogenous weighting ratio (50:50 for structural and content similarity) and results in 78%. If no specific restrictions have to be considered or a first overview is requested, this kind of weighting can be used. Often, however, the number as well as the arrangement of the building elements is fundamental and leads to higher weighting for the structural similarity. Otherwise, some construction methods strongly depend on building element content specification and require higher content similarity weights. However, the calculated similarity measures for the mentioned project objectives of the industry building are summarized in Table 1. Furthermore, the main reason of divergence for each objective is noticed.

**Table 1 Overall similarity measure calculation**

<table>
<thead>
<tr>
<th>Project objective</th>
<th>Overall similarity measure</th>
<th>Main divergence reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple frame</td>
<td>78 %</td>
<td>Strip footing</td>
</tr>
<tr>
<td>Two-bay frame</td>
<td>86 %</td>
<td>Bucket footing</td>
</tr>
<tr>
<td>Beam erection</td>
<td>93 %</td>
<td>Element number</td>
</tr>
</tbody>
</table>

Despite the low calculated similarity measures, the execution processes that describe single project objectives are very similar. Fig. 5 shows the execution sequence for building up the simple frame as it is in the stored case. This execution sequence is accepted as a suggested solution for the new problem and presented to the project manager. The constraint strip footing finished does not correspond to the requirement in the new problem, which is pad footing. After removing the differing constraint from the execution sequence, all tasks affected by this constraint have to be adapted. For this purpose, a database with stored tasks is browsed for tasks which contain the constraint pad footing finished. Depending on further constraints in the new problem, stored tasks are more or less suitable to close the gap after removing the differing constraint. One possible solution is presented in Fig. 6. The suggested task-constraint pairs from the database result in an execution sequence with a surplus task building up a footing. For this reason, one single footing is built in for each column, corresponding to the construction of the pad footing. The described procedure of the constraint adaptation is repeated for each project objective. Subsequently, all execution sequences (those of simple frame, two-bay frame, and beam construction) have to be joined to achieve the working schedule for the case

---

**STORED PROBLEM**

![Fig. 5 Stored case “simple frame erection”](image-url)
of constructing an industry building. In most cases one or more tasks will be surplus or missing. The appropriate modifications have to be implemented by the project manager. In this manner, suitable support can be offered in situations where new decisions are required.

**NEW PROBLEM WITH COMPLETED TASK**

![Diagram showing execution sequence of building up a column](image)

Fig. 6 Execution sequence of “building up a column”

## 5 Conclusions and Future Work

Construction scheduling is a complex process that contains many decisions in order to find an effective execution solution. To support the project manager and to achieve more effective and transparent planning, experiences from previously executed projects can be reused in new situations using CBR. In this paper, an approach for representing construction projects as cases by Feature Logic and according to the IFC has been introduced. Furthermore, similarity evaluations have been presented to compare stored cases with new problems and, hence, to enable reasoning for further projects. Several suggestions for the adaptation of stored solutions have been presented in order to achieve an execution sequence for the new problem.

The suggested approach has many advantages due to its potential for variable application. Case retrieval can be provided with different precision specifications. Further fields of interest include extended specifications within the similarity evaluation process. The case base maintenance, as well as the implementation of adaptation methods within the reasoning process, will follow next. Further work will specify the procedure for combining single execution sequences for complex construction tasks.

### References


