CONSISTENT INFORMATION MANAGEMENT FOR STRUCTURING CONSTRUCTION ACTIVITIES

W. Huhnt, F. Enge

Department of Process Modelling in Civil Engineering, Technische Universität Berlin, Germany

ABSTRACT: Follow up charts require a list of activities, inter-dependencies between activities, and the duration of each activity as its input. This is also true to construction processes where follow up charts for complex projects can cover thousands of construction activities. The management of such a huge list requires a structure. For this purpose, project specific work breakdown structures are developed. Software tools are used in practice for the development of follow up charts, but these tools only document lists of activities and their structure. The structure of the activities is specified by a project manager. A project manager is responsible for the specification of summary tasks and the assignment of activities to summary tasks. Of course, experts know that for instance a summary task named "structural work" should not include painting activities. However, there exists no formalism that prohibits such an assignment. As a consequence, an extensive manual checking of follow up charts is required. This checking includes the verification of the list of activities concerning completeness and correctness, the verification of summary tasks concerning their structure and their assignments of activities, the verification of the inter-dependencies between activities and the verification of each activity duration. This paper is focused on a formalism to guarantee the correctness of the assignment of construction activities to summary tasks including a consistent structure of summary tasks, which is, from a mathematical point of view, the correctness of a structure in the set of construction activities. A mathematical formulation is presented based on set theory and relational algebra. This formulation is applied to construction activities. A prototype has been implemented, and an example is presented where a revitalisation project has been modelled based on the approach that is presented in this paper.

KEYWORDS: construction scheduling, information management, consistent structuring, construction processes, process modelling.

1 INTRODUCTION

The quality of follow up charts depends on the experience of the project managers that create these charts. Core information of follow up charts is the list of activities, inter-dependencies between activities, and the duration of each activity (Brandenberger, J. and Ruosch, E. (1993)). In practice, this information is captured by project managers and documented in project management software tools, specifically scheduling tools (e.g. Harris, P. E. (2006) and Jäger, M. and Holert, R. (2003)). Lists of construction activities can cover several thousand entries, thus it is necessary to structure this list by specifying summary tasks. Typically, these summary tasks are based on the work breakdown structure which is developed individually for each project. All information in follow up charts needs to be checked. This is an extensive task which is executed by experienced project managers, and this task is not supported by formal methods. For instance, an incorrect assignment of a painting activity to electrical work needs to be detected by a project manager. Of course, rules for naming construction activities reduce the specification and verification effort. The aim of this paper is to develop a formalism (1) that guarantees a consistent and correct structure of construction activities and (2) that is flexible enough so that it can be applied to the variety of construction projects.

Structuring information based on different trees is considered as faceted structuring. Kang and Paulson (1997) introduced a Construction Information Classification System (CICS) based on four facets, including facility, space, element and operation. Chang and Tsai (2003) picked up the CICS and introduced an Engineering Information Classification System (EICS) addressing the gap between construction and engineering work. The aim of CICS and EICS is to use common information, both for cost estimation and for schedule planning. Applying CICS and EICS in the context of project management requires the process relevant information units to be structured correctly. A formal method to create the structure is the main focus on this paper.

The approach presented in this paper is based on the consideration that modelling construction processes involves the use of project independent elementary information such as elementary activity descriptions or performance factors. Project independent information is collected in extensive data bases. Adequate structuring trees are required to support the user locating specific entries. Commonly accepted structures are classification systems like

STLB-Bau (2007), DIN 276 (2006) or MasterFormat™ (2004). STLB-Bau (2007) is a structured set of German text elements for standard conform specification writing. An equivalent is MasterFormat™ (2004), which is predominant in the North American building industry. DIN 276 is a German standard for cost evaluation breaking down costs into specific groups. The approach of this paper is to make use of the elementary information structure for the benefit of composed or derived project dependent information. Elementary information structure is used to guarantee the correctness of the assignment of construction activities to a project individual structure. A specific modelling technique of construction activities is presented where construction activities are in relation to elementary information. Their underlying structures are used as a framework for structuring construction activities. The theoretical background is based on set theory and relational algebra. It is described in section 2. The application to construction activities is presented in section 3. It allows a formal checking of correct assignments of construction activities to a project specific and individual structure. The approach has been used for scheduling a real project. Section 4 is focused on this. The paper concludes with a summary and an outlook.

2 MATHEMATICAL BACKGROUND

This section addresses the mathematical background of structuring sets by trees on the basis of relational algebra. The aim of this approach is to establish a methodology, which allows for structuring sets in such a way that at a specific tree node the encountered subset of elements holds tree node specific conditions. Formal checking while assigning an element to a tree node guarantees for validated and reliable tree node specific subsets. The tree itself is not preset. It is defined by a project manager. This guarantees the required flexibility. Definitions and notations are according to the work of Pahl and Damrath (2001).

2.1 *Structuring elements of a set*

In favour of reducing the complexity while managing and dealing with large sets, it is necessary to inculcate a structure to the sets. This shall be realised in four steps.

Step one: Partitioning of a set into a system of disjoint subsets

Let A be a set of elements $a_1,\cdots,a_k$. For the partitioning of the set A a set S is introduced. S consists of structuring terms $s_1,\cdots,s_m$. A relation $R_{AS} \subseteq A \times S$ is introduced. R_{AS} puts the elements of A in relation to the elements of S.

The relation R_{AS} holds the following properties: R_{AS} is left-total and right-unique. Each element $a \in A$ is assigned to exactly one element $s \in S$. Merging those elements $a \in A$, pointing to one and the same element $s \in S$, results in a system of disjoint subsets. Thus the mapping can be used to obtain a classification of the elements $a \in A$, in the mathematical sense.

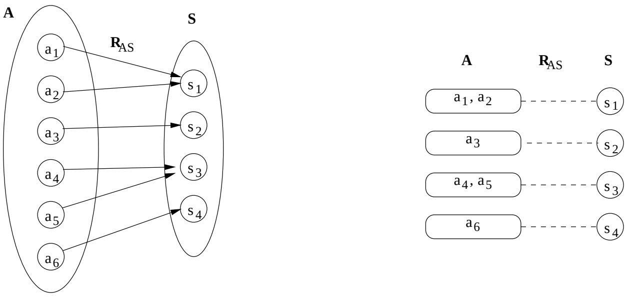

Figure 1. Classification of the elements $a \in A$.

Step two: Setting up a relation based tree structure

A relation $R_{SS} \subseteq S \times S$ is introduced, which holds the following properties:

$R_{SS} \circ R_{SS}^T \subseteq I \quad \Leftrightarrow \quad$ the relation R_{SS} is left-unique,

$R_{SS}^+ \cap R_{SS}^{+T} = 0 \quad \Leftrightarrow \quad$ the relation R_{SS} acyclic and

$R_{SS}^{*T} \circ R_{SS}^* = E \quad \Leftrightarrow \quad$ all elements $s \in S$ are pairwise quasi-strongly connected via a root element. R_{SS}^+ is the transitive closure of the relation R_{SS} and R_{SS}^* is the reflexive transitive closure of the relation R_{SS}.

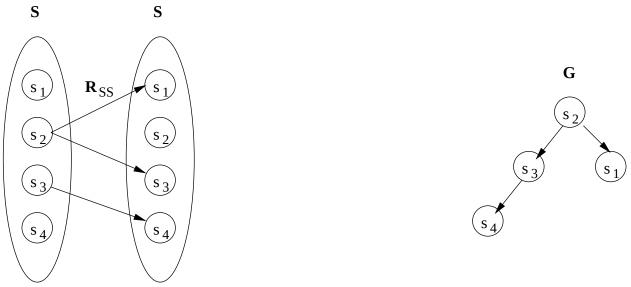

Figure 2. Relation-based tree structure.

The domain $G(S;R_{SS})$ represents an acyclic rooted graph which, due to the left-uniqueness of the relation R_{SS}, even holds the required properties for a rooted tree. Hence, the domain $G(S;R_{SS})$ is adequate to represent a hierarchical structure and is therefore denominated by *structuring tree*. The elements of S represent the set of tree nodes, which span a tree structure according to the relation R_{SS}.

Step three: Structuring elements

Assigning those disjoint subsets of A to those nodes of the tree, which are related to one and the same element $s \in S$, results in a tree structure in the set A on the basis of the structuring tree $G(S;R_{SS})$. As the assigned subsets build a system of disjoint subsets, the resulting tree is denominated by *classification tree*.

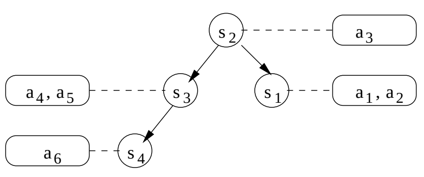

Figure 3. Assigning elements $a \in A$ to the structuring tree $G(S;R_{SS})$ based on R_{AS}

Step four: Levels of detail

Considering the tree presented in Figure 2, it is obvious to interpret the level of a tree node, which is the distance of a tree node to its root, as a *level of detail* in a technical sense. The root node represents the overall generalisation. Following a path to some tree node represents a progressive specialisation. As a consequence, the root node of the tree should be related to the entire set A, and a sub node of the tree should be related to an adequate subset of A, depending on the specific level of detail. Referring to the tree presented in Figure 3, further assumptions are necessary to result in a system of subsets according to the mentioned properties. The following building rule is set up to define the subset, which shall be related to a specific tree node. An arbitrary tree node is considered. Its associated subset shall consist of the union of all subsets related to the tree node and any of its child nodes. The following relation $R = R_{AS} \circ R_{SS}^{T*}$ describes a connectedness between the elements of A and the elements of S that holds the required properties. The relation R concatenates the subsets of A shown in Figure 3 according to the inverse tree order, applying logical OR for each edge in the tree.

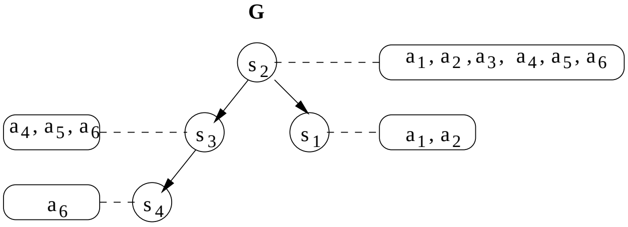

Figure 4. Assigning elements $a \in A$ to the structuring tree $G(S;R_{SS})$ based on $R = R_{AS} \circ R_{SS}^{T*}$

2.2 *Transferring the structure of a set*

Let A be a set of elements $a_1,\cdots,a_k$ and $G(S;R_{SS})$ a structuring tree as described in Figure 2. $G(S;R_{SS})$ inculcates a structure to the elements of the set A. It is considered to be an existing tree. Furthermore let Z be a set of elements $z_1,\cdots,z_m$ and $R_{ZA} \subseteq Z \times A$ a relation which describes dependencies between the elements of Z and the elements of A. The relation R_{ZA} is said to be left-total.

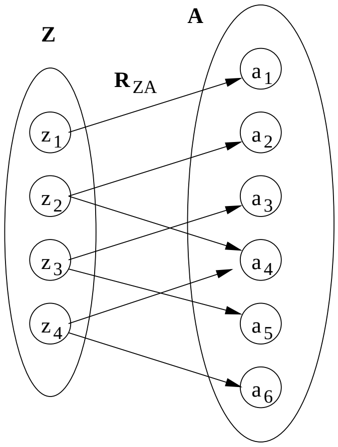

Figure 5. Relation between elements $z \in Z$ and elements $a \in A$.

Applying the concatenation operator to the relations R_{ZA} and R_{AS}, as defined in relational algebra, allows for calculating a relation $R_{ZS} = R_{ZA} \circ R_{AS}$. This relation specifies which element $z \in Z$ is connected to which element $s \in S$ via some element $a \in A$. R_{ZS} represents a structure of set Z. This structure can be computed based on a given structure of the set A. Figure 6 illustrates the connectedness of the elements $z \in Z$ and the elements $s \in S$. Hence, the tree structure presented in Figure 2 is used to derive a structure for the set Z as shown in Figure 7.

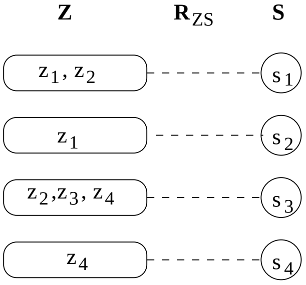

Figure 6. Connectedness between the elements $z \in Z$ and the structuring terms $s \in S$ via elements $a \in A$.

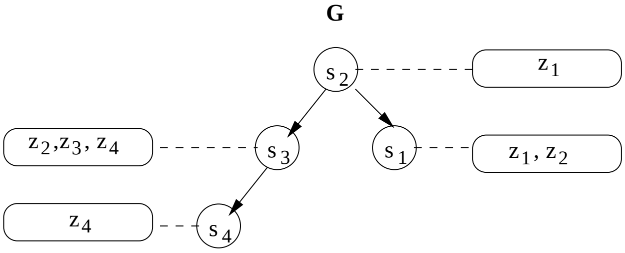

Figure 7. Transferred structure on the basis of $R_{ZS} = R_{ZA} \circ R_{AS}$.

For the purpose of structuring an unstructured set on the basis of an existing structure, it is crucial that the relation R_{ZS}, describing the remote connectedness of $z \in Z$ and $s \in S$, holds left-totality. This guarantees that for each element $z \in Z$ there is a related element $s \in S$. It is not required that R_{ZS} holds right-uniqueness. As a consequence, R_{ZS} can not be used to obtain a classification in the set of elements $z \in Z$. The main aspect of the discussed matter in this section is to describe a formal method, which allows for transferring the structuring of an element to a related element. Figure 8 shows the result of applying step 4 to the set of related elements $z \in Z$ according to the relation $R = R_{ZA} \circ R_{AS} \circ R_{SS}^{T*}$.

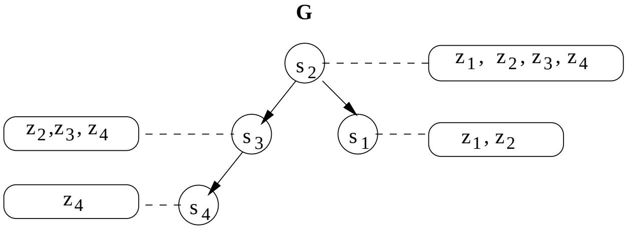

Figure 8. Transferred structure on the basis of $R = R_{ZA} \circ R_{AS} \circ R_{SS}^{T*}$.

Transferring the structure of a set to a related set results in a disposition of elements $z \in Z$ that corresponds to the technical sense of the level of detail of the structure. The principle of transferring an existing structure of a set to another set can be applied to each remotely related set.

2.3 *Method for set structuring based on formal criteria*

In this section a formalism is developed, which allows to assign checkable criteria to each node of a user defined structuring tree. As a consequence, only a validated subset of elements is related to a specific position in the tree. In other words, assigning an element to a specific tree node is only possible, if the element passes a formal checking according to the required set of criteria. The addressed user defined tree represents a structure that shall be set up to receive data. The data is unstructured. If there is a relation from the unstructured data to some other data that is already structured, then those existing structures can be used for structuring the actual data.

The following description is divided into three parts: First, sets, relations, and structuring trees are described that are named elementary data. This is followed by a description of a set and relations that are named related data. Once all the required data is introduced, the approach for an evaluable structuring tree is developed.

Elementary data

Let the sets $A_1,\cdots,A_n$ be sets of different elementary information entities. Each set $A_1,\cdots,A_n$ is related to an associated set of structuring terms $S_1,\cdots,S_n$ via a relation $R_{A_1S_1},\cdots,R_{A_nS_n}$.

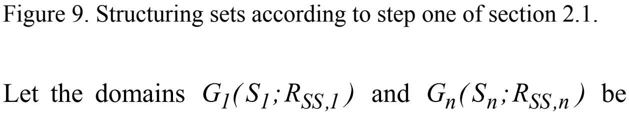

Figure 9. Structuring sets according to step one of section 2.1.

Let the domains $G_1(S_1;R_{SS,1})$ and $G_n(S_n;R_{SS,n})$ be structuring trees that are used to structure the elementary information entities. These structuring trees are considered to be existing trees. They are used to structure elementary datasets.

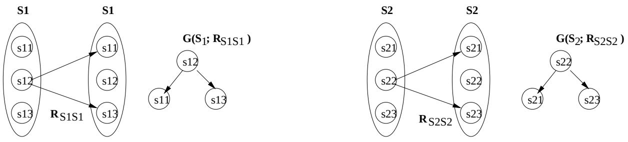

Figure 10. Structuring trees according to step two of section 2.1.

Applying the tree structures shown in Figure 10 to their corresponding set A_i on the basis of relation R_{AiSi} results in the structuring presented in Figure 11.

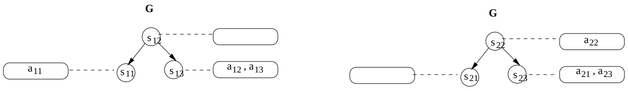

Figure 11. Tree structures applied for sets according to step three of section 2.1.

Applying the tree structures shown in Figure 10 to their corresponding set A_i on the basis of relation $R = R_{AiSi} \circ R_{SiSi}^{T*}$ results in the structuring presented in Figure 12. The tree structures shown correspond to the technical sense of *level of detail.*

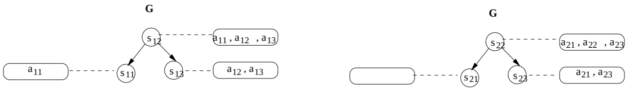

Figure 12. Tree structures applied for sets according to step four of section 2.1.

Related data

Let set Z be a set of information entities that is related to different sets of elementary information entities. The connectedness of the elements $z \in Z$ and the elements of $A_1,\cdots,A_n$ is denoted by $R_{ZA_1},\cdots,R_{ZA_n}$. Each relation $R_{ZA_1},\cdots,R_{ZA_n}$ holds left-totality.

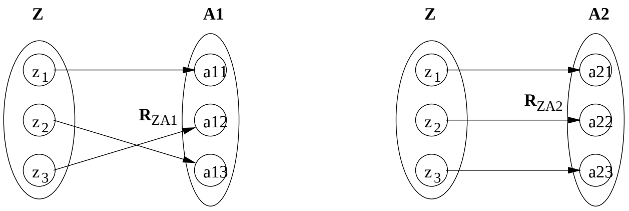

Figure 13. Relation between unstructured elements $z \in Z$ and structured elements $a \in A_i$.

Approach

Let $G(T;R_{TT})$ be a user defined structuring tree as presented in Figure 14. In contradiction to the trees $G_1(S_1;R_{SS,1})$ and $G_n(S_n;R_{SS,n})$ a user defined structuring tree is not considered as an existing tree. The structuring tree $G(T;R_{TT})$ shall be used to inculcate a structure to the elements of the set Z.

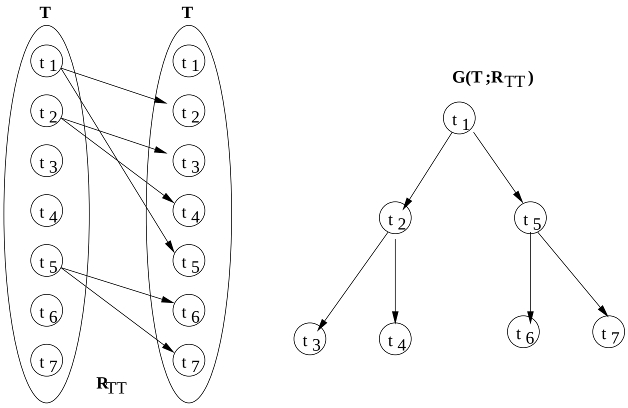

Figure 14. User defined structuring tree.

Let R_{SiT} be a set of relations, where each relation maps the elements of $S_1,\cdots,S_n$ onto the set of structuring tree nodes T. This step represents the assignment of criteria to the nodes of the user defined structuring tree. There is a single essential condition in setting up the relations R_{SiT}. Each root node of an existing structuring tree that is contained in the sets $S_1,\cdots,S_n$, needs to be related to the root of the user defined tree. These relations are presented with bold arrows in Figure 15. The necessity of this restriction is explained later in this section.

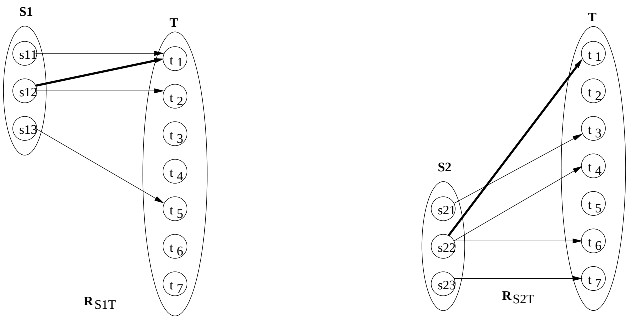

Figure 15. Assigning criteria to the nodes of the user defined tree.

Based on the introduced sets, relations, and structuring trees it shall now be described, how the existing structuring trees $G_1(S_1;R_{SS,1})$ and $G_n(S_n;R_{SS,n})$ can be used to validate inserting of related information entities $z \in Z$ into a user defined structuring tree on a formal basis. Thus the related subset of a specific tree node consists of a validated subset only.

The path from the root of the tree to a specific location represents a sum of conditions. Adding an element $z \in Z$ to a specific tree node requires each condition to be satisfied. The sum of conditions is composed of node relevant conditions and a path relevant condition. The path relevant condition represents the intersection of each node relevant condition (logical *AND*).

The node relevant conditions are based on the relation $R_{ZT} \subseteq Z \times T$ and calculated as follows:

$$R_{ZT} = \bigcup_{i=1}^{n} R_{ZAi} \circ R_{AiSi} \circ R_{SiSi}^{T*} \circ R_{SiT} \, .$$

R_{ZT} describes the connectedness between the elements of Z and the elements of T. R_{SiSi}^{T*} represents the transposed reflexive transitive closure of R_{SiSi}. If there is a remote relation zRt between $z \in Z$ and $t \in T$, then the node relevant condition for node t is satisfied. Assigning the elements $z \in Z$ to the structuring tree $G(T;R_{TT})$ based on the relation $R_{ZT} \subseteq Z \times T$ is presented in Figure 16. This accords to step tree of section 2.1.

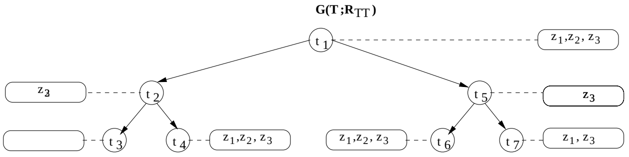

Figure 16. Node relevant conditions.

Applying the path relevant condition to the tree, presented in Figure 16, results in the final control structure, presented in Figure 17. Based on the system of subsets presented in Figure 17, it is now possible to check the assignment of an element $z \in Z$ to a user defined structuring tree. An element may only be assigned to a specific tree node if it is part of a tree node related subset.

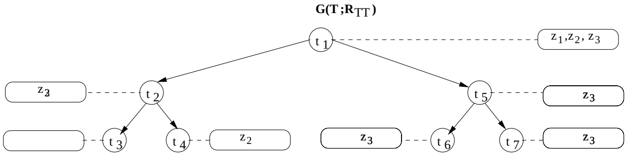

Figure 17. Concatenation of node relevant conditions by logical AND.

Depending on the assignment of criteria to a user defined tree, it can easily happen that the resulting control structure, as shown in Figure 17, contains empty subsets. In this case, the set of conditions is restrictive in such a way that there is no element that can be assigned to the node. To make sure that there is at least one node in the tree that is able to take any element, the above mentioned restriction has been set up. Each root node of an existing structuring tree that is contained in the sets $S_1,\cdots,S_n$ needs to be related to the root of the user defined tree. Without this restriction it would be possible to set up a user defined tree, which would not guarantee that each element $z \in Z$ is allowed to be added to the tree. This, however, is the crucial purpose of this tree.

If a classification is required, each element $z \in Z$ may only be added once to the tree.

3 INFORMATION MANAGEMENT

In this section, the information management of a process modelling technique for scheduling construction tasks is described. It has been developed at the Department of Process Modelling in Civil Engineering at Technische Universität Berlin. The modelling technique has been addressed in the article “Can algorithms support the specification of construction schedules?” by Huhnt and

Enge (2006). The information management is presented in this paper. It is based on sets of units of information and trees used for structuring the elements of the sets.

Different sets of information units are introduced. Some sets are project independent. They are specified only once in a construction company. Other sets are project specific. They are specified for each project. Some sets are structured by a single tree. Other sets are structured by several trees. The set of project specific construction activities is introduced as related data. A project specific construction activity is an activity executed at a specific component. There are different types of components. The structure of the project specific set of construction activities is checked according to section 2.3.

3.1 *Project independent sets of information units*

Starting point for modelling a construction process are sets of elementary data. This includes a set of elementary activities, a set of status values, and a set of component types. Elementary activities are non-divisible and trade specific units of information. Performing an elementary activity is time consuming and describes the transformation of a building component. The state which is reached after terminating the activity is described by a status value. In this respect, the set of status values represents a set of state descriptions, each corresponding to an elementary activity.

Based on the set of activities and the set of status values, component oriented manufacturing processes are modelled. These are denoted by *component types*. A component type describes the manufacturing of a component of a specific type in a generalised way (Enge 2005).

3.2 *Project dependent sets of units of information*

Buildings are the matter of projects in civil engineering. In the context of the underlying modelling technique, a building is decomposed into a set of components. Each component is assigned to a component type. The elements of the set of components are elementary information units. Based on this set of components and their related component types, a set of process relevant activities is generated. Grouping elementary activities in such a way that they form an appropriate unit in the context of scheduling leads to a set of summarised activities. Each set of summarised activities forms a unit of information and is denoted as *construction activity*.

Each construction activity consists of at least a single elementary activity that is executed at a specific component. The elementary activity and the addressed component form a tuple. A construction activity can be related to more than one tuple. Modelling pre-requisites and results of construction activities based on components in states is described in Huhnt and Enge (2006).

3.3 *Structuring trees for elementary units of information*

The following trees are introduced to structure the elements of the sets of elementary information units:

- *Trades*
  is an elementary structuring tree, structuring according to STLB-Bau (2007). The tree is project independent. It is used to structure the elements of the set of elementary activities and the set of status values.
- *Type of building works*
  is an elementary structuring tree, structuring according to the type of work that is executed at a component, e.g. construction, deconstruction, conversion, temporary construction, etc. The tree is project independent. It is used to structure the elements of the set of components.
- *Location*
  is an elementary structuring tree, structuring according to the location. The tree is project specific. It is used to structure the elements of the set of components.
- *Type of components*
  is an elementary structuring tree, structuring according to DIN 276 (2006). The tree is project independent. It is used to structure the elements of the set of component types.

The set of components is structured based on *Type of building works* and *Location*. Due to the relation of a component to a component type and via a component type to activities and status values, all other introduced structuring criteria are available for components as well as for component types. Transferring structures of sets to remotely related sets is described in section 2.2.

3.4 *Structuring trees for construction activities*

The following tree is introduced to structure construction activities:

- *Tasks*
  is an evaluated structuring tree, structuring according to a user defined pattern. The tree is project specific. It is used to structure the elements of the set of construction activities.

The tree *Tasks* represents a special tree. So far, the addressed structuring trees conform to the general understanding of a structuring tree. They represent a hierarchical structure that can be used to hold elements. There is no underlying logic that validates the composition of tree node specific subsets. In other words, there is no mechanism that guarantees that a specific tree node only holds elements corresponding to the description of the tree node. The element disposition depends on the user and its correctness needs to be checked manually. This approach is prone to errors. One possibility of avoiding inconsistent dispositions is to set up generalised trees with predefined related subsets. This concept is applied e.g. to manage the project independent information. As this approach does not allow any flexibility, it is restricted to few use cases only. Construction processes are individual, each one with different requirements concerning the structuring. Hence, it is impossible to set up a general structure to organise any construction processes. A different approach is necessary to satisfy the requirements concerning flexibility. For this purpose, the special approach presented in this paper has been developed. The user defined structuring tree which has been introduced in section 2.3 is an adequate structure. This tree allows for setting up a project specific structure. The concept of the described user-defined structuring tree is applied to the tree *Tasks*. A construction activity can only be inserted into the tree if node and path relevant conditions are satisfied.

The specification of the tree *Tasks* requires the definition of the tree and the assignment of criteria to the tree nodes. The set of assignable criteria is made up by the node sets of the trees *Trades, Type of building works, Location,* and *Type of components*. These sets represent the different sets of structuring tree nodes that are used for structuring the set of components, the set of elementary activities and the set of component types. The connectedness described in section 3.2 between construction activities and both, components and elementary activities, as well as the connectedness of components and component types, allows for structuring construction activities on the basis of all of these sets. Once the tree is set up, process specific construction activities can be inserted. The structuring results in a classification of the process relevant construction activities if each element is assigned only once.

In this context, two significant differences compared to conventional approaches shall be pointed out. On the one hand the set of process relevant activities is complete, assuming that the decomposition of the project into components is complete, and on the other hand the structuring guarantees for reliable subsets. These two aspects significantly increase the quality of information management in modelling construction processes.

4 USE CASE

A pilot implementation has been developed based on the approach described in this paper. The pilot implementation includes a modelling technique where construction activities are specified independently, and the technological inter-dependencies between these activities are computed. Correctness and completeness of the calculated inter-dependencies can be guaranteed with respect to user input. A proposal for a follow up chart is generated which can be used as a basis for further scheduling activities.

The pilot implementation has been used to develop a schedule for a real construction project. The result has been compared to a conventionally developed project schedule. The project covers the reconstruction of a nine story office building with approximately 1000 m² surface per floor. The building is torn down to its skeleton and rebuilt again. The tasks cover outside facilities, the replacement of the facade, the technical building equipment and the reconstruction of the interior. The reconstruction of floor one to four is not part of the project. This work is treated in a separate project.

In the context of the process modelling, the building has been decomposed into 197 components of 44 component types. Generating the set of project relevant construction activities resulted in 363 elements. The underlying process of the conventional schedule consisted of 216 activities. The assimilation of the levels of detail has been realised by defining adequate summary tasks in the set of construction activities. The definition of summary tasks is part of the structuring of the construction activities. For this purpose the tree *Tasks* has been set up. Each node of this tree represents a summary task in the conventional sense. The tree consisted of 84 nodes. To ensure reliable subsets, each node has been configured by a single criterion.

As a consequence of using the described approach, each summary task contains only those construction activities that semantically fit to the summary task. Inconsistencies where a construction activity, which is to be executed on the first floor, is assigned to a summary task of the second floor cannot occur any more. A manual checking is not necessary to guarantee the correctness of the structure.

5 CONCLUSION

This paper describes in detail the mathematical foundation of relation-based set structuring, using specific trees. Based on this concept progressively different structuring techniques are introduced. The main focus is on a structuring, which allows to assign checkable criteria to each node of a user defined structuring tree, so that consequently only a validated subset of elements is related to a specific position in the tree.

The described structuring techniques are applied to the information management of a process modelling technique developed for scheduling construction activities. Based on data of a real construction process the modelling technique has been applied, and the resulting schedule has been compared to a conventionally developed schedule. The results of the investigations, stated in this paper, are focused on structuring aspects.

It is assumed that a construction process involves project independent and project specific information. Depending on the type of information, different structures are applied. Particularly, the project specific structure of construction activities has been of special interest. It turned out that structuring construction activities on the basis of a structuring tree, satisfies the required flexibility of individual construction processes and contributes to the clarity of the underlying information management. The ease in locating specific activities increases performance while modelling.

Further investigations are necessary in the context of evaluating subsets of tree nodes. Adding an element to a tree node requires node specific conditions and path specific conditions to be satisfied. A specific evaluation rule has been applied here. Further rules still need to be investigated. Different combinations of the logical operators may be considered.

REFERENCES

Brandenberger, J. and Ruosch, E. (1993). “Ablaufplanung im Bauwesen”, 3., Aufl., Baufachverlag beim hep verlag

Chang, A. Shing-Tao and Tsai, Ya-Wen (2003). “Engineering Information Classification System”, ASCE, J. Constr. Eng. Manage., Vol. 129, No. 4, July/August 2003, pp. 454-460

DIN 276 (2006). “Kosten im Hochbau”, Deutsches Institut für Normung e. V., Berlin : Beuth-Verlag

Enge, F. (2005). “Zustandsmodellierung als Grundlage für Ausführungsterminpläne”, Schley, F., Weber, L. (editor), Forum Bauinformatik, Cottbus, pp. 123-130

Harris, P. E. (2006). “Planning Using Primavera Project Planner P3 Version 3.1”, Rev 2006, spiral bound, Eastwood Harris Pty Ltd, Australia

Huhnt, W. and Enge, F. (2006). "Can algorithms support the specification of construction schedules?", ITcon Vol. 11, Special Issue Process Modelling, Process Management and Collaboration, pg. 547-564, http://www.itcon.org/2006/39, 27.02.2007

Jäger, M. and Holert, R. (2003) "Microsoft Project 2000/2002 - Das Profibuch", 1. Aufl., Microsoft Press

Kang, L. S. and Paulson, B. C. (1998). "Information Management to Integrate Cost and Schedule for Civil Engineering Projects", ASCE, J. Constr. Eng. Manage., Vol. 124, No. 5, September/October 1998, pp. 381-389

MasterFormat™ (2004). "MasterFormat™ - 2004 Edition Numbers & Tittles" Construction Specifications Institute, CSI-*Net*: http://www.csinet.org/s_csi/docs/9400/9361.pdf, 27.02.2007

Pahl, P. J and Damrath, R. (2001). "Mathematical Foundations of Computational Engineering: A Handbook", Springer-Verlag, Berlin

STLB-Bau (2007). "STLB-Bau Dynamische BauDaten", GAEB – Gemeinsamer Ausschuss Elektronik im Bauwesen, http://www.gaeb.de, 27.02.2007