Neural networks in the re-engineering process based on construction drawings

S. Komorowski
HOCHTIEF Construction AG, Bauen im Bestand NRW, Essen, Germany

V. Berkhahn
Institute of Computer Science in Civil Engineering, University of Hannover, Germany

ABSTRACT: In this paper an approach is presented to digitize a drawing, to build up geometric and topologic models, to recognise construction parts and to interpret dimension lines and inscriptions. All recognized parts are transformed into a three-dimensional geometric model which provides all necessary geometric information for a product model. The recognition process of construction parts is based on a line search and topological analysis, which are not suitable for the recognition of drawing inscriptions and hand writings. Therefore, the information of dimension inscriptions has to be neglected in former case studies. Because dimension inscriptions deliver significant information about the dimensions of construction parts, a neural Kohonen network is implemented and adapted in order to recognise inscription text. Finally the gained information about dimensions is related to significant details of construction parts.

1 INTRODUCTION

There have been great efforts in the development of software tools concerning the design and the realization of new buildings. Nowadays, an engineer has access to different kinds of computer based product models and CAD-models. But if he has to deal with the management, reorganisation or recalculation of older buildings, most times there are no digital data available. The only information about the building could be gained from in situ measurements or from paper based drawings, which have to be analyzed manually by the engineer. This is a very time-consuming and boring job hardly supported by effective software tools in general. In order to close this gap between drawings of an existing building and a digital product model an approach is presented in this paper to digitize a drawing, to build up geometric and topologic models, to recognize construction parts of the building and to interpret dimension lines and inscriptions.

Case studies presented by Berkhahn et al. (2003, 2004) are based on a line identification process and topological information about the identified lines. The corresponding topologic and geometric models are used to interpret drawing lines and to identify construction parts. Unfortunately, this approach based on the analysis of topological information is not sufficient for the recognition of inscriptions. Consequently, the information of dimension inscriptions has to be neglected in the corresponding case studies presented by Berkhahn et al. (2004). In these case studies the dimensions of construction parts are gained exclusively by scaling the scanned lines. This approach leads to remarkable inaccuracies for dimensions of construction parts and in some cases the topological correctness of the geometric model could not be assured.

To overcome these difficulties of inaccuracies the inscriptions of drawings have to be interpreted, which is realized by merging the information obtained by a neural network and the topological model. Therefore a Kohonen network (Heaton, 2004) is adapted to recognize standard lettering as well as handwriting in drawings (Komorowski, 2004). The information gained from the Kohonen network and the topological model is merged in order to identify single characters, to combine single characters to inscriptions and finally to relate the inscriptions to dimension lines and construction parts. Finally all recognized parts are transformed into a three-dimensional geometric model which provides all necessary geometric information for a product model.

This contribution is an enhancement of a research paper presented by Berkhahn & Esch (2003) at the CIB W78 conference “Construction IT Bridging the Distance” on Waiheke Island, New Zealand. In this actual contribution theoretical basics and practical applications of merging neural networks and topological models and of the re-engineering process are presented.

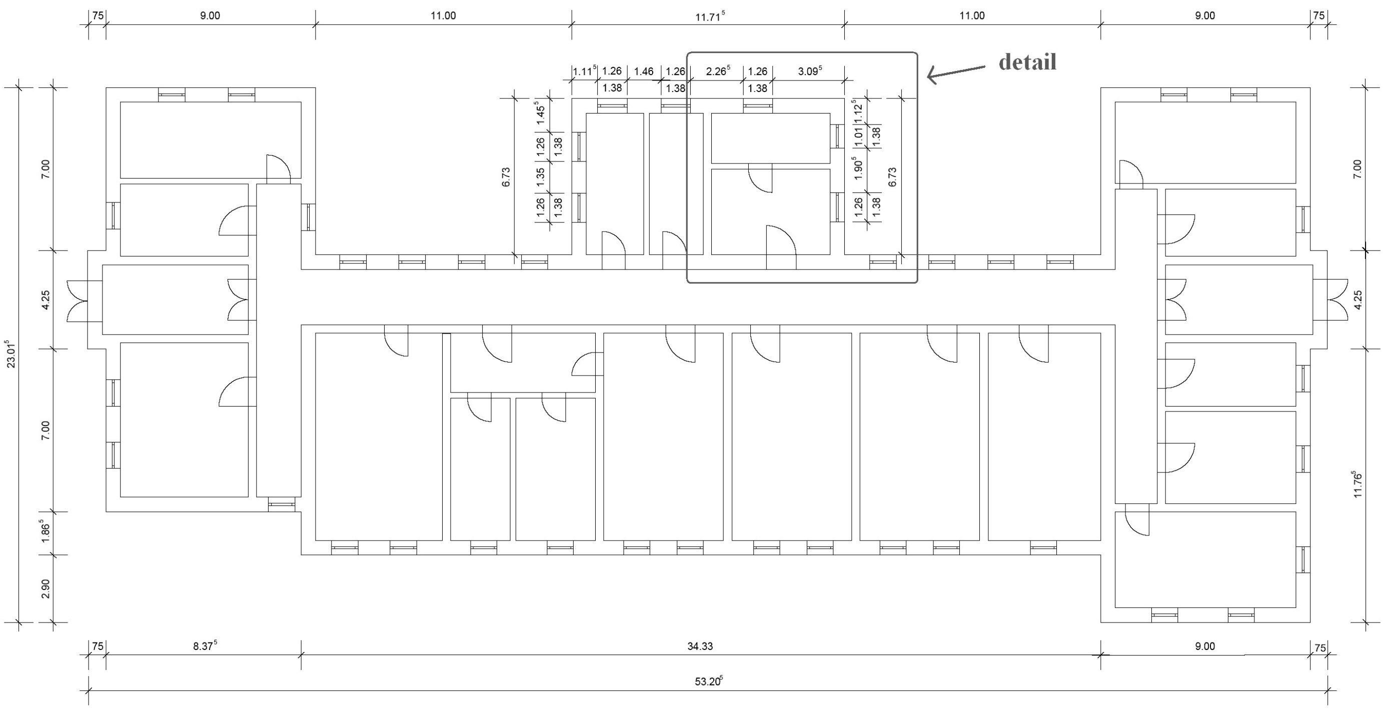

Figure 1. Scanned ground floor plan of the barrack building

All algorithms and methods are implemented with the Java programming language. A case study of an existing building demonstrates the usability and efficiency on the outlined approach. This case study concerns an old barrack build in the beginning of the last century and used nowadays as offices of the University of Hannover. A ground floor plan of the first storey of this building is shown in Figure 1.

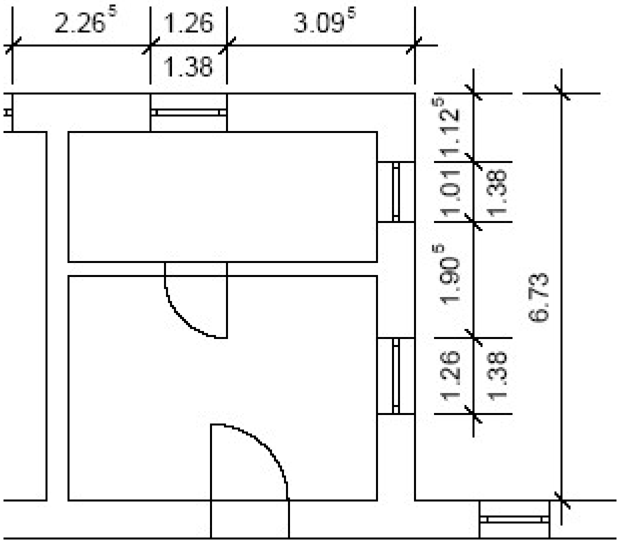

Figure 2. Detail of the ground floor plan

The identified inscriptions, dimension lines and in particularly the corresponding relationship to construction parts are illustrated in detail. For a demonstrative explanation a detail (Figure 2) of the ground floor plan is used. Finally the whole ground floor plan is shown in Figure 14 as result of the identification process after the integration into the IFC-product-model. Product modeling is one of the key issues of the DFG priority program 1103 (DFG 2005) concerning network based co-operative planning processes in structural engineering. Consequently, the actual research work has been considered in strong correlation to this priority program with a special focus to the re-engineering process of existing buildings.

2 NEURAL NETWORKS

2.1 *The Kohonen neural network*

From the multitude of neural networks the Kohonen neural network, which is named after its creator Tuevo Kohonen, suits very well for the recognition of characters. The Kohonen network is a quite simple and fast, but also effective network for this purpose. In contrast to other neural networks the output of a Kohonen network does not consist of the output of several neurons, but it chooses one neuron as a winning neuron. In this way the Kohonen network classifies samples into several groups (e.g. digits or letters). The network only exists of two layers: An input layer which gets information from the outside as well as an output layer which gives information to the outside. When a pattern is presented to the Kohonen network one of the output neurons is selected as a winner. This neuron is the output of the network and corresponds with one of the classified groups. In Figure 3 a simple Kohonen network with three input neurons and two output neurons is shown.

2.1.1 *Determine the output*

By presenting an input vector to the Kohonen network, which contains the values for the input neurons, the network can determine an output for every output neuron. The output of the output neurons will be saved in the output vector **y**. The number of components in the input and output vector therefore is equal to the number of input and output neurons. The Kohonen network requires inputs normalized to a range between -1 and 1.

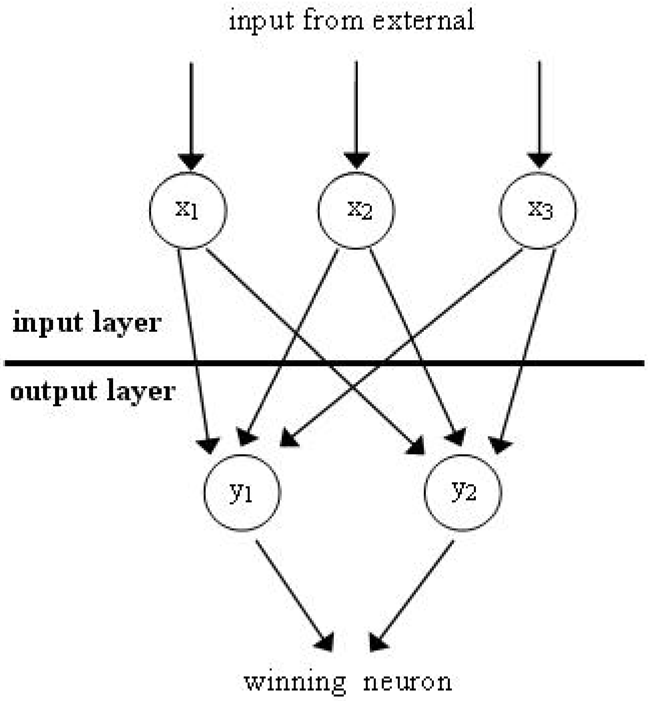

Figure 3. A simple Kohonen network

The normalization factor NF, used in (3), is the vector length of the input vector. Consequently, the value of an output neuron is determined by scalar multiplication of the input vector with the connection weights matrix. For the Kohonen network in Figure 3 the input vector **x** is

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \qquad (1)$$

and the connection weights matrix **w** is

$$\mathbf{w} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}. \qquad (2)$$

The matrix **w** consist of components w_{ij} where the index i denotes the input neuron and the index j the output neuron. The output vector **y** is determined by

$$\mathbf{y} = \begin{pmatrix} y_1 & y_2 \end{pmatrix} = \frac{\mathbf{x} \cdot \mathbf{w}}{\mathrm{NF}} \qquad (3)$$

respectively

$$\mathbf{y} = \frac{\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{21} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}}{\mathrm{NF}}. \qquad (4)$$

Consequently, the output values have to convert to bipolar values. Because the input of the network was normalized in advance, only the value of one has to be added and the result divided by two. The winning neuron is the one with the largest value. Thus, the presented pattern is classified in the group which is represented by this winning neuron.

2.1.2 *The training process*

Before a network is able to recognize any pattern it has to be trained. In the case of the Kohonen network this training is unsupervised and quite fast. For the training process training data sets are needed. Every time training data are presented to the network the connection weights are adjusted based on the result of this item of training data. The adjustment of the weights should produce a network that will yield more favorable results the next time the same training data is presented.

A training data set contains of an input vector which holds the pattern of a character and the character that is presented. Because the training is unsupervised, there is no output neuron given in the training data set. The network determines the output neuron itself. This winning neuron represents the classified group representing the character.

The training process consists of repeating cycles. It continues until one of two criteria is satisfied: If the calculated error e is below acceptable level a best weight matrix is found and the process stops. On the other hand, if the error rate has only changed by a very small amount the cycle is aborted, the entire weight matrix is reset to new random values and the training process begins again.

The error e_j is calculated for every training data set and every output neuron as follows

$$e_j = \left\| \mathbf{x} - \mathbf{w}_j \right\|, \qquad (5)$$

where w_j represents the j^{th} column of the matrix **w**. The error e_j is not an error in the normal sense of words. It is only a percentage number which gives an idea of how well the Kohonen network is classifying the input in output groups. If the error is not acceptable the connection weights have to be readjusted. The original method for adjusting the connection weights, which was proposed by Kohonen, is the additive method:

$$w_{ij}^{k+1} = \frac{w_{ij}^{k} + \alpha \cdot x_i}{\left\| w_{ij}^{k} + \alpha \cdot x_i \right\|}. \qquad (6)$$

The variable w^k is the weight of the neuron, and the variable w^{k+1} is the new weight of the same neuron. The learning rate α is a constant between 0 and 1, which determines the speed of the training process. Setting the learning rate to a larger value will cause the training to progress faster. But a large value could cause the network to converge never.

Most times the learning rate is set to a value of 0.4 up to 0.5 or it will be set relatively high in the beginning and decreases throughout the training process.

Most times the additive method works quite well. Though, in cases this method fails to converge the subtractive method can by used:

$$w_{ij}^{k+1} = w_{ij}^{k} + \alpha \cdot (x_i - w_{ij}^{k}) \ . \qquad (7)$$

This method is used in the program presented in this contribution.

2.2 *OCR in construction drawings with the Kohonen network*

For the optical character recognition (OCR) in construction drawings an existing Java implementation of a Kohonen network (Heaton 2004) is used. This program projects a written character onto a grid with given height and width. This process is called down-sampling and assures that the size and position of the character is no issue. For the down-sampling a frame is drawn around the character, in which the grey scale value of all pixels is determined. Based on this grey scale value a pixel is classified to be part of a digit (= digit pixel) or to be part of the background (= background pixel). The character is projected onto the grid by coloring all caskets of the grid black containing digit pixels. Every casket of the grid stands for one input neuron.

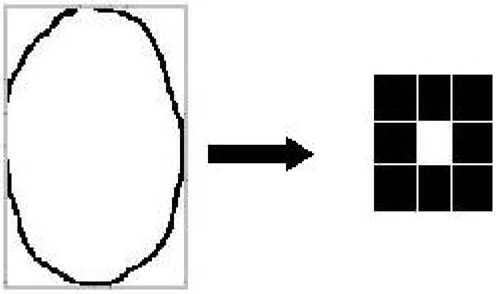

Figure 4. Down-Sampling

The neuron will be assigned a value of -0.5 if it is white or a value of 0.5 if it is black. Thus, the input vector for the grid in Figure 4, which represents the digit 0, will be as follows

$$\mathbf{x} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 & -0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix} \ . \qquad (8)$$

This input vector is given to the Kohonen network which recognizes this pattern as the digit 0. Because characters in construction drawings normally are not only readable from the bottom, but also from top or sideways, a program for OCR in drawings should be able to deal with this as well. Thus, the implementation is enhanced by these features.

3 DIMENSION MODEL

3.1 *Identification of dimensions chains*

3.1.1 *Dimension values*

For character recognition a Kohonen network is used as described before. But to determine a dimension value it is not sufficient to recognize single digits but the whole number has to be interpreted. This is the point where topological and geometric properties are taken additionally into account.

Most numbers do not consist of one single digit. Therefore, it is necessary to determine how many digits a number contains and how they are arranged. Then the number must be calculated.

To confine single digits the same method is used as for the line identification (Berkhahn et al. 2003, 2004). According to its grey scale value every pixel is either a digit pixel or a background pixel.

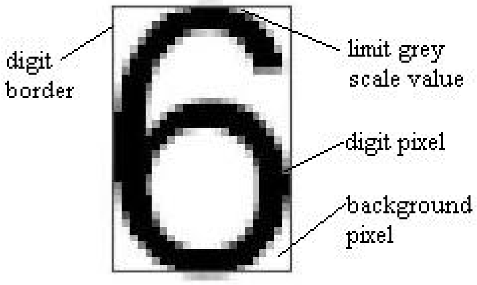

Figure 5. Digit borders and different grey scale values

A well defined area is searched for a digit pixel. Hence, its eight neighboring pixels are examined as well. In this manner all pixels of a digit as well as the minima and maximal pixel coordinates (= digit borders) can be found and saved. The same procedure is done for the next digit pixels. With the pixel borders the digit pixels are presented to the OCR and a single digit is recognized.

All digits of a number will be saved in a vector randomly. For calculating the number the digits have to be resorted. For this purpose the geometrical center of every digit is determined and the right order of the digits can be set. Along with the digits the place of a decimal point will be saved. Also the existence of an exponent is examined. Therefore the centers of the digits are compared to each other. With this information it is quite simple to calculate the value of the number.

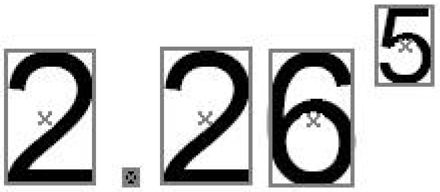

Figure 6. Digit borders and centers

3.1.2 *Dimension line points and dimension lines*

For the identification of dimension lines the topologic model is used. This topologic model is described in detail by Berkhahn & Esch (2003). Thus, here only a short summary for a better comprehension is given. The topologic model of a pixel-based drawing contains information about the relationship of different lines of the drawing. The model is composed of topologic sections which are bordered by two topologic points. Every topologic section refers to exactly two points and every topologic point refers to all connected sections.

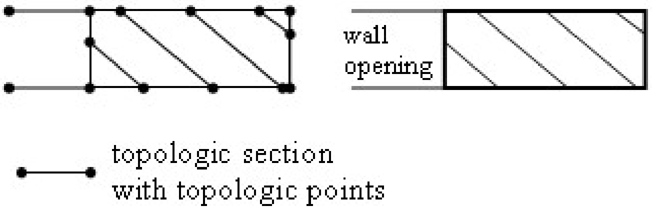

Figure 7. Topologic model

In the topologic model points are searched for which restrain a dimension line, in the following they are called dimension line points. Afterwards the dimension lines between the dimension line points are identified. The dimension model of a drawing is built up by dimension line points and dimension lines.

The ending of a dimension line in a construction drawing is marked by a dimension line termination. In the majority of cases this dimension line termination is a short diagonal slash at the corner of the dimension line and the projection line. By looking at the topologic model of a dimension line termination you recognize four topologic sections (dimension line and projection line), which have a right angle to each other, as well as two topologic section (dimension line terminations) with an angle of approximately 45° or135° to the other sections.

These geometric properties are used to find dimension line points. All found dimension line points are saved in a vector. Every dimension line point refers to one topologic point and a topologic point can only refer to one dimension line point.

After the identification of dimension line points dimension lines are generated. Therefore all topologic sections with the following property are identified: Both topologic points of the topologic section have to refer to a dimension line point. In this case a dimension line with these two dimension line points is created and saved to the dimension model. The coherences between the topologic model and the dimension model are shown in Figure 9.

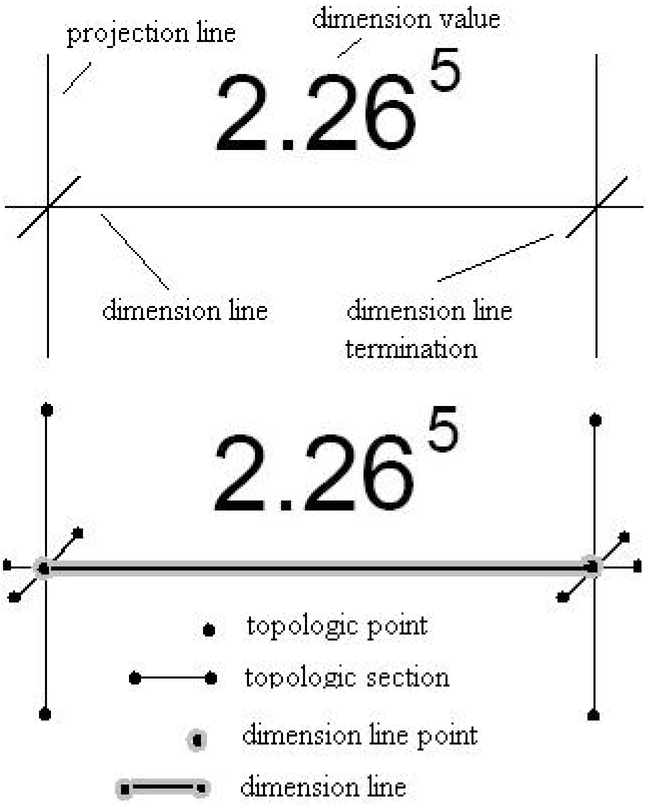

Figure 8. Dimensioning of a drawing with corresponding topologic model and dimension model

Now it is possible to identify the dimension values which correspond to the respective dimension line. Thus, a well defined area on both sides of the dimension line is searched for a dimension value. Depending on the place of the value it will be saved as the length or the height of the measured construction element.

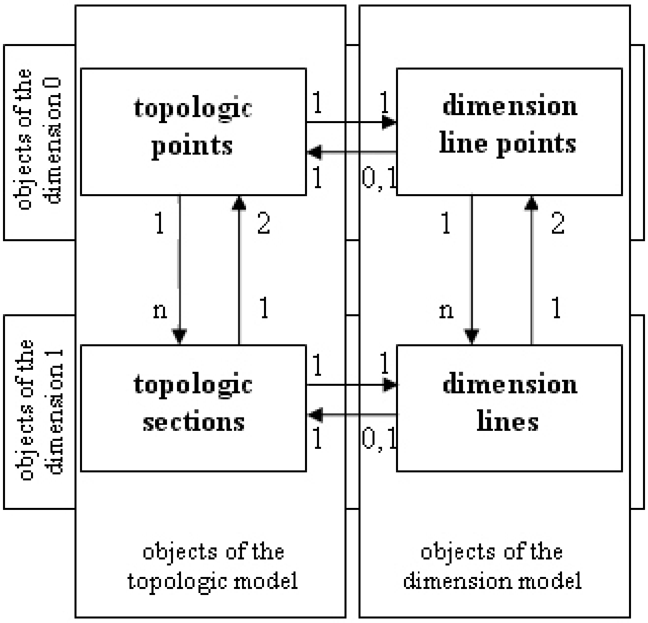
objects of the dimension 0
objects of the dimension 1
topologic points
dimension line points
topologic sections
dimension lines
1
1
1
0,1
1
2
n
1
1
2
n
1
1
1
1
0,1
objects of the topologic model
objects of the dimension model

Figure 9. Coherences between the topologic model and the dimension model

3.2 *Allocation of construction elements*

As described in Berkhahn et al. (2004) construction elements are identified by searching for closed loops of topological sections. These construction elements can be straight walls but also wall corners or junctions. In a second step these construction elements are split up into simple construction elements which has exactly four corner points (= construction element point) and four edges (= construction element sections). This simplification is performed in order to determine quite easily centerlines of construction elements and to facilitate the assembly of independent and dependent construction parts. These later procedure steps are explained in section 4.

Every dimension line holds information about one or more dimension of one or more construction elements. I.e. it corresponds to construction element section of one or more construction elements. For finding the corresponding construction elements of a dimension line in a paper based drawing an engineer will look at the projection lines. Along an imaginary line, that elongates the projection line, all points referring to the dimension line are found. Accordingly the existing construction elements are searched for construction element points, which are on a rectangular line to a dimension a line and cross a dimension line point. Hence, one or more construction element points are found for every dimension line point. An example of this is shown in Figure 10.

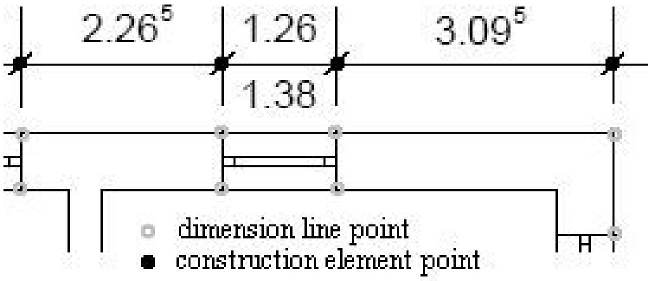

Figure 10. Dimension line points with associated construction element points

The next step is checking out every dimension line and its dimension line points, if a construction element section exists between the associated construction element points. In this case the identified construction element section will be measured by the dimension line and therefore the dimension line now refers to the construction element section and vice versa.

Furthermore, a dimension line may refer to more than one construction element section. In this case construction element sections are searched fulfilling the following criteria: 1.) They are parallel to the dimension line. 2.) They are arranged between the dimension line points of the dimension line and between the construction element points, respectively. The coherences between the topologic model and the dimension model are shown in Figure 11.

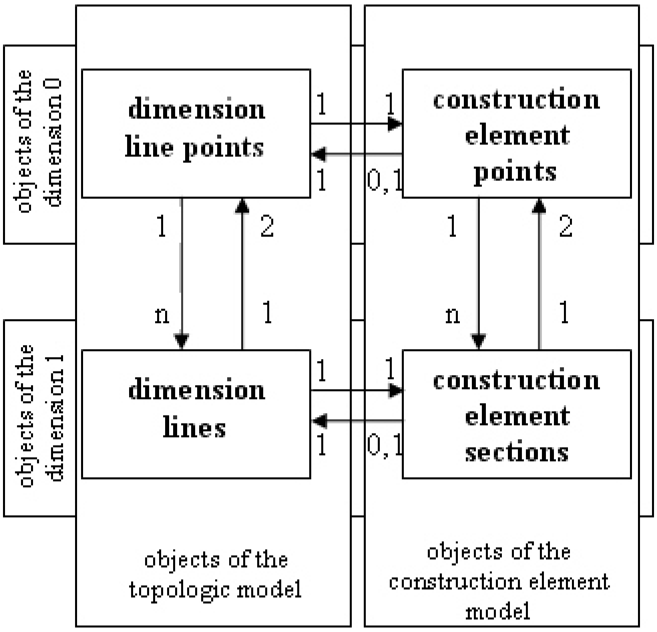
objects of the dimension 0
objects of the dimension 1
dimension line points
construction element points
dimension lines
construction element sections
1
1
1
0,1
1
2
n
1
1
2
n
1
1
1
1
0,1
objects of the topologic model
objects of the construction element model

Figure 11. Coherences between the dimension model and the construction element model

3.3 *Inconsistencies*

Finally, the identified dimension lines are displayed in a frame, in which the dimension values are colored. Black values quote that the values are alright, grey colored values quote an error. These indications result from a comparison between the identified dimension values and the user-given scaling factors. In this way all errors resulting from the identification process are recognized. But in the same way inconsistencies in the actual drawing can be determined and made visible to the user. Now the user can correct the errors and the result will be an accurate drawing.

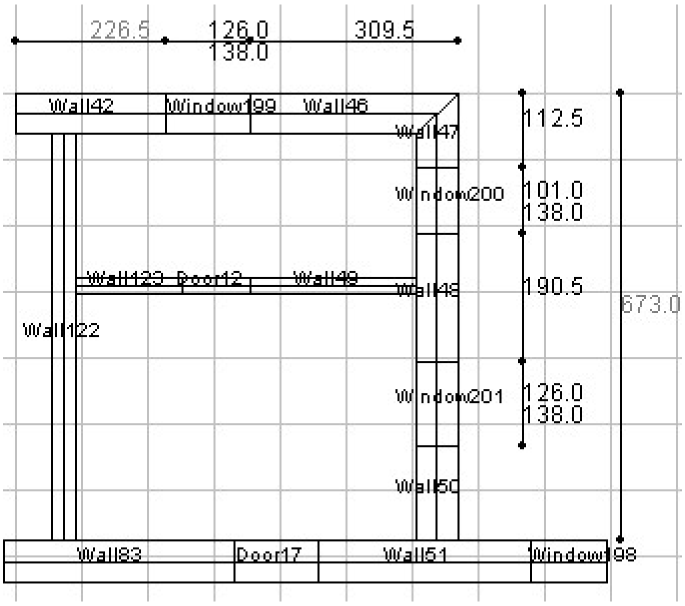

Figure 12. Identified detail of the barrack building

4 IMPORT TO A PRODUCT MODEL

For the import of construction parts into a product model the system centerline has to be determined. This is quite simple for the simple construction elements with its four edges. After the determination of the system centerline of every construction element independent and dependent elements are defined. Independent construction parts exist without any reference to any other part. In contrast to this, window or door construction parts are dependent parts, which are defined by a reference to a wall construction part. A ground floor plan implies only the information about the wall parts beside a window or a door part. For the import into a product model the wall parts at both sides of a window or a door part are merged to one independent entire wall part. The corresponding dependent windows and door parts refer to this entire wall part. The corresponding dependent window and door parts refer to this entire wall part (Figure 13).

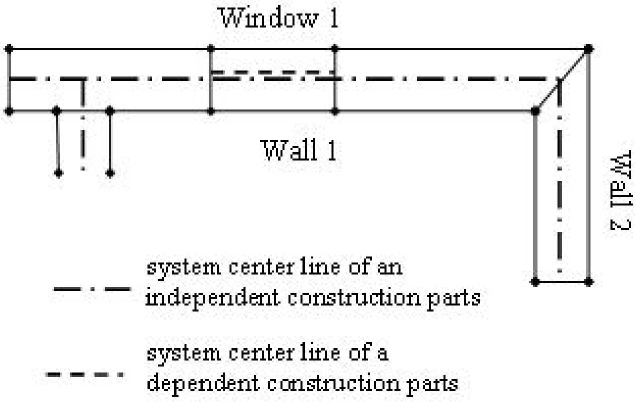
Window 1
Wall 1
Wall 2
system center line of an independent construction parts
system center line of a dependent construction parts

Figure 13. Dependent and independent construction parts

The identified construction parts, defined by their centerline and thickness, are imported into the geometry kernel of a product model. Additional information about the height between floors and the heights of window parapets or of door lintels has to be defined generally for the whole building. Exceptions of the standard heights have to be specified explicitly for the relevant construction parts.

In addition to geometric and topologic data product information of the building is managed by the product model, which is relevant for all states of design, planning, construction, creation and usage.

5 CONCLUSION

The testing of the implemented software tool with a ground floor plan of a real building has shown, that the identification of construction elements and their dimensions and the transfer into a product model is generally possible. Yet hitherto the appropriate criteria for the identification have to be adjusted to the particular drawing. As neuronal networks have been shown very effective within the identification of dimension values, succeeding consideration should involve neural networks within the identification of construction elements. Though construction elements always have the same criteria of recognition, they are seldom exact identically. Therefore, the application of neuronal networks in combination with information gained from geometric and topological models shows a great promise.

Furthermore, the user interface of the software tool has to be adapted to the requirements and conditions of the every day practice. For the different input values sensible standard values have to be provided or appropriate algorithms for automatic determination of the input values have to be implemented. Particularly, dealing with recognized inconsistencies between drawing, dimension and inscription requires fine tuning of user interaction.

6 ACKNOWLEDGEMENT

The authors would like to thank Prof. Ernst Rank and Matthias Schleinkofer, Lehrstuhl für Bauinformatik at the Technical University of Munich, for the fruitful cooperation concerning the data import into a product model. Based on their long experience in product models a common interface was developed and as result the recognised construction parts are visualized within the IFC-product-model (Figure 14).

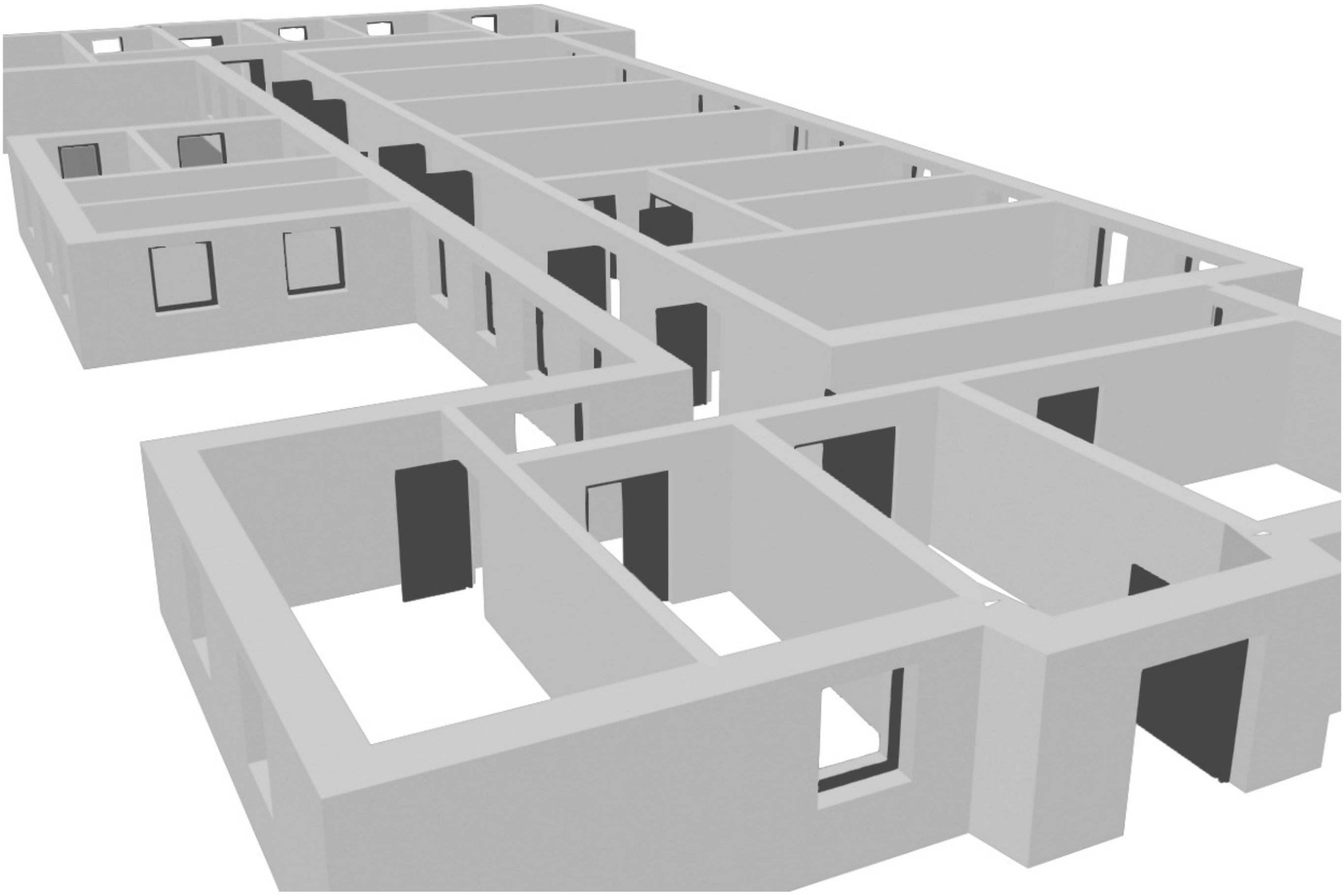

Figure 14. IFC-product-model of the barrack case study

REFERENCES

Berkhahn, V. & Esch, C.; 2003. Re-Engineering of Objects in Construction Drawings. In: R. Amor (ed.), *Construction IT Bridging the Distance, CIB Report: Publication 284, Proc. of CIB W78's 20th International Conference on Information Technology for Construction, Auckland*

Berkhahn, V., Kinkeldey, C., Schleinkofer, M. & Rank, E.; 2004. Re-Engineering Based on Construction Drawings, From Ground Floor Plan to Product Model. *Proceedings of the Xth International Conference in Computing in Civil Engineering, ICCCBE 2004, Weimar*

Berkhahn, V.; 2004. Recalculation of Construction Elements Based on Drawings; *Proceedings of SEMC 2004, The Second International Conference on Structural Engineering, Mechanics and Computation, Cape Town*

Berkhahn, V. & Komorowski, S.; 2005. Analyse von Bauzeichnungen; Dokumentation der Methoden und Software. Institute of Computer Science in Civil Engineering, University of Hannover

Deutsche Forschungsgemeinschaft (DFG); 2005. Network-based Co-operative Planning Processes in Structural Engineering, DFG Priority Program 1103, www.dfg-spp1103.de

Heaton, J.; 2003. Programming Neural Networks in Java. www.jeffheaton.com

Industrieallianz für Interoperabilität e.V. (eds.). IFC - Industry Foundation Classes. Management Übersicht, www.iai-ev.de/ifc/documents/overview_management.htm

Komorowski, S.; 2004. Rechnergestützte Bestandsaufnahme, Identifizierung vom Bauteildimensionen in papierbasierten Baubestandsplänen. Studienarbeit, Institut für Bautechnik und Holzbau and Institute of Computer Science in Civil Engineering, University of Hannover