Recognition of paper-based conceptual models captured under uncontrolled conditions

Stefan Oppl[1], Chris Stary[1], Simon Vogl[2]

***Abstract*— Today, modeling and subsequent digital model representations are essential constituents in collaborative endeavors on organizational change. Once created, models need to be digitized for shared stakeholder understanding and further processing. Whenever paper serves as carrier medium it is likely to disrupt further processing, elicitation and modeling. While digital environments support transformation processes in collaborative modeling from its very beginning, the necessary technical infrastructure still might hamper situated capturing of models. Hence, this contribution aims to reduce the need for sophisticated technical components by enabling stakeholders to capture their paper-based models in a situation-sensitive way. We present a system that enables capturing paper-based models with mundane technical means by end-users under uncontrolled conditions. We describe the components developed for recognition of these models and embed it in a mixed-modality workflow supported by a tabletop interfacing a web platform for further processing. As our empirical evidence demonstrates, this approach enables both, situated and error-tolerant capturing of hand-drawn conceptual models by individual users. Moreover, it can be integrated with more sophisticated IT-based modeling tools for further digital processing.**

***Index Terms*— Conceptual model capturing, knowledge elicitation, collaborative modeling, recognition, representation.**

I. Introduction

The collaborative development of conceptual models [1] is a crucial activity in the area of concept mapping [2,3] or user-centered requirements engineering [4-6]. Both fields of research aim at creating a digital representation of these models to enable documentation and further processing [1,7]. Historically, collaborative conceptual modeling has been carried out with paper-based means [8]. Paper-based modeling provides the benefit of representing a model in the form initially conceived as a mental concept. Due to its efficacy in externalizing concepts, paper-based sketching is used in development practice for externalizing designs. Moreover, it allows the gradual development of more sophisticated forms of conceptual knowledge [9]. Paper-based models, however, need to be transformed to digital representations in a separate step. The process of manual transformation requires considerable effort to carry out and disrupts model processing, and thus should be avoided [10]. Research in the last years has proposed to create fully digital environments that support the collaborative modeling process from the very beginning [11-14]. Such approaches have been shown to have deficiencies with respect to the social dimension of the collaboration process during modeling [15,16], particularly in terms of equal access to contribute to the model and facilitating appropriation of the represented concepts [17] and in terms of mutual awareness of modelers' activities [1]. With the advent of interactive surfaces, a trend back to co-located direct collaboration based on a digital representation of the model can be recognized [18-21]. These approaches have solved the prevailing problems of digitally capturing a conceptual model while maintaining a social setup that appropriately supports the collaborative processes during modeling [15, 22-24]. The required infrastructure, however, is complex, expensive, and usually can only be re-located with high effort [25] [26]. The ability to bring the modeling environment to the modeler is especially important in organizational settings, where situated elicitation supports capturing a more rich set of information from domain experts [27]. Tablet computers are not an appropriate solution in this context, because they are single user devices and lack the embodiment of representation necessary to facilitate the collaboration dimension during modeling [16,28]. A variety of approaches has tried to tackle the challenge of capturing digital representations of manually created conceptual models via image recognition (e.g., [29-32]). They rely on paper-based modeling techniques and extract digital model representations from images taken of the paper based models. The problem to be solved here is a subclass of sketch recognition, which has been a subject of research as early as in the 1970s [33]. Since then, numerous approaches to improve recognition speed and reduce recognition errors have been proposed. The problem of sketch recognition thus can largely be considered as solved when performed on images captured under controlled conditions. Jiang et al. [29] describe a system to derive digital representations from sketched conceptual models and give an overview about the historic development of the field.

The current article thus does not set out to advance the state of the art in sketch recognition itself, but proposes a system that is embedded in an ensemble of instruments supporting collaborative modeling processes. Its novelty is in the integration of the described instruments to a complete, self-contained system. The system focuses on enabling users to capture their sketched models without constraining them to controlled capturing settings. Extracting features from images captured under uncontrolled conditions has hardly been a topic in the area of sketch recognition, but rather has only recently been addressed in face recognition [34-36] and optical character recognition (OCR) [37].

[1] Johannes Kepler University of Linz, Austria

[2] VoXel Interaction Design, Austria

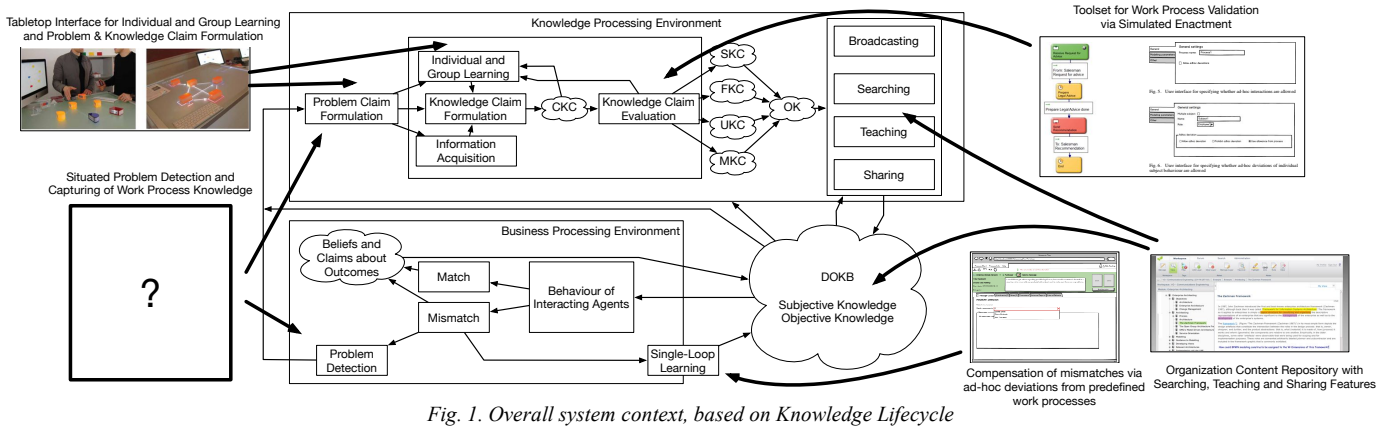

Fig. 1. Overall system context, based on Knowledge Lifecycle

The aim of this was to integrate paper-based modeling as seamlessly as possible in a modeling workflow comprising different computer-supported collaborative conceptual modeling tools. This is in line with current research in the field of requirements engineering (e.g., [17,38]). This research, however, does not address the recognition of models from pictures captured under uncontrolled conditions, but rather focuses on semantic interpretation of sketched models.

The present research is to be motivated from a design science perspective [39]. Our earlier research has focused on supporting collaborative modeling on interactive tangible tabletop surfaces (e.g., [22]). Experiences from evaluation in real-world settings have made evident that users demand to reduce the technical (in terms of required infrastructure) and cognitive effort (in terms of needing to learn the handling and constraints of a new tool) to contribute to collaborative modeling processes. From a practitioner's perspective, the contribution of the present research is thus an IT artifact that allows for situated capturing of sketched conceptual models by users themselves, which can be aligned easily with more sophisticated IT-based modeling tools. Of scientific originality thereby is the introduction of a recognition system that enables the extraction of paper-based model sketches from images captured under uncontrolled conditions.

The proposed capturing system sets out to relax the constraints on model extraction from images of paper-based models. It should tolerate shortcomings in both, image capturing (e.g., superficial content, skew) and the drawings themselves (e.g., bad pen quality, suboptimal drawing accuracy) in order to allow situated modeling and capturing by users [29]. At the same time, it must be easy to use and inexpensive to deploy in users' workplaces and should not rely on dedicated technology [40]. It should not be limited to specific modeling languages and should produce generic model representations that can be directly propagated to other IT tools for further processing [17].

The remainder of this paper is structured as follows: in the next section, we describe our context of use and derive the requirements to be fulfilled by a set of tools for the recognition of paper-based conceptual models. In Section 3, we present the current state of the art of model extraction from paper-based models and identify the contribution of the present work. Section 4 reports on the implementation of the current prototype of the toolset. Section 5 reviews the current performance and reveals the limitations of these tools from an application perspective. We proceed with a field study, in which we report on the practical use of the toolset in an organizational development setting. The paper closes with an account on further directions of research.

II. Requirements on Model Recognition for End User-Driven Conceptual Modeling in Knowledge-Intense Organizational Environments

The user-centric design of the envisioned system needs to take into account its context of use and the socio-technical system environment it is embedded in. We therefore briefly describe the overall system that has been designed to support collaborative modeling processes in organizational settings to facilitate reflection and learning processes. We then derive requirements on the system, taking into account the constraints imposed by the envisioned context of use.

A. Socio-technical System Context

The context of the system described here has been developed based on the *Knowledge Life-cycle* (KLC) [41], which describes learning processes in organizations. The KLC fundamentally distinguishes between a "business processing environment" (lower part of Figure 1), in which operative work is carried out, and a "knowledge processing environment" (upper part of Figure 1), in which new knowledge about organizational work is developed, assessed and distributed. Work in organizations is based on the "distributed organizational knowledge base" (DOKB, lower right corner of Figure 1), which comprises the codified knowledge of an organization as well as the subjective knowledge of its members. Whenever work leads to results that have not been expected based on the knowledge available in the DOKB, a mismatch is diagnosed that leads to compensation activities (lower center region of Figure 1). If a mismatch cannot be compensated ad hoc, the knowledge processing environment is entered (top left corner of Figure 1). In this process, a "problem claim" is explicitly formulated. Based on the "problem claim", "knowledge claims" —i.e., potential solutions to the problem—are developed. They undergo a validation process, which can lead to refusal of the knowledge claim or trigger the need for revision (upper center region of Figure 1). If a knowledge claim "survives" validation, it is distributed across the organization via different means of sharing (top right corner of Figure 1). In this way, it is integrated in the DOKB and becomes part of the foundation for future organizational work. Figure 1 also shows the main instruments that have been developed for supporting the implementation of

the KLC. Those instruments are briefly described in more detail below to enable the identification of requirements on the system described in the present article.

Our research has focused on support for activities in the knowledge processing environment so far, with tools supporting the collaborative development of a shared understanding about a work context [22], negotiation processes for setting up or revising work procedures [42], validation of proposed work processes [43], and sharing [44] and contextualizing [45] the developed artifacts in an organization. All of them support the operationalization of the knowledge processing environment depicted in the upper part of Figure 1. Further instruments have been developed to support compensation activities for mismatches occurring in the business processing environment depicted in the lower part of Figure 1 [43,46]. The remaining gap is to identify problem claims directly in a work situation. Bridging this gap would enable organizational members to articulate their view on the occurred mismatches. It could facilitate the transition from the business processing environment to the knowledge processing environment and allow starting knowledge production by providing context from the actual work situation.

The instruments developed for supporting the knowledge processing environment [18,47] have been shown to fulfill their design goals [22], but require dedicated technical infrastructure. This prevents their deployment in operative work settings as well as operation without support staff trained to solve technical issues. The design goals for the system to be developed for bridging the gap indicated in the lower left corner of Figure 1 can thus be formulated, based on the requirements from the intended context of use, as follows:

1. The system must not require technical infrastructure beyond mundane devices, such as digital cameras.
2. The system has to allow operation by non-technical staff and consequently must not require detailed technical knowledge for setup and use.
3. The system has to be technically compatible with the other instruments deployed in the overall ensemble in the KLC. In particular, the data representation for models has to be identical, and interfaces for interoperability have to be provided.
4. The system should enable users to produce identical outcome in terms of created models and modeling documentation as the other deployed modeling tools. In particular, it has to enable to build diagrammatical conceptual models with different types of nodes that can be linked via directed or undirected connections.

B. Requirements on Implementation

In this section, we describe how the abstract requirements described above should be operatively considered in the course of implementation, based on the existing set of tools.

The tabletop interface used for collaboratively articulating and reflecting on work-related knowledge has been implemented with optical recognition of passive modeling elements [18]. In order to be able to re-use the existing technology stack for model processing and in the light of the first requirement made above, the design decision has been made to use paper-based models as replacement for the tabletop system, which should be extracted by optical recognition from pictures captured by users. The elements representing model concepts should remain tangible for the positive effects on collaboration caused by concept embodiment [16,22]. Optical capturing should not require any specialized capturing equipment. Consequently, the system should accept digital photos taken by traditional digital cameras or smartphones. The photos should be provided to the system via a web interface. In addition, uploading via a smartphone app should be enabled.

Requirement 2 operationally needs the model recognition system to not make any assumptions about the qualities of the provided picture aside from the need that the model needs to be captured completely and the image resolution has to be sufficient to appropriately extract the model content. More specifically, the system has to be able to extract model information from images with superficial image content, arbitrary orientation and skew. The system needs to tolerate the fact that users take pictures of their models blindly (because they are holding the camera over their heads) or with significant skew (because they have to step back in order to capture the whole model). Pictures with distortions caused by wide-angular lens of traditional smartphones must also be accounted for.

Requirements 3 and 4 should be accounted for by using the data format specified for model exchange in the overall system environment, which has been adopted by all tools that have been involved so far. This also allows users to re-use already existing interfaces to external tools, such as the export to the concept mapping toolset CMapTools [48].

III. State of the Art in Recognition of Paper-Based Conceptual Models and Modeling

The challenges involved with creating digital representations of paper-based conceptual models were recognized as early as in the 1970s [33]. Operable solutions were enabled by increasing the processing power of computer systems and improved capturing devices (digital cameras), which were introduced in the 1990s (e.g., [49]). Still, limitations remain in existence from a user-centric perspective [50], as most proposed approaches for recognition of sketched conceptual models rely on controlled capturing settings or assume the pre-existence of a digital drawing (e.g., created on tablets).

In the following analysis of the state of the art, we focus on model extraction from a digital representation of a drawing and on capturing pen-and-paper based models for further digital processing. Particular focus is placed on the feasibility of operation by end users under uncontrolled conditions.

1) Interpretation of Sketched Models

Hammond and Davis [51] introduce a system for sketch recognition of UML-diagrams. It is not paper-based, but it supports input via tablet devices. Multi-stroke objects are recognized by their geometrical properties on several layers, and convey users' natural drawing, comparable to paper. The users receive visual feedback by the system while drawing. Sezgin and Davis [52] also consider sketching as an interactive process which allowed them to ground recognizing sketches on Hidden Markov Models. Their user study revealed that in certain domains people have preferred ways of drawing objects. They were able to support people through recognizing regularities by consistent ordering of strokes without restricting the users to sketch in a certain way.

Stapleton et al. [31] focus on bi-directional conversion of sketched and digitally created diagrams to facilitate an interactive, mixed-modal editing process. They illustrate their approach by applying it to Euler diagrams. Drawings are created with digital pen input. Their results show that a modeling process benefits from combinations of sketching and digital model manipulation, as both satisfy different user requirements in the course of model creation and manipulation. Similar results are presented in [38] for domain-specific modeling in software development.

The system proposed by Wüest et al. [17] pursues similar objectives as the present research. They focus on facilitating collaborative sketching for requirements elicitation. Their approach follows a distributed collaboration model using multiple tablet computers for sketching. In their study, they found that—while the evolution of a common understanding of the model semantics is facilitated—the system lacks awareness features to communicate what other users are doing. This is an issue introduced by the use single-user devices for model sketching. Co-located, paper-based settings as deployed in the system proposed here do not suffer from this limitation.

2) Digitizing Paper-Based Models

Hwang and Ullman [49] have proposed an approach for capturing sketched models as early as 1990. They have followed a domain-specific approach, focusing on recognizing CAD drawings. Due to the technical constraints of that time, capturing was assumed to happen under controlled settings, and focus was placed on algorithmic solutions for drawing recognition.

Jiang et al. [29] aim to meet objectives similar to our work, as they focus on the extraction of concept maps from paper-based drawings. In their work, they combine dynamic programming and graph partitioning. Their algorithm can extract node blocks and link blocks of a sketched concept map by recognizing the text content of each concept node, and generating a concept map structure by relating concepts and links. Their results still need to be validated for uncontrolled capturing settings.

Ghorbel et al. [32] present a system to create digital representations of handwritten architectural plans captured by end users with mobile devices. While their use case is not located in the area of conceptual modeling, they tackle the need for ex-post verification of recognition results by interactively involving users in the recognition process and let them intervene in case of structural or symbolic ambiguity of the model extracted from the captured image. As such, they were the first to explicitly address issues introduced by capturing under uncontrolled conditions. Their results from experimental validation show that it is feasible to provide meaningful recognition candidates for interactive verification. The approach currently lacks evidence of its applicability in real-world settings because empirical studies have yet to be performed.

3) Summary

Although current research already tackles the application scenario addressed in this article, several requirements according to the objective of our work have hardly been addressed explicitly and have not been examined in combination: (1) low cost or mundane devices are not explicitly addressed; (2) the level of expertise required for capturing is assumed to be available in most of the studies. Depending on the expected digital literacy of 'average' users, (3) tool chaining for embedding model capturing in larger frameworks such as the KLC has not been explicitly addressed, as it has not been a dedicated objective or research. While semantic, social, and pragmatic issues have been addressed with respect to user-centeredness, developers of stakeholder-oriented modeling support still need to integrate these findings, in order to ensure identical outcomes in terms of created models and modeling documentation when providing a chain of modeling and recognition tools. Technology-wise, related work does hardly address the challenges that come with capturing of model information under uncontrolled settings. Existing results [29,53], however, point at the difficulty of distinguishing nodes from connections in fully hand-drawn conceptual models. It thus can be useful for capturing under uncontrolled settings to adopt approaches used in augmented reality [54] and interactive tabletop interaction [15], where relevant elements are identified using clearly recognizable visual markers.

IV. Implementation

In this section, we report on the current implementation of the toolset based on the requirements identified above. We start with an overview about the workflow of model recognition and then describe the tool support for the different steps. The different modules of tool support are indicated in Figure 2. Figure 2 also gives an overview about the (intermediate) result sets produced by the modules, which are further elaborated on in Section D.

A. Overall Workflow

The process of model digitization already starts during modeling. The proposed toolset needs to deal with an arbitrary set of modeling languages elements, which are not constrained in shape or size. In order to meet this requirement, the elements bear unique visual markers to allow for reliable recognition.

Once the model is created, digitization starts with taking pictures of the model. Pictures do not need to be taken from a particular angle. It is possible to have several pictures, each showing only a part or details of the model. The pictures can be taken with a standard digital camera and subsequently uploaded to the recognition engine via a web platform. As an alternative, a smartphone app can be used to directly upload the pictures.

The recognition engine processes the pictures in a multistep procedure, which is described in more detail below. The first step is the identification of modeling the elements. Starting from the set of identified elements, the connections are identified in the next step. Finally, the recognition engine searches for labels of elements and connections, and extracts them for future reference. The result is displayed in a web platform and can be interactively refined there. This includes adding textual representations of the extracted labels. The final result is encoded as an XML file replicating the conceptual model structure and the visual layout.

Further processing can be performed with various tools. This includes model visualization, model editors, or post-processing software that interprets the results according to their conceptual or visual structure.

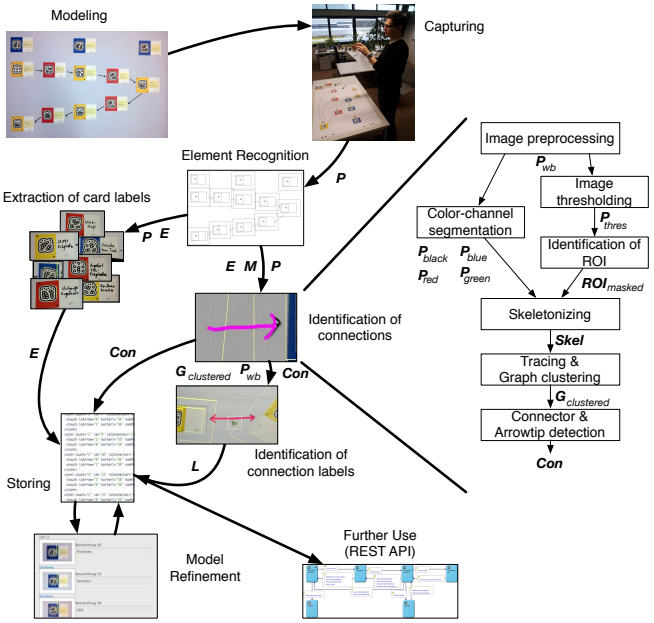

Fig. 2. Overall workflow for recognition of paper-based models (italic designators refer to (intermediate) result sets as described in section D)

B. Modeling

Following the requirements, conceptual models can be created in any form, with any notation, and with any number of elements. The recognition engine also does not constrain modeling to a particular modeling surface, as long as it provides a sufficiently light-toned background for the modeling elements and connections sketched among them. As neither shape nor color of the modeling elements is constrained, their identification relies on the availability of a visual marker. Currently, ReacTIVision [55] is used as a marker recognition engine for its stability, robustness, and appropriate data output. The physical size of the markers needs to be identical and known to the system in order to appropriately scale the extracted model. ReacTIVision was adapted to work with single images stored in a folder and output its result as an XML file. Drawing connections is supported between an arbitrary number of elements, i.e., connections can be forked to connect more than two elements. Intersecting connections of the same color consequently cannot be recognized—they need to be of different colors to be recognized reliably. Connections need to start and end nearby a modeling element to be recognized. Labels of elements need to be written in a dedicated area of the modeling card. Labels of connections have to be placed nearby somewhere along the connection line.

C. Capturing

Model extraction requires one or several pictures of the paper-based model. One design goal of the developed toolset was to put as little constraints as possible on the pictures used for model extraction in terms of camera parameters, perspective, lighting, and number of contained model elements. One obvious constraint is the resolution of the digital image, which needs to allow for both the identification of visual markers and tracing of the connections. When multiple images are taken, a single model overview picture is required in the current implementation of the recognition engine to identify the positions of the model parts captured in the more detailed images. Matching of the model parts is performed on basis of the identified model elements. Consequently, the current constraining factor of model size is that all model elements need to be recognizable in the overview picture.

Following the requirement of not assuming the availability of any special hardware infrastructure, pictures can be taken with any digital camera and provided to the recognition engine via a web-based upload interface. The future envisioned standard gateway, however, is a smartphone app, which would also allow interactively providing feedback about recognition results to the user. The current prototype of this app allows capturing pictures and directly providing them to the recognition engine without the need to manually upload them.

D. Model Extraction

Model extraction is carried out by a set of instruments that implement the steps outlined in Figure 2. The entire collection is referred to in the following as *MoDig engine* (MoDig is used as an abbreviation of "model digitizer" here). Extraction is described here for single-picture models. Algorithmic differences for multiple pictures are described further below.

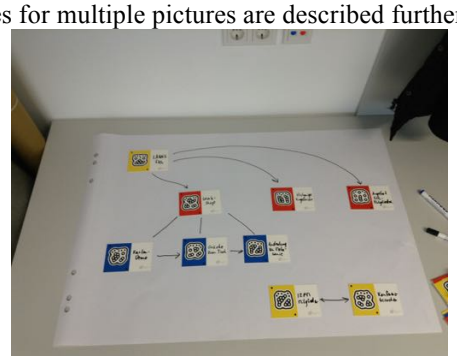

Fig. 3. Sample picture used for illustration of recognition steps

For illustration purposes, the model depicted in Figure 3 is used as an example in the following sections. It has been created to visualize the planned activities to be carried out within a workshop at a scientific conference. The picture has been taken with a smartphone camera by a user standing in front of the table, on which the flipchart used as a modeling surface was placed. Full-resolution images of Figures 3-9 are available as digital supplemental material to this article.

The picture comprises superficial parts in its border regions, most notably the office artifacts on the left- and right-hand sides as well as the power outlets at the upper border. In the following formalization, the pictures is described as a set of pixels $P = [p_1, \dots, p_n]$ with $p_i = (x_i, y_i, r_i, g_i, b_i)$, where (x_i, y_i) are the coordinates and (r_i, g_i, b_i) are the R/G/B values.

1) Element Recognition

Element recognition is performed by providing the source picture P to the ReacTIVision engine, which identifies markers and outputs their position and rotation. More formally, this step produces a set $M = [m_1, \dots, m_n]$, with $m_i = (id_i, x_i, y_i, rot_i, size_i)$ where id_i is the unique identifier of the marker, (x_i, y_i) indicates the coordinates of the center point of the marker in the 2D plane, rot_i is the rotation of the marker in *rad* related to the vertical axis of the picture, and $size_i$ is the length of the marker's diagonal (top left to bottom right).

Based on this information, the MoDig engine determines the size of the element by scanning for its border through

identifying the colored rectangular region around it. It produces a set $E = [e_1, \dots, e_i]$, with $e_i = (id_i, x1_i, y1_i, x2_i, y2_i)$.

The points $(x1_i, y1_i, x2_i, y2_i)$ determine the rectangular bounding box of the element. The extraction of card labels is based on this information. For each e_i, an rectangular image region is extracted. Its dimension is 1.5× the dimensions of the element's bounding box to provide the local context of the element. This supports later re-contextualization by the users. The image is rotated by $-rot_i$ and saved as a jpeg image element_{id_i}.jpg (cf. Figure 4).

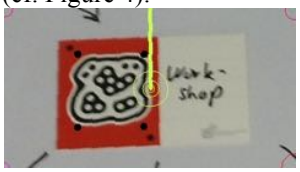

Fig. 4. Extracted model element

Analog to the ReacTIVision engine, the system can deploy a second marker extraction engine that detects QR Codes. The output is a similar set of markers that can be processed identically while enabling users to link to marker information via smartphone apps during modeling.

2) Identification of Connections

Overview: Identifying connections between elements requires processing the source image in multiple steps. In general, the approach taken here first identifies a polygonal region of interest $ROI = [p_{1_{roi}}, \dots, p_{n_{roi}}]$ with $p_{i_{roi}}$ being the pixels selected as the polygon's vertices. The ROI designates the area, in which connections can be expected. It then skeletonizes the content contained in the ROI and searches for graphs $G = [g_1, \dots, g_n]$, with $g_i = [s_{i_1}, \dots, s_{i_n}]$, where s_{i_a} designate the line segments identified by the tracing algorithm to belong to the graph. It finally examines the identified graphs in regard to the likelihood of being an intentionally drawn connection. The result is a set $Con = [con_1, \dots, con_n]$ with $con_i = (m_{i_{start}}, m_{i_{end}}, dir_i, g_i)$, where $m_{i_{start}}$ and $m_{i_{end}}$ are the markers of the start and end elements, dir_i contains information on whether the connection is directed, and g_i refers to the graph the connection consists of.

Image Preprocessing: The aim of this step is to make potentially relevant "foreground" pixels better distinguishable from "background" pixels. In addition, color channels should be clearly separable in order to reliably detect differently colored connections. In the preprocessing step, the image is white-balanced (following a white patch approach [56], as we assume that the modeling surface is light-toned) by identifying the dominant color tone of the brightest 10% pixels through averaging their RGB values, thus producing r_{white}, g_{white}, and b_{white}. The pixels of the white-balanced picture P_{wb} are then calculated as follows for compensating varying lighting conditions:

$$p_{i_{wb}} = \left(\frac{r_i}{r_{white}}, \frac{g_i}{g_{white}}, \frac{b_i}{b_{white}}\right)$$

The result of preprocessing is shown in Figure 5 (left) for the sample image.

Image thresholding and Identification of ROI: The next two steps identify the region of interest ROI, in which connections can potentially be expected. As the source image P not necessarily only contains the modeling surface, image parts with other contents need to be excluded from the ROI.

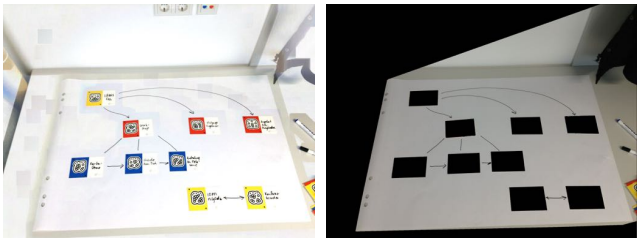

Fig. 5. Left: result of preprocessing, right: masked region of interest

The ROI is identified by applying an adaptive threshold (similar to Otsu's method [57]) to a grayscale version of P_{wb} and a morphological closing with a filter kernel matching the biggest marker size to remove artifacts, e.g., caused by sketches. In the resulting binary image P_{thres}, the convex hull of the biggest blob defines the $ROI = [p_{1_{roi}}, \dots, p_{n_{roi}}]$ with $p_{i_{roi}} = (x_{i_{roi}}, y_{i_{roi}})$, where the polygon is described by the segments connecting $p_{i_{roi}}$ with $p_{i+1_{roi}}$, where $i = [1, \dots, n[$. For further processing steps, the modeling elements are not part of the area of interest and can also be excluded. The region to be processed therefore is $ROI_{masked} = ROI \backslash E$. Figure 5 (right) shows an overlay of the ROI_{masked} with the original image. Black regions are excluded from further processing. In order to identify line drawing such as text and connections, the pixel data needs to be condensed into a higher-level abstraction, which is performed in the next steps.

Color-channel segmentation: The input to the skeletonizing algorithm is a binarized image representing pixels that potentially belong to a sketched structure ("foreground"). One requirement has been to allow for different pen colors to draw connections. To allow for color segmentation, we produce a normalized version of the picture $P_{normalized}$ by calculating

$$p_{i_{normalized}} = \left(\frac{r_{i_{wb}}}{r_{i_{wb}}+g_{i_{wb}}+b_{i_{wb}}}, \frac{g_{i_{wb}}}{r_{i_{wb}}+g_{i_{wb}}+b_{i_{wb}}}, \frac{b_{i_{wb}}}{r_{i_{wb}}+g_{i_{wb}}+b_{i_{wb}}}\right)$$

$P_{normalized}$ is transformed to an HSV representation, P_{HSV}, which is further brightened by maximizing the V values for each pixel. Black strokes are identified in $P_{normalized}$ by thresholding it based on intensity to identify dark areas and producing a set of binary pixels P_{black}. For color pens, P_{HSV} is thresholded with a heuristically determined pen-specific hue range identifying (bright) blue, green or red pixels, thus producing P_{red}, P_{blue}, and P_{green}, respectively. The four binary images act as an input to tracing algorithm one by one.

Skeletonizing: Identification is based on the existence of dark or highly saturated colored pixels in the preprocessed image P_{bw}, which have been extracted during color-channel segmentation. Figure 6 (left) shows an overlay of P_{black}, P_{red}, P_{blue}, and P_{green} masked with ROI_{masked} for the upper right quadrant of the sample picture (white regions are excluded from skeletonizing). Preprocessing still leaves many artifacts in this quadrant due to the black cloth visible in Figure 3.

The skeletonizing algorithm takes each of the binary images as an input and extracts connected regions of foreground pixels ("blobs") $B = [b_1, \dots, b_n]$ with $b_i = [p_1, \dots, p_n]$ where each p_i is a binary pixel. To extract structure information, each b_i is skeletonized to a single-pixel width line $skel_i$ via the approach proposed by Saeed et al. [58] and is then added to the set $Skel$.

Tracing and graph clustering: The pixels in each $skel_i$ are classified as line end points (points 1, 7, 10 in Figure 7), crossing points (point 6 in Figure 7) or interior points depending

on the foreground pixels in a 3×3 pixel neighborhood. The tracing algorithm creates a graph representation g_i consisting of line segments s_{i_a} by recursively traversing $Skel$ and following each $skel_i$ from an end/crossing point to the next end/crossing point. Each of these parts in a $skel_i$ constitutes a s_{i_a}. In Figure 7, a graph g_1 is shown consisting of s_{1_1}, which is located between point 1 and 6, s_{1_2}, which is located between points 6 and 7; and s_{1_3}, located between points 6 an 10. In addition to the structural information, each s_{i_a} comprises information on length, and an array of support points inside the segment (points 2-5, 8-9 in Figure 7), including thickness data of the line stroke in the non-skeletonized picture at these points (based on a distance transform of the binary picture [59]). All g_i are added to the set G_{raw} including information on their color based on the binary image they were extracted from.

A clustering step is applied on G_{raw} that aims to merge disconnected regions that are assumed to belong together. For each end point $p_{i_{end}}$ in a graph g_i, we search for graphs g_j with a point p_j being closer than a minimum distance. This allows us to join disconnected sketched lines and merge arrow tips that do not touch the line itself. We have heuristically determined a limit to 0.2 times marker height as providing good results. The resulting merged graphs are added to the set $G_{clustered}$.

Fig. 6. Upper right quadrant of sample picture (left: input to skeletonizer, right: output of tracer)

Figure 6 (right) shows the graph set generated by the tracer for the upper-right quadrant of the sample picture (graphs thickened for better visibility). The graphs on the right have been created due to the misrepresented area of interest.

Connector and arrow tip detection: $G_{clustered}$ is used to identify the connection candidates: As a first test, graphs with an average thickness higher than a threshold are rejected (e.g., eliminating the artifacts in the top right image corner as visible in Figure 6, left). A number of rules are then applied to identify actual connections in $G_{clustered}$ that link two modeling elements e_i and e_j. The rules used for connection identification take several factors into account. Most notably, connections are defined to end nearby modeling elements. Graphs are thus identified as connections if there are graph points outside a nearby region (suppressing local scribbles) and if it is nearby at least two modeling elements (suppressing loops). The threshold for "nearby" is implemented as a fixed ratio of the element size as extracted from *P*. In Figure 7, points 1, 2, and 5-10 are rated to be nearby an element. Points 1 and 6 are considered to be end points of a connection as they are located more closely to the border of the modeling elements than their neighbors. Successfully tested graphs are added to the connector set $Con = (con_1, \dots, con_n)$ and removed from the set $G_{clustered}$.

The algorithm also recognizes connections with multiple endpoints by following all line segments at intersections or forking points. Consequently, intersecting connections of the same color cannot be recognized, as they would be considered a single multiple endpoint connection. Intersecting connections thus have to be of different colors to be recognized correctly.

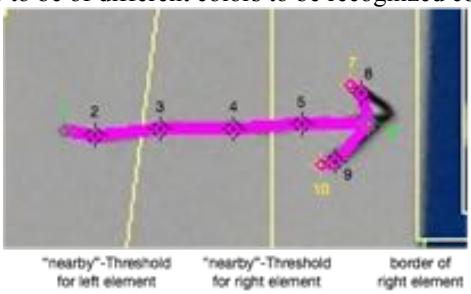

Fig. 7. Detection of connections from identified segments

Each identified connection is additionally scanned for nearby line segments that could constitute an arrow tip, indicating directed connections (cf. points 7-8 and 9-10 in Figure 7). Here, rules take into account the length of potential line segments and the angle enclosed between them and the final line segment. This information is stored in dir_i and is added to con_i.

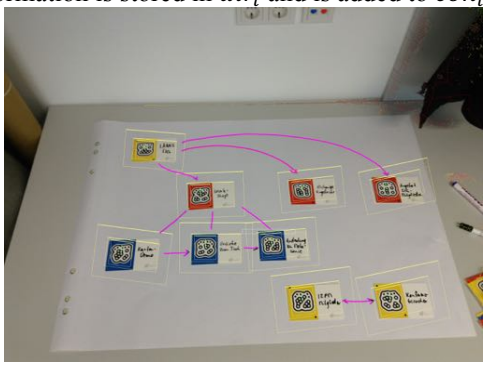

Fig. 8. Overall recognition result

Graphs which do not start nearby a modeling element are not considered further during connection search. This creates a more robust behavior in case of the pictures containing content not belonging to the actual model. As an example, the pen located at the left border of the sample picture is contained in the *ROI*. Still, as none of the resulting graphs are located nearby an element, the resulting graphs were not considered further.

The final result of connection identification is shown in Figure 8. All connections have been successfully recognized, and the picture also shows no false-positive connections. The arrow tips also have been recognized correctly.

3) Identifying Connection Labels

Connection labels are extracted from $G_{clustered} \backslash Con$ after connector identification. For each graph in this set, a bounding box is computed and expanded by 80%. Graphs with dimensions over a heuristically determined threshold (half a modeling element height) are disposed. For the remaining graphs, all graphs in set Con are examined if at least one support point is located inside the bounding box. If so, a new candidate label $lcand_i$ is added to set L_{raw} (cf. Figure 9, label "4" recognized, label "not relevant" ignored by label recognition).

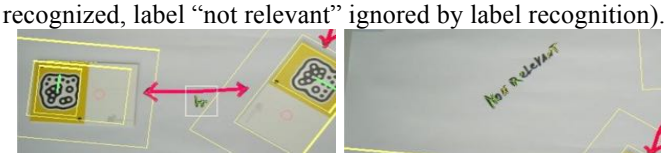

Fig. 9. Detection of connection labels (left: recognized label, right: ignored graphs)

As labels can consist of several individual letters, represented by individual graphs, a merging stage iterates over set L_{raw} and the remaining graphs from $G_{clustered} \backslash Con$. Once a label candidate $lcand_i$ and a second graph have intersecting bounding boxes, the latter is merged into $lcand_i$. Label recognition thus operates incrementally in the surrounding region of a connection, i.e., it only requires the label to start in the region. As no letter recognition is currently performed, the identified labels are extracted as images. For each label, a rectangular bounding box is defined comprising all graphs identified to belong to the label, and the respective part of the picture is saved as an image named label_{i}.jpg for later reference, where i is a counter incremented for each label.

4) Multi-picture Extraction

When multiple pictures are used, the algorithm relies on the availability of a single overview picture. This overview picture, however, does not need to be explicitly marked as such. Rather, the picture with the highest number of identified elements is used as the overview picture. Model recognition is performed on each picture separately as described above. No pixel-based or conceptual stitching is performed in the current implementation. Consequently, no connections or labels that are spread over two pictures can be recognized. Added value, however, currently is generated for label extraction, where the label with the highest resolution is included in the final result.

More sophisticated multi-picture recognition is currently being implemented. By identifying overlapping areas of model parts in the extraction results of the single pictures, the constraint of having a single overview picture can be modified to only require overlapping areas between single pictures, thus allowing for arbitrarily sized models.

5) Model Refinement

Model refinement allows improving interactively the recognition results by manually providing further information about the model to be recognized. In its current implementation, this step is supported by features of a web-based platform. They allow replacing the extracted pixel-based labels with their textural representations (cf. Figure 10).

The next iteration of the toolset will offer refinement support via a mobile interface, which allows providing feedback about the recognition results already during capturing, enabling immediate user reaction on results with insufficient quality. Current prototypes of the recognition engine already allow for live skeletonizing of a video input stream, which enables directly assessing the recognition quality to be expected during capturing. However, results are too preliminary here to elaborate on this improvement of tool handling.

Fig. 10. Interactive refinement of model information

E. Integration with Overall Set of Instruments

The results of model recognition are stored as XML files using the same data format as the other modeling tools in the overall KLC-based system (cf. Section II). They can thus be directly used for import to the interactive tabletop modeling system. By making use of the functionality provided there, models can be stored in other file formats to be compatible for legacy tools, like the concept mapping tool CMapTools [48].

The XML output focuses on representing the conceptual structure of the model and largely omits its layout. In particular, connections are only represented by their endpoints, and lack information on the connection path. For use cases, where exact presentation of the original layout is required, an SVG (scalable vector graphics) output is generated to provide a model version that exactly resembles the original layout of the paper-based model, including the identified connection graphs. In combination with the extracted modeling elements and connection labels, this information can be used to digitally re-create the original model in both layout and logical structure. This representation, for example, is used for display and exploration of models in the content repository used to share organizational content in the KLC.

Recognized models are provided to the other tools via a REST API (Representational State-Transfer) and thus rely on standard technologies (HTTP-requests) for interoperability. The API exposes the model in different formats (XML-based model exchange format, SVG, unprocessed overview picture) as well as the extracted model parts (modeling elements, connection labels). It furthermore provides access to processing metadata in the form of an XML-based log-record. The API provides read-only access to the data, as the MoDig-engine acts as a provider for modeling information to other tools. For integration with external providers of pictures, such as the mobile capturing tool (cf. Section II), an interface to push new pictures to the engine via a HTTP-POST request is provided.

V. Validation

The aim of the present research is to enable users to capture paper-based sketched conceptual models under uncontrolled conditions and extract digital model representations that can directly be used further in IT-based conceptual modeling tools. For this aim to be achieved, the system has to provide a level of recognition accuracy that requires minimal effort by users for continuing to work with the digitized version of the model. At the same time, users should be free from technical constraints on capturing to enable them to focus on their actual tasks. The validation of the proposed approach is thus carried out on two levels. First, we assess the recognition quality of the developed system in terms of accuracy of the identified model information. This assessment has been carried out with a set of sample pictures provided by potential users of the system. This part of the evaluation is described in the next subsection.

The second subsection provides insights in a field study, during which the system was operatively deployed in an organizational development workshop in a hospital. We report on the results of the workshop, how these were embedded in the operative IT systems and give an account on the users' experiences and expectations when using the system.

A. Evaluation of Recognition Quality

The aim of this article is to show the feasibility of a recognition engine for paper-based conceptual models captured

by users under uncontrolled conditions. In order to allow for seamless integration of the developed tool in the mixed modality toolchain, the amount of recognition errors, which would require manual intervention for correction, need to be as low as possible. Recognition quality alone, however, does not determine the potential value of the tool in daily use. The actual effort for transformation is a better metric for that. The less effort (in terms of time necessary for transformation) is required, the higher the potential value of the tool for users.

The baseline for recognition quality and transformation effort is the manual transformation of the paper-based models to digital models by a human actor. Recognition speed is not critical, as the use case does not comprise synchronous operation of paper-based and digital tools.

1) Methodology

Recognition quality is assessed quantitatively by calculating the recognition accuracy of MoDig engine in relation to human transformation of the models as a baseline.

Each model is analyzed by a human actor regarding the amount of model elements, connections, and arrow tips it contains (these categories are referred to as items in the following discussion). The distinction is made as those types are algorithmically treated differently during recognition. Connection labels were also considered for analysis, but were later removed, as they were hardly being used in the available sample models. The quality of their recognition is discussed qualitatively in the report on the field study in the next section. For all item types, accuracy is calculated by calculating the sum of missing (i.e., false negative) and superficial (i.e., false positive) recognized items and dividing this value through the amount of items identified by the human actor.

The interpretation of results also needs to take into account the *effort necessary to recognize and correct errors in the extracted model*. The effort is quantified by the amount of time necessary to compare the extracted model to the original picture(s), search for errors, and correct them. The baseline for comparison is manual transformation of the model. For each extracted model, the time needed for manual transformation, as well as the time needed for finding and correcting errors in the automatically extracted model, are measured. Aside from this, the remaining errors are counted for both, manual transformation and correction of automatic extraction. In order to minimize the effects introduced by the performance of individual users, the transformation and error correction is carried out on the same set of pictures by several users independently. The system is considered to provide the intended value, if the time needed for correcting recognition errors is significantly lower than the effort required for manual transformation, while the remaining errors must not significantly exceed those introduced in manual transformation.

2) Data Collection

Data for the assessment of *recognition quality* has been collected over a duration of 3 months in field studies. Practitioners in organizational development have been asked to provide real-world examples of how they would use the system in their daily practice. The practitioners were recruited from the special interest group of the research project the present work was embedded in. Consequently, all of these practitioners had prior experiences with the interactive tabletop modeling system, and thus were knowledgeable in the general modeling approach. None of them had experiences with, or prior knowledge in, IT-based image recognition and its technical requirements and constraints. They were only instructed to take pictures with their commonly used camera (or smartphone). Of the provided images, only those in which all items were also readable and interpretable for humans were used, as human interpretability was used as a baseline. Furthermore, images that violate the present fundamental recognition constraints of the modeling engine (no overlapping modeling elements, no intersecting connections of the same color) have been removed. Overall, 43 models overall comprising 936 items (338 modeling elements, 320 connections, 278 arrow tips) created collaboratively by 73 different people and photographed by 12 different people were considered for evaluation of recognition quality. The median number of modeling elements per model amounted to 8 (min=3, max=16); for connections the median was 8 (min=0, max=15), and for arrow-tips the median was 7 (min=0, max=14).

Data for the assessment of *effort for transformation* have been collected in a quasi-experimental setup. Fourteen users (3 females, 11 males, with age ranging between 22 and 42 years) with a mixed amount of digital conceptual modeling experiences (4 persons with several years of experience, 7 persons with limited experiences, and 3 persons without any conceptual modeling experiences) were recruited as participants. Three models were selected from those collected in the field studies based on the archived recognition quality: one of the models with no recognition errors, the model with the highest overall error rate, and a model with an of overall error rate near the average number of errors. As an additional selection criterion, the selected models had to contain at least 5 modeling elements, 5 connections and 4 arrow tips (each of this numbers designating the 25% quartile for each set of items). For each of these models, a digital representation in a computer-based conceptual drawing tool (http://draw.io) was created based on the recognition results. Those models consequently contained only elements recognized by the MoDig-engine.

For completing their tasks, the users were provided with printouts of the source pictures. They were asked to make sure that the digital representation resembled the model contained in the picture in terms of conceptual structure (i.e., modeling elements, connections, and arrow tips) as well as in layout. Textual transformations of captions on modeling elements or connections were omitted. If discrepancies were found, the users were instructed to correct them in the editor. The time required to complete this task was measured. Afterwards, users were asked to manually create a digital representation of the model from scratch using the same tool. The time required to complete this task again was measured. Upon completion, the remaining errors in the manually created digital representation as well as the automatically extracted and subsequently corrected digital representation were counted.

3) Results

The metrics for *recognition quality* are summarized in Table 1. It shows the minimal, maximum and median accuracy for each item type and for the sum of all items. In addition, the mean values and the standard deviations have been calculated to give an idea about the overall distribution.

Table 1. Accuracy for different item types

Accurracy	Elements	Connections	Arrowtips	Overall
Min	91,7%	7,7%	25,0%	48,6%
Median	100,0%	100,0%	85,7%	95,2%
Max	100,0%	100,0%	100,0%	100,0%
Average	99,8%	88,8%	82,9%	90,3%
StdDev	1,3%	21,5%	20,8%	13,7%

The average accuracy of 90.3% in an averagely sized model [based on the collected data, models overall on average contained around 22 items (modeling elements, connections and arrow tips)] amounts to approximately 2 errors per model. For interpretation, it is necessary to recognize that errors do not occur independently of each other. If a connection is not recognized, its respective arrow tips also cannot be recognized, causing follow-up errors. The same is true for recognition errors of modeling elements, which cause all attached connections and their arrow tips to be missed.

The low minimum accuracy for connections (7.7%) is attributed to a model which has shown a high number of false-positive recognition results for this item type. This was caused by connecting modeling elements that were positioned so closely to each other, so that their "nearby" region overlapped. The connections were still recognized appropriately. However, users also added labels to the connections, which consequently also were located in both nearby regions at the same time. The graphs of the labels were thus identified as connections by the recognition algorithm, and, in this way, nearly doubled the number of recognized connections. The same model has caused the minimum accuracy for arrow tips for the same reasons.

Figure 12 gives a graphical visualization of the observed overall error rates as well as for each item type. The x-axis is segmented in discrete categories for the error rate in steps of 5%. The y-axis shows the number of models that have an error rate in the respective interval.

As can be seen in Figure 11, 17 out of 43 models have been recognized without any error (40%). Thirty-three models (77%) have been recognized with an accuracy of 90% or higher. The accuracy does not drop below 65% for 88% of the models.

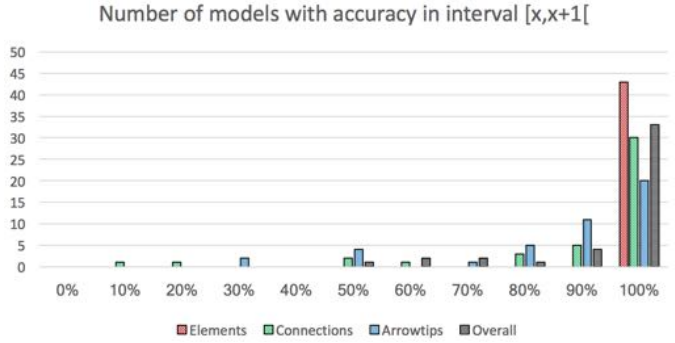

Fig. 11. Distribution of accuracy for different item types

A separation by item categories shows a more differentiated picture. For modeling elements, 98% of the models have been recognized without any error, and no model drops below an accuracy of 90%. In absolute numbers, this is caused by a single recognition error in a single model out of 43. The approach of using optical markers to reliably identify modeling elements can thus be considered successful. For connections, 30 models (70%) have not shown any recognition errors. Accuracy above 80% (i.e., 1-2 recognition errors in averagely sized models) is achieved for 81% of the models. The amount of models without any recognition errors for arrow tips is lower than in the other categories (18 models, 42%). This, however, is mainly caused by follow-up errors of unrecognized connections, which lead to one or multiple unrecognized arrow tips. Forty models (86%) show an accuracy of more than 60%. When compensating for the calculations for follow-up recognition errors, the accuracy for arrow tips never drops below 75% (amounting to 2 recognition errors in absolute numbers), with 26 models (60%) showing no recognition errors.

These numbers show that—while recognition quality of modeling elements hardly leads to any error—the error rate for connections and arrow tips still require manual checking and correction by the user. The system can only be considered successful, if the compensation of recognition errors causes significantly less *effort for transformation* than manually transforming the models. In the following, we present the results of the quasi-experiment that has been carried out for assessing this effort as described above.

Table 2 and the boxplots in Figure 12 show the descriptive parameters for the distributions of task completion time in seconds for each task (x_corr denotes the correction task, x_self denotes the manual redrawing task for each model). The wide range of required completion time can be attributed to the different amounts of effort invested by the participants to resemble the model layout as closely as possible and their different levels of experience in computer use. None of the models contained any conceptual errors after completion of the tasks. Layout resemblance varied to a large degree in the manually created models, but no severe layout deviations were recognized. Further analysis thus focuses on task completion time as a metric for transformation effort.

Learning effects that resulted in more efficient use of the drawing tool were identified as a confounding variable. Its effect was minimized by confronting users with an up-front training task to get used to the tool, and the fact that only task completion times of successive tasks were compared pairwise, and consequently were subject to similar learning effects.

Table 2. Task completion time for error correction and manual transformation

	N	Minimum	Maximum	Mean	Std. Deviation
m1_corr	14	10.0	531.0	83.714	133.4117
m1_self	14	75.0	638.0	198.214	161.3053
m2_corr	14	27.0	271.0	87.714	79.7799
m2_self	14	53.0	295.0	148.714	77.8415
m3_corr	14	149.0	346.0	223.786	62.4465
m3_self	14	181.0	558.0	323.643	119.0460
Valid N (listwise)	14				

For each pair of variables (x_corr and x_self), it has been tested whether x_corr was significantly lower ($p<0.05$) than x_self. For model 3 (that with the highest error rate), the completion times for both tasks are normally distributed (tested with Kolmogorov-Smirnov test for one sample). This is not the case for the models ($p<0.05$). Consequently, a paired samples one-sided t test was performed and showed, that the correction of model 3 took significantly less time than manual transformation ($p<0.05$).

For model 1 (without any recognition errors) and model 2 (with an average number of recognition errors), significance was tested with a one-sided Wilcoxon test for paired samples. Both tests showed that correction for each of the models took significantly less time than manual transformation ($p<0.05$).

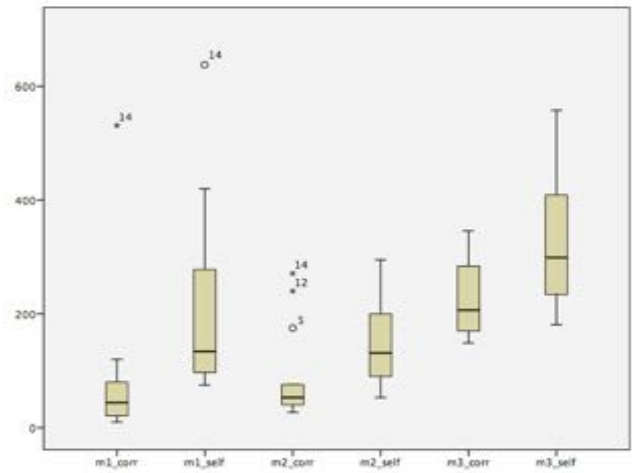

Fig. 12. Distribution of task completion times in seconds

In addition, qualitative feedback was collected from the users performing the transformation. Here, we summarize the statements that were given by at least 3 users. Independently of the time required for transformation, several users stated that they found the correction of the recognized model cognitively less demanding than drawing it from scratch, as the already available elements provided anchor points for easier orientation. Users also stated that they considered it infeasible to manually create models that resemble the original layout as closely as the recognized model has done. As a suggestion, several users stated that they found identification of false negatives (i.e., missing items) less demanding than the identification of false positives (i.e., superficial items). They consequently prefer conservative recognition that rather produces false negatives (i.e., drops actual items) than identifying false positives (i.e., adding superficial items), as the former can be recognized more easily. Error correction, however, has appeared to be faster in the case of false positives (i.e. deleting superficial items is faster than adding missing ones). Thus, identifying the optimal thresholds for item recognition does not appear to be trivial and might even be depended on by the users who are operating the system. As this aspect only has been addressed in qualitative statements so far, it requires further examination in future research.

B. Field Study

The field study has been performed in the healthcare sector involving in-house organizational developers. They support clinics in reflecting and re-designing work procedures. Development projects are collaboratively designed with each clinic and subsequently managed by the organizational developers. The descriptive investigation sought to confirm and extend the previously performed Value Network driven design of work processes (cf. [60]) by semi-automatically capturing evolving structures of work processes. Even though the created models have no formal meaning, they provide a key memory for documenting and sharing work knowledge, both in terms of results and process history (cf. [61]). For this reason, the models were not only captured in their final version, but also in intermediate steps that were considered critical to understand the process of model development.

The model elements were used to visualize relevant concepts within the work process. The terms used to denote concepts referred to roles (e.g., physician), tasks (e.g., treatment), and/or documents/data (e.g., a patient record). The process designs were based on named connections drawn between the modeling elements representing the concepts (e.g., indicating a sequence of steps). In order to overcome experienced shortcomings when capturing medical processes, in particular with respect to critical procedures [62], the meaning of encoded card items and relations has been assigned by the participants.

For modeling, the participants were provided with white flipchart paper, the modeling cards including the codes required for recognition, and appropriate pens. They were instructed on how to create a model and were asked to use different pen colors, connection shapes, element positions, and patterns. Then, 3 organizational developers began to collaboratively design a process, namely how a patient should be handled in a clinic.

In doing so, they created two models sketching two variants of the relevant tasks and their accomplishment (cf. Figure 13). The participants used red, blue and black pens, as well as various styles in writing and denoting items, in particular, connections. Modeling elements encoded primarily task-specific activities and roles, whereas connections encoded directed relations without label (explicit meaning), sequence numbers, and context information that were considered relevant for handling incoming patients.

Each model was pictured using a mobile phone camera, and processed by the MoDig engine before being displayed for further editing in an external tool (CMapTools; cf. [48]). The recognized elements were compared to the originally created maps, as the purpose of the descriptive investigation was to test the quality and effective use of the results for the participants.

1) Recognition Result

The result of model recognition for the two produced is shown in Figure 13. The left image is referred to as “Plan A” in the following discussion, whereas the right image is labeled “Plan B”. The quantitative metrics given in the following refer to both pictures. All 30 modeling elements have been recognized correctly (accuracy: 100%). Of the 25 connections, 2 have not been recognized, and no false-positive connections were identified (accuracy: 92%). The connections labeled with “6” in Plan A and “8” in Plan B have been omitted, as one of their ends was placed slightly outside the nearby region of the modeling item it should have been attached to. The heuristics used to identify nearby connections have been improved since the study was conducted and do not lead to this recognition error anymore in the current implementation.

Each connection was assigned a direction, i.e., sketched with a single arrow tip. Accordingly, 25 arrow-tips should have been identified. As arrow tip recognition is dependent on the recognition of the according connection, the arrow tips of the two missing connections were not recognized. All other arrow tips have been identified correctly, and no false-positive arrow tips were identified (accuracy: 92%).

The recognition of connection labels shows a weaker performance. For Plan A, 13 labels have been added to the model (several numbers, used redundantly, and two explanatory text labels). The recognition engine could not identify any of these labels. For Plan B, 13 labels have been added to the model (numbers 1-11, and two explanatory text labels). The recognition engine was able to correctly identify 5 labels (“1”,

"2", "3", "4", "9"). The other labels in Plan B have not been recognized. This amounts to an accuracy of 19%.

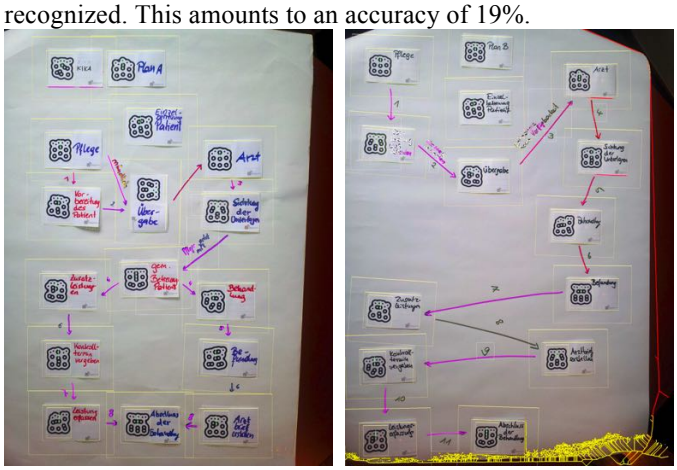

Fig. 13. Recognition results for models created in the field study (full-resolution images available as digital supplemental material to this article)

Two different reasons for this poor performance could be identified in ex-post analysis. In Plan B, the labels 5-9 and 10-11 as well as the explanatory label of connection 3 were placed slightly outside of the regions considered to contain potential labels for the according connections. The heuristics used to identify nearby labels have been improved since the study was conducted and do not lead to this recognition error anymore. The missing recognition of the explanatory label in Plan B and all of the missing labels in Plan A are caused by the assumption of the interpretation algorithm that labels will be placed outside the "nearby" region of modeling elements. While this assumption seemed reasonable based on the available testing data, the field study has shown the inappropriateness of this assumption. When modeling items are positioned nearby (as is the case in Plan A), labels are generally located within one of the concerned "nearby" regions, preventing their recognition. While this has been corrected in the current implementation, it did have implications in the field study and thus is reported here accordingly.

2) Further Use of Workshop Results

The recognized models were loaded on a Comprehand Table in a subsequent session for further reflection. The recognition errors were corrected by the users themselves directly on the table. The labels, which are attached to modeling elements and connections as images directly after import, were replaced by textual representations. Content-wise, no significant changes were made to the models aside from minor rearrangements affecting the spatial layout of the models.

The organizational developers managed all of the models they were working on within in a repository provided by a CMapTools [48] server. Consequently, the resulting model was exported to the legacy file format used to store and exchange models in CMapTools. From this repository, the models were also available for the operatively affected people in the clinic, which they used as a point of reference for their work processes, to assess which variant worked better in daily practice. In a subsequent workshop a few weeks later, the models were again opened on the Comprehand Table for reflection on the variants.

The integration with the locally used ensemble of tools and the contribution to the overall workflow worked as intended in the field study. Still, the organizational developers refrained from permanently deploying the card-based modeling tool, as their workshop setup required immediate availability of digital model versions and did not allow for manual correction of recognition errors. They consequently chose to remain with the workshop setup based on the interactive tabletop modeling interface, although they recognized that the required organizational and technical effort was much higher than for the card-based system.

3) Qualitative Feedback of Users

In a final step, the users participating in the field study provided qualitative feedback on the use of the system in a focus group. The following critical factors for permanent deployability of the MoDig tool were identified in this setting. First, the system as a whole was rated as easy to use. The required workflow was perceived to be understandable, but still considered rather inconvenient and cumbersome to use. This was mainly attributed to the fact that taking pictures and uploading them to the recognition systems was not integrated in a single step (i.e., in one app supporting the whole workflow). The participants were willing to accept some trade-off due the 'low-cost approach', as they termed it, but still expected a more seamless experience during capturing and more immediate feedback on the archived recognition quality.

Second, the participants felt disturbed by the cognitive effort to be spent on positioning elements in the course of drawing, in order to avoid recognition problems due to the limitation of the deployed version of the recognition engine. In particular, they criticized the required accuracy in ending drawn connections nearby an element. The deployed version showed problems in recognizing connections that ended slightly too far away from elements. In combination with cards that slipped out of their initial position during modeling this caused interruptions in the modeling flow that were considered to be hardly acceptable in everyday use.

Despite these shortcomings, the participants felt the systems could reduce the initial effort required for digitizing the model. However, they expected a more reliable recognition of modeling elements and connections in order to focus on content entries, either using the editor of the system or any other third-party modeling tool after exporting the data.

VI. Conclusion

In the present paper, we have introduced a system to support the user-driven capturing and automated recognition of paper-based conceptual models. The aim was to develop a system that can be operated by end users with commonplace capturing devices such as smartphone cameras. The model recognition engine has been designed to process such pictures and provide the results in a format that is interoperable with other tools in the overall system environment, which supports organizational learning processes. The system has been evaluated regarding its recognition quality and the required effort to compensate for potential recognition errors. These results show that the system has achieved its fundamental aims, while still being negatively affected by the technical limitations of the current version. Both, capabilities and limitations also could be confirmed in the field study. Consequently, the contribution of the present work is the introduction of a recognition system for conceptual drawings, which was explicitly designed to be operated with

uncontrolled input created by users without technical background or knowledge. In comparison to existing related research, the proposed system advances the state of the art by introducing an approach for extracting conceptual model information from paper-based sketches captured under uncontrolled conditions in a fully integrated, directly deployable system. Related work so far has always assumed the availability of a digital representation of the sketch (e.g., created on tablet computers) [17] or digitization to be performed under controlled conditions [29]. Attempts for tackling the challenge of uncontrolled capturing so far have only been tested in experimental settings [32]. Our approach has shown appropriate recognition performance in a quasi-experimental evaluation, and has demonstrated its feasibility for real-world use in a field study.

The research presented here has several limitations. First, in terms of tools performance, the quality of recognition especially for connections between model elements is insufficient, especially when the source pictures suffer from suboptimal conditions (mainly in terms of shadows caused by lighting and poor quality of the pens used for drawing connections). This hampers unattended deployment for end users, who are not necessarily aware of the current limitations. Second, the integration in the overall set of tools is still limited from a user's perspective, which limits acceptance and usability for novel users. Third, from a methodological point of view, the system requires a more extensive evaluation to examine the recognition workflow from both user and technical perspectives, to develop informed design hypotheses to further support measures and improvements of recognition quality.

These limitations will be addressed in future research and development. We plan to improve recognition quality by using more robust connection tracing algorithms as identified in Section III. Furthermore, the heuristically determined parameters for the recognition modules (in ROI detection, graph tracing, and recognition of arrow tips and labels) will be systematically assessed based on an extended database of sample pictures to improve overall recognition accuracy. The capabilities of model extraction from multiple pictures will be developed further to avoid the need for a single overview picture, which currently limits the maximum model size. Finally, integration in the overall set of tools will be made more transparent by the provision of a mobile app as a gateway for capturing, recognition and model refinement. This gateway is envisioned to interactively guide the capturing process in a later revision, intertwining it with model recognition to provide immediate feedback on the recognized result and allow for in-situ revision of the captured models, as outlined by Ghorbel et al. [32]. Interactive capturing will also allow implementing features for tracing design history, which was suggested to be supportive for end user-driven design by Mangano et al. [63].

REFERENCES

[1] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo, "How do practitioners use conceptual modeling in practice?," *Data & Knowledge Engineering*, vol. 58, no. 3, pp. 358–380, Sep. 2006.

[2] N. Stoyanova and P. Kommers, "Concept mapping as a medium of shared cognition in computer-supported collaborative problem solving," *Journal of Interactive Learning Research*, 2002.

[3] H. Gao, E. Shen, S. Losh, and J. Turner, "A Review of Studies on Collaborative Concept Mapping: What Have We Learned About the Technique and What Is Next?," *Journal of Interactive Learning Research*, vol. 18, no. 4, pp. 479–492, 2007.

[4] J. Barjis, G. L. Kolfschoten, and A. Verbraeck, "Collaborative Enterprise Modeling," in *Advances in Enterprise Engineering II*, vol. 28, F. Harmsen, J. L. G. Dietz, W. van der Aalst, M. Rosemann, M. J. Shaw, and C. Szyperski, Eds. Springer Berlin Heidelberg, 2009, pp. 50–62.

[5] G. P. Mullery, "CORE-a method for controlled requirement specification," Proc. of ICSE '79, pp. 126–135, 1979.

[6] P. Rittgen, "Collaborative modeling of business processes: a comparative case study," *Proceedings of ACM SAC 2009,* pp. 225–230, 2009.

[7] J. D. Novak, "Concept mapping to facilitate teaching and learning," *Prospects*, vol. 25, no. 1, pp. 79–86, 1995.

[8] H. D. Dann, "Variation von Lege-Strukturen zur Wissensrepräsentation," in *Struktur-Lege-Verfahren als Dialog-Konsens-Methodik.*, vol. 25, B. Scheele, Ed. Aschendorff, 1992, pp. 2–41.

[9] P. Farrugia, K. P. Camilleri, and J. C. Borg, "A language for representing and extracting 3D geometry semantics from paper-based sketches," *J. of Visual Languages and Computing*, vol. 25(5), pp. 602–624, 2014.

[10] D. Wüest, N. Seyff, and M. Glinz, "Semi-automatic generation of metamodels from model sketches," Proc. ASE 2013, pp. 664–669, 2013.

[11] P. Rittgen, "Collaborative Modeling - A Design Science Approach," *Proc. of HICSS 2009*, pp. 1–10, 2009.

[12] T. Herrmann and A. Nolte, "Combining Collaborative Modeling with Collaborative Creativity for Process Design," in *COOP 2014 - Proceedings of the 11th International Conference on the Design of Cooperative Systems, 27-30 May 2014, Nice (France)*, no. 23, Cham: Springer International Publishing, 2014, pp. 377–392.

[13] F. M. Santoro, M. R. S. Borges, and J. A. Pino, "CEPE: Cooperative Editor for Processes Elicitation.," *HICSS 2000*, vol. 1, p. 10, 2000.

[14] J. Mendling, C. Hahn, and J. C. Recker, "An exploratory study of IT-enabled collaborative process modeling," *Workshop-Proceedings of BPM 2010*, vol. 66, p. 61, 2011.

[15] S. Do-Lenh, F. Kaplan, and P. Dillenbourg, "Paper-based concept map: the effects of tabletop on an expressive collaborative learning task," *Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction*, pp. 149–158, 2009.

[16] E. Hornecker, "A Design Theme for Tangible Interaction: Embodied Facilitation," *Proceedings of the 9th European Conference on Computer-Supported Cooperative Work (ECSCW)*, 2005.

[17] D. Wüest, N. Seyff, and M. Glinz, "Sketching and notation creation with FlexiSketch Team: Evaluating a new means for collaborative requirements elicitation," presented at the Requirements Engineering Conference (RE), 2015 IEEE 23rd International, 2015, pp. 186–195.

[18] S. Oppl and C. Stary, "Tabletop concept mapping," *TEI '09*, pp. 275–282, 2009.

[19] P. Dillenbourg and C. Shen, "Tabletops in Education," STELLAR, 2nd Alpine Rendez-Vous on Technology Enhanced Learning, 2009.

[20] J. Kay, K. Yacef, and R. Martinez-Maldonado, "Collaborative concept mapping at the tabletop," *ACM International Conference on Interactive Tabletops and Surfaces*, pp. 207–210, 2010.

[21] S. Baraldi, A. Del Bimbo, and A. Valli, "Interactive concept mapping through gesture recognition on a tabletop," *Proc. of CMC2006*, 2006.

[22] S. Oppl and C. Stary, "Facilitating shared understanding of work situations using a tangible tabletop interface," *Behaviour & Information Technology*, vol. 33, no. 6, pp. 619–635, 2014.

[23] E. Hornecker, "Graspable Interfaces as Tool for Cooperative Modelling," *Proceedings of IRIS*, vol. 24, pp. 215–228, 2001.

[24] A. Lucchi, P. Jermann, G. Zufferey, and P. Dillenbourg, "An empirical evaluation of touch and tangible interfaces for tabletop displays," *Proceedings of TEI 2010*, 2010.

[25] C. Müller-Tomfelde and M. Fjeld, "Tabletops: Interactive horizontal displays for ubiquitous computing," *Computer*, no. 2, pp. 78–81, 2012.

[26] C. Soares, R. S. Moreira, J. M. Torres, and P. Sobral, "LoCoBoard: Low-Cost Interactive Whiteboard Using Computer Vision Algorithms," *ISRN Machine Vision*, vol. 2013, 2013.

[27] V. T. Nunes, F. M. Santoro, and M. R. Borges, "A context-based model for Knowledge Management embodied in work processes," *Information Sciences*, vol. 179, no. 15, pp. 2538–2554, 2009.

[28] M. Bang and T. Timpka, "Ubiquitous computing to support co-located clinical teams: Using the semiotics of physical objects in system design," *international journal of medical informatics*, vol. 76, pp. S58–S64, 2007.

[29] Y. Jiang, F. Tian, X. L. Zhang, G. Dai, and H. Wang, "Understanding, Manipulating and Searching Hand-Drawn Concept Maps," *Transactions on Intelligent Systems and Technology*, vol. 3, no. 1, pp. 1–21, Oct. 2011.

[30] K. D. Forbus and J. Usher, "Sketching for knowledge capture: A progress

report," presented at the Proceedings of the 7th international conference on Intelligent user interfaces, 2002, pp. 71–77.
[31] G. Stapleton, B. Plimmer, A. Delaney, and P. Rodgers, "Combining sketching and traditional diagram editing tools," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 1, p. 10, 2015.
[32] A. Ghorbel, A. Lemaitre, E. Anquetil, S. Fleury, and E. Jamet, "Interactive interpretation of structured documents: Application to the recognition of handwritten architectural plans," *Pattern Recognition*, vol. 48, no. 8, pp. 2446–2458, 2015.
[33] N. Negroponte, "Recent advances in sketch recognition," presented at the Proceedings of the June 4-8, 1973, national computer conference and exposition, 1973, pp. 663–675.
[34] A. Adamo, G. Grossi, and R. Lanzarotti, "Face Recognition in Uncontrolled Conditions Using Sparse Representation and Local Features," in *Proc. of ICIAP 2013*, Springer, 2013, pp. 31–40.
[35] H. Wechsler, "Face Recognition Methods for Uncontrolled Settings," in *Face Recognition in Adverse Conditions*, M. De Marsico, M. Nappi, and M. Tistarelli, Eds. Hershey, PA, USA: IGI Global, 2014, pp. 38–68.
[36] M. De Marsico, M. Nappi, D. Riccio, and H. Wechsler, "Robust face recognition for uncontrolled pose and illumination changes," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 43, no. 1, pp. 149–163, 2013.
[37] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 785–792.
[38] M. Vogel, T. Warnecke, C. Bartelt, and A. Rausch, "Scribbler---drawing models in a creative and collaborative environment: from hand-drawn sketches to domain specific models and vice versa," presented at the Proceedings of the Fifteenth Australasian User Interface Conference-Volume 150, 2014, pp. 93–94.
[39] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
[40] M. Wei, S. Huang, J. Wang, H. Li, H. Yang, and S. Wang, "The study of liquid surface waves with a smartphone camera and an image recognition algorithm," *European Journal of Physics*, vol. 36, no. 6, p. 065026, 2015.
[41] J. M. Firestone and M. W. McElroy, "Doing Knowledge Management," *The Learning Organisation Journal*, vol. 12, no. 2, 2005.
[42] S. Oppl, "Articulation of subject-oriented business process models," *S-BPM ONE '15*, pp. 1–11, 2015.
[43] S. Schiffner, T. Rothschädl, and N. Meyer, "Towards a Subject-Oriented Evolutionary Business Information System," Proc. of IEEE EDOCW 2014, 2014, pp. 381–388.
[44] M. Neubauer, S. Oppl, C. Stary, and G. Weichhart, "Facilitating Knowledge Transfer in IANES - A Transactive Memory Approach," in *Innovation through Knowledge Transfer 2012*, Springer, 2013, pp. 39–50.
[45] S. Fürlinger, A. Auinger, and C. Stary, "Interactive annotations in web-based learning systems," *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, pp. 360–365.
[46] U. Kannengiesser and S. Oppl, "Business Processes to Touch: Engaging Domain Experts in Process Modelling," *Proceedings of the BPM Demo Session 2015*, vol. 1418, pp. 40–44, 2015.
[47] D. Wachholder and S. Oppl, "Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes," in *S-BPM ONE - Scientific Research*, C. Stary, Ed. Berlin, Heidelberg: Springer, 2012, pp. 145–162.
[48] A. J. Canas, G. Hill, R. Carff, N. Suri, J. Lott, T. Eskridge, G. Gómez, M. Arroyo, and R. Carvajal, "CmapTools: A Knowledge Modeling and Sharing Environment," *Concept Maps: Theory, Methodology, Technology, Proceedings of the 1st International Conference on Concept Mapping. Pamplona, Spain: Universidad Pública de Navarra*, 2004.
[49] T.-S. Hwang and D. G. Ullman, "The design capture system: Capturing back-of-the-envelope sketches," *Journal of Engineering Design*, vol. 1, no. 4, pp. 339–353, 1990.
[50] B. Jonson, "Design ideation: the conceptual sketch in the digital age," *Design studies*, vol. 26, no. 6, pp. 613–624, 2005.
[51] T. Hammond and R. Davis, "Tahuti: A geometrical sketch recognition system for UML class diagrams," ACM SIGGRAPH 2006 Courses, 2006, p. 25.
[52] T. M. Sezgin and R. Davis, "HMM-based efficient sketch recognition," Proceedings of IUI 2005, pp. 281–283, 2005.
[53] L. B. Kara and T. F. Stahovich, "Hierarchical parsing and recognition of hand-sketched diagrams," ACM SIGGRAPH 2007 courses, 2007, p. 17.
[54] C. Koch, M. Neges, M. König, and M. Abramovici, "Natural markers for augmented reality-based indoor navigation and facility maintenance," *Automation in Construction*, vol. 48, pp. 18–30, 2014.
[55] M. Kaltenbrunner and R. Bencina, "reacTIVision: a computer-vision framework for table-based tangible interaction," Proc. of *TEI '07*, pp. 69–74, 2007.
[56] E. Y. Lam and G. S. Fung, "Automatic white balancing in digital photography," *Single-sensor imaging: Methods and applications for digital cameras*, pp. 267–294, 2008.
[57] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285, pp. 23–27, 1975.
[58] K. Saeed, M. Tabędzki, M. Rybnik, and M. Adamski, "K3M: A universal algorithm for image skeletonization and a review of thinning techniques," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 2, pp. 317–335, 2010.
[59] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, "2D Euclidean distance transform algorithms: A comparative survey," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, p. 2, 2008.
[60] C. Stary, "Non-disruptive knowledge and business processing in knowledge life cycles–aligning value network analysis to process management," *Journal of Knowledge Management*, 2014.
[61] P. Hayes, T. C. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff, "Collaborative knowledge capture in ontologies," Proceedings of the 3rd international conference on Knowledge capture, 2005, pp. 99–106.
[62] S. Christov, B. Chen, G. S. Avrunin, L. A. Clarke, L. J. Osterweil, D. Brown, L. Cassells, and W. Mertens, "Rigorously defining and analyzing medical processes: An experience report," in *Models in software engineering*, Springer, 2008, pp. 118–131.
[63] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "Supporting informal design with interactive whiteboards," Proceedings of ACM CHI 2014, pp. 331–340, 2014.

Stefan Oppl received a diploma degree in computer science and a master degree in applied knowledge management from the Kepler University of Linz, Austria, in 2004 and 2007, respectively, and a Ph.D. in computer science from the Vienna University of Technology, Austria, in 2010. He is an Assistant Professor of Business Information Systems at the Kepler University of Linz. His current research interests focus on human-centered support articulation and alignment of knowledge about collaborative work processes.

Christian Stary received the diploma degree in computer science from the Vienna University of Technology, Austria, in 1984, and the Ph.D. degree in usability engineering in 1988, and his Habilitation degree in 1993 from the same university. He is currently a full Professor of Business Information Systems with the University of Linz. His current research interests include the area of interactive distributed systems, focusing on method-driven learning and explication technologies for personal capacity building and organizational development.

Simon Vogl received his Ph.D. degree in computer science 2002 for his work on public interactive displays from the Kepler University of Linz, Austria, after a MSc in computer science in 1998 from the same university. His research interests lie in sensor-rich applications to simplify complex work environments, which was the focus of his academic career and is a development focus of his company that he runs full time since 2012.