Journal of Intelligent Manufacturing (2021) 32:201–220
https://doi.org/10.1007/s10845-020-01568-z

A transformation of human operation approach to inform system design for automation

Simon Micheler[1] · **Yee Mey Goh**[1] · **Niels Lohse**[1]

Received: 15 March 2019 / Accepted: 28 March 2020 / Published online: 5 June 2020

Abstract

Design of automation system relies on experts' knowledge and experience accumulated from past solutions. In designing novel solutions, however, it is difficult to apply past knowledge and achieve design *right-first-time*, therefore wasting valuable resources and time. SADT/IDEF0 models are commonly used by automation experts to model manufacturing systems based on the manual process. However, function generalisation without benchmarking is difficult for experts particularly for complex and highly skilled-based tasks. This paper proposes a functional task abstraction approach to support automation design specification based on human factor attributes. A semi-automated clustering approach is developed to identify key functions from an observed manual process. The proposed approach is tested on five different automation case studies. The results indicate the proposed method reduces inconsistency in task abstraction when compared to the current approach that relies on the experts, which are further validated against the solutions generated by automation experts.

Keywords Task analysis · Human factors · Clustering · Task function · Process design · Automation · Manufacturing

Introduction

The current manufacturing sector is experiencing transformational changes. New paradigms like Industry 4.0, Robotic and Autonomous Systems (RAS) and a decentralisation of decision-making driven by implementation of agent-based intelligent systems have the potential to significantly enhance the capabilities of manufacturing (Foresight 2013; Zhong et al. 2017). Adapting to technical challenges and skill shortages is of crucial importance for modern manufacturing businesses to stay competitive in a globalised market (Sungur et al. 2016). A potential solution is through increasing the use of sensors to enhance flexibility and intelligence of automation solutions in applications beyond repetitive, dangerous and traditional automation tasks, and decrease in the total cost of ownership. However, more efficient ways to support the implementation of automation into manufacturing businesses are required in the context of Industry 4.0 (Yao et al. 2019; Kong et al. 2019).

✉ Yee Mey Goh
y.goh@lboro.ac.uk

[1] Intelligent Automation Centre, Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK

Many industries, particularly in the UK, have not embraced automation due to the uncertainty and difficulty to design solutions that are flexible and meet the current productivity of manual processes. Root causes for feasibility related problems dealing with the automation of human tasks have been recognised (Bainbridge 1983; Reason 1987; Xiao et al. 1997), in particular related to the requirement for deep knowledge of the tasks for automation. (Goodrich and Boer 2003) stated "a lesson learned from process automation is that, in the absence of human factors consideration, even state-of-the-art technological systems can be more problematic than beneficial". Part of the challenges is describing human tasks in a useful way as the basis for implementing automation (Everitt et al. 2015).

The investigation of human factors is the subject of extensive research in complementary areas. Examples are the investigation of trust-through-transparency impact on performance (Oduor and Wiebe 2008) or the impact of system complexity on trust (Bailey and Scerbo 2007). Even though the analysis of human tasks for automation has been investigated, for example Caird-Daley et al. (2013), no reliable way to automatically abstract human tasks has been identified when decision-makers evaluate automation business cases.

In this paper, a clustering-based approach is proposed to identify automation functions from a human task analysis

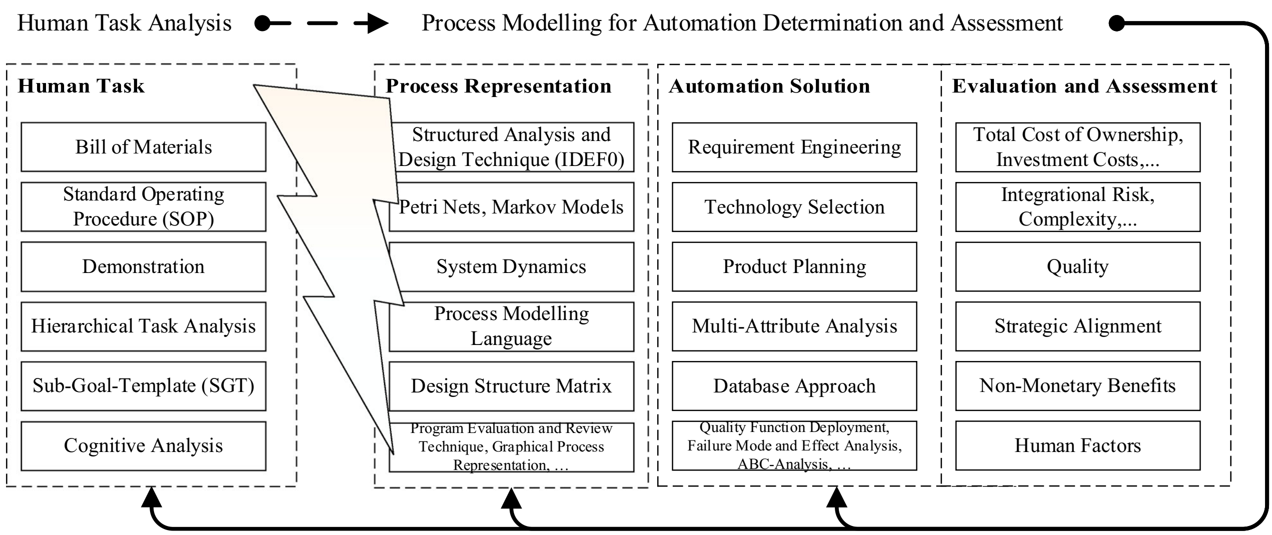


Fig. 1 Stages in automating a manual manufacturing process

enabling a more comprehensive and consistent specification of automation design for business case evaluation. The underlying assumption is that the automated solution is not necessarily like the manual task due to different skills and capabilities of humans and robots (de Winter and Dodou 2014). Nevertheless, the ability to identify the functions provided by manual processes will enhance and widen the solution search space in the early design phase prior to technology selection.

The paper starts with a literature review ("Literature review" section) to identify the research gap and the methodology ("Proposed method" section) describes how a functional abstraction of human tasks is developed and introduces the proposed method before case studies are used to evaluate and compare the results to industrial experts' solutions ("Results" section). Finally, the last section discusses and concludes the paper reflecting on the contributions to the research topic ("Discussion and conclusions" section).

Literature review

Research in manufacturing automation has highlighted the need to consider human factors (Goodrich and Boer 2003). The general process of automating a manufacturing process consists of the creation of a process representation model derived from a manual task to design and create a technical solution. Figure 1 summarises the current methods and tools to support the automation design from the human task analysis to the evaluation and assessment. In this paper, the authors assume an automation process would generically follow the demonstrated stages:

1. *Understanding the human task*
2. *Formalising the process representation*
3. *Synthesising the process representation model for automation*
4. *Evaluating the proposed automation solution*

Starting with the collection of information about the human task via a human task analysis (HTA), process representation models are used to formalise the production process. The review of literature suggests there is a disconnect due to a weak link between HTA performed by human factors experts and process representation models typically constructed by engineers. Furthermore, the transition of a task analysis into the process representation model is exposed to the expert's interpretation. A solution is then created by synthesising the process model to specify the appropriate automation solution, relying on the accuracy and completeness of the formalised representation.

Each of the steps in this process, as well as the overall process, may be conducted iteratively until a desired level of usability is attained. However, iterations are expensive and time consuming. The first three stages of this process are reviewed for the methods and tools used to date. The final stage is an evaluation of the designed solution against the business case, considering economic and other factors before the solution is commissioned. For brevity, the readers are referred to other publications (Ketipi et al. 2014; Koulouriotis and Ketipi 2014) for more detail.

Human task analysis

Human factors has been studied widely in the manufacturing domain but their focus is wide-ranging and not necessarily on automation. For automation, the literature ranges from applications of artificial intelligence to automatically transfer human skills via demonstration to automation systems

Table 1 Efficiency, effectiveness and empirical evidence in task analysis research derived and extended from Crystal and Ellington (2004)

Perspective	Technique	Efficiency	Effectiveness	Evidence
Continuous	Machine learning	Task demonstration	Works rather on action than on process level	Zhao et al. (2016)
		Learning from demo		
Discrete	Hierarchical task analysis (HTA)	Decompose complex tasks into subtasks	Improves problem diagnosis and useful for concurrent operations	Sheperd (2005) and Annet and Stanton (2004)
		Complex activities demand extensive hierarchy construction	Does not account for system dynamics	
Discrete-elemental	Sub-Goal Template (SGT)	Builds upon HTA	Improves the level of detail	Ormerod et al. (1998)
		Decompose tasks into actions using elemental building blocks	Irreproducible results due to lack of user expertise possible	
Cognitive	Cognitive task analysis (CTA)	Defines a coherent knowledge representation of the domain being studied	Increases the understanding of cognitive aspects of the task	Salmon et al. (2010)
			Captures task expertise	
			Fails to fully incorporate learning, contextual and historical factors	
Humanist	Activity theory	Analyse the activity, not the task, implying a potentially great increase in scope and complexity	Accounts for learning effects	Kuutti (1995)
		Requires in-depth knowledge of culture and social aspects	Extents scope of technology	
			Requires a high level of abstraction	
			No disciplined set of methods	
			Difficult to apply systematically	
Demanding	Competency assessment	Analyse the required work skills needed for a specific task	Improves understanding of the workers' skill sets needed for a specific task.	Green et al. (2016), Perry and Helmschrott (2014)
		Literacy, numeracy and problem-solving skills analysed	Does not consider process order	

(Chuck et al. 2017) and automation component mapping, for example a humanoid task-component mapping (Hanai et al. 2016). Other studies contribute towards the mental assessment and strains on humans combined with related decision-making for automation (Caird-Daley et al. 2013).

In this paper, the focus will be on a comprehensive task analysis and decomposition informing later processes of automation. The current state-of-the-art covers task decomposition (Phipps et al. 2011) from a physical and mental perspective. Other literature discusses the analysis of human tasks illustrating the importance of learning via demonstration, the hierarchical task analysis (HTA), the Sub-Goal Template (SGT), the Conceptual Task Analysis (CTA), and the work-process and Program for the International Assessment of Adult Competencies (PIAAC) approach. Table 1 summarises the efficiency and effectiveness for a selection of commonly referred methods to date.

The automation research community recognised the issue related to the abstraction of tasks without losing substantial information. Some researchers have investigated the HTA and cognitive work analysis to produce a comprehensive picture of manufacturing task analysis and present a variety of different applications (Stanton 2006; Salmon et al. 2010). The first approach by Phipps et al. (2011) extended the

HTA by adding cognitive elements of tasks and information design requirements adding significant detail to the current knowledge of manufacturing task analysis. Caird-Daley et al. (2013) executed a task decomposition based on an HTA to capture physical and cognitive tasks to include the physical analysis for automation. Fasth-Berglund and Stahre (2013) confirmed the need to consider cognitive as well as a physical task as part of the automation strategies for reconfigurable and sustainable systems. Based on an HTA analysis, Everitt and Fletcher (2016) tackled the goal of a "robust, formal skill capture for assessing the feasibility and implementation of intelligent automation". Their dual methodology approach combines the existing HTA methodology with a classification system aimed to further increase the understanding of what an automated solution might look like. They extended the analysis with human perception senses, and a specific task classification, as well as a description of the decisions.

The research literature demonstrates the transfer of human tasks into automated tasks is essential. Yet, a transfer has not been achieved without substantial effort for the user in combination with a limited area of applications requiring detailed domain knowledge (Wong and Seet 2017; Pedersen et al. 2016; Wantia et al. 2016). In short, Zhao et al. (2015) pointed out a knowledge gap, where the transition from an extended HTA process towards automation system design is still missing.

Process representation method

A model is used to systematically represent the inner relations and functions of a system in an abstract way, according to a certain perspective, and to reduce the complexity. A key aspect of production research is the development of process representation models. Numerous standard process-representation models have been developed. Existing process representation models in manufacturing are divided into four different categories: production layout, production information, production schedule, and production optimisation:

- *Production layout* represents a category for process representation tools capturing the setup of the production system. Models are used to describe the dimension of the production system, as well as skills and capabilities and the components of the production system.
- *Production information* refers to the related models used to provide information about the manufacturing system. These models often display requirements of the production process. The process models can inform various aspects such as the representation of knowledge, dependencies between tasks, or the workflow and value stream within the production system.
- *Production scheduling* describes a category of representations for modelling the schedule of a production system. The representations are used to provide information about the time structure of a production process, the overlap within the production, as well as the transition from one production moment to another. The results can later be fed into optimisation tools or inform the design of the production system.
- *Production optimisation* methods are used to improve the current situation by using a model that describes a production reality. An improvement of the situation can be achieved by predicting a future outcome, identifying bottlenecks, or optimising the service at production stations.

The current landscape of existing production representation models is analysed, and the outcomes are summarised in Table 2. The third column is provided to indicate whether the models will require the input of a task analysis, which was described in "Human task analysis" section.

The results presented in Table 2 indicate a lack of connection between the manual task analysis and the process representation models. As evident from the table, many of these models do not require a task analysis to be completed, which suggests that the process models are usually made by the opinion/understanding of engineers and not necessarily based on observation of how the tasks are being performed by the operators. This is important in automation because any decisions/variations performed by the operators that is not captured in the process models will be missed when specifying automation solution. Therefore, for a successful translation of a human task into an automation solution, this gap needs to be overcome.

Automation design specification

Four common ways are generally used to specify automation systems, namely requirements engineering, database approaches, technology selection, and generic design models. *Requirements engineering* is the development of an automation system based on anticipated requirements from the previous manual production process (Kaindl et al. 2009). The basis of this production process is an observation of requirements. The requirements can be recorded or extracted differently like, for example, via demonstration or requirements analysis as well as a via a combination with process representation models (Park et al. 2009). Examples for requirements engineering can be found throughout the literature related to design and architecture for holistic solutions (Hegenberg et al. 2012), functional requirements for reconfigurability and flexibility (Boschi et al. 2016), vision systems (Sitte and Winzer 2007), or software testing (Chung and Hwang 2007). *Database approaches* have been used to improve the design of automation solutions driven by CAD and composite structural data (Mayer et al. 1992; Hunten et al. 2013; Sanders et al. 2016). The overall use of *multi-*

Table 2 Process representation models

Process representation model	Abbrev.	Task analysis required	Production layout	Production information	Production schedule	Production optimisation
Control theory models	CTM (Ragazzini and Bergen 1954)	No				
Goals, operators, methods, selection	GOMS	Yes/No				
HAMSTERS	HAMSTERS	No				
Task (-RELATED) KNOWLEDGE STRUCTure	TKS (Johnson et al. 1988)	Yes/No				
Signposting	SP (Clarkson and Hamilton 2000)	No				
Activity Networks	(Flowcharts, PERT) (Kelley Jr and Walker 1959)	No				
Activity/phase overlapping	AO, PO (Krishnan et al. 1997)	No				
Generalised precedence relation	GPR (Elmaghraby and Kamburowski 1992)	No				
Graphical evaluation and review technique	GERT (Pritsker 1966)	No				
Petri nets	PN (Zhou 1995)	No				
Markov models	MM (Doltsinis et al. 2014)	No				
System dynamics	SD (Forrester 1997)	No				
Design structure matrix	DSM (Radice et al. 1985)	Yes/No				
Structured analysis and design technique	SADT, IDEF0 (Ross and Schoman 1977)	Yes				
Business process modelling	BPM (White and I B M Corp. 2005)	Yes/No				
Input-process-output, entry-task-validation-exit	IPO, ETVX (Radice et al. 1985)	Yes/No				
Process grammars/languages	UML,SysML, YAWL,… (Ryo Hanai et al. 2012)	Yes/No				
Value stream mapping	VSM	No				
Queuing theory		No				
Research aim	ESDS	Yes				

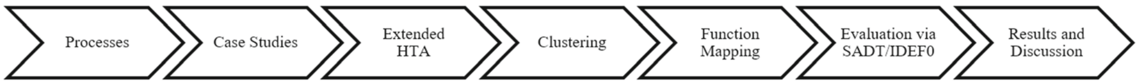


Fig. 2 Applied methodology

attribute analysis in the context of technology selection is a decision-making support based on multiple criteria. For most of the models, the criteria may also be qualitative or subjective. The number of multi-criteria decision-making tools has steadily increased since the last decades and commonly used approaches include the Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), and the Weighted Sum Model (WSM) (Roy 1968; Ketipi et al. 2014; Koulouriotis and Ketipi 2014). Other *generic design methods* that can be used for the system design can be found in the tools Quality Function Deployment (QFD), Failure Mode and Effects Analysis (FMEA) and Activity Based Costing (ABC) analysis to select different processes for manufacturing based on cost, quality and risk aspects (Hassan et al. 2010). These methods require robust input information from the process requirements.

Summary

Although significant progress has been made regarding the analysis of a production task, the issue of transferring and mapping of human tasks against automation has not yet been tackled sufficiently. Several authors have contributed to the decomposition of human tasks and adding detail to the HTA. However, the additional detail were reliant on the experts leading to a higher variance and reduced reproducibility (Sheperd 2005). The problem is a reliable and systematic transfer of the qualitative human factors into requirements engineering. One could envisage the automation of the task analysis, using technology such as tracking and activity recognition (Rude et al. 2018), could be performed in the future. Therefore, the identified research gap in manufacturing literature is to provide an approach to *bridge the gap between task analysis and the automation system design based on the identified task functions*. The next section presents the methodology adopted in this paper to address the current knowledge gap.

Proposed method

The methodology presented in Fig. 2 is used to develop the approach to bridge the gap between the manual task analysis and automation system design. The authors propose a semi-automated approach using a clustering algorithm to classify and map the manufacturing tasks against automation functions. More specifically, after the analysis of human tasks, the paper presents a classification scheme allowing task breakdown from a process to be mapped against automation functions. Those functions can later be linked to automation components.

Table 3 HTA example—welding case study reported by Sanchez-Salas (2016)

HTA level	Process hierarchy level
1 Setup	Process
1.1 Select filler rod	Task with 1 operation
1.2Set up welding torch	Task with 4 operations
1.2.1 Select electrode	Operation
1.2.2 Grand tip of the electrode	Operation
1.2.3 Select collet and ceramic nozzle	Operation
1.2.4 Assemble torch	Operation
1.3 Prepare the parent metal for welding	Task
1.3.1 Remove grinding leftovers	Operation
1.3.2 Setup welding pieces in a welding fixture	Operation
…	…
2 Simulate laying a weld	Process
2.1 Place foot on foot pedal, and depress	Task with 1 operation
2.2 Put on gloves	Task with 1 operation
…	…

Hierarchical task analysis (HTA)

For each case study, multiple methods currently used for the decomposition of human tasks are combined. The initial input to the proposed approach is a hierarchical task analysis (HTA) as presented in Table 3. The table shows a fraction of the task analysis for the welding case study (the full operational structure is shown in Table 5). The task is decomposed into operations performed during a manufacturing process. At the same time, the operations are sorted with respect to time in a chronological manner starting with the first task (only operations are used in the clustering process in "Clustering" section).

The data structure established of the operation analysis is shown in Fig. 3, based on a defined decomposition structure. The hierarchical task structures of five case studies presented in "Results" section are used initially and extended to include the different SGT elements based on Ormerod et al. (1998). Every operation is labelled with a name and a specific sequential ID. The operation contains not only physical actions but

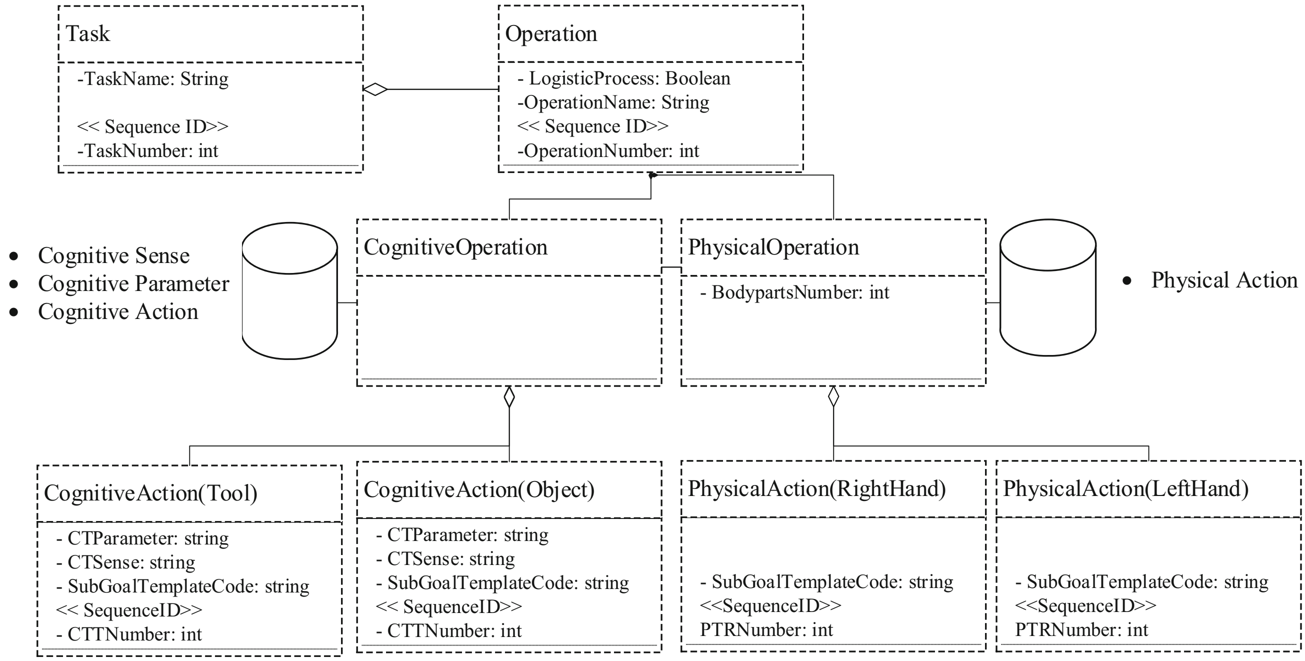


Fig. 3 Extended HTA data structure for operation analysis

also cognitive actions performed during a manufacturing process. The cognitive activity is related to the object or the tool, whereas the physical activity is related to specific multiple body parts.

After the HTA is performed on the process, the clustering algorithm is applied to identify the task functions.

Clustering

The application of clustering is to create task functions based on a HTA. The necessary requirement to enable clustering is a database to represent specific operation attributes of a specific action. Those attributes should be used to differentiate among dissimilar task functions. Before the clustering algorithm can be applied, however, a decision had to be made on the granularity as well as the factors of interest for the application. In this case, the clustering algorithm is abstracting manufacturing functions to define automation solutions. Therefore, a classification scheme is developed based on existing standards and the existing literature to attribute the process operations.

Classification scheme

The first task in developing a classification scheme is to identify existing classifications to enable a structured separation of manufacturing operations. At this point, different attributes according to the current literature of human factors are feasible for the classification process. For the application of a classification in this paper, the author has selected manufacturing attributes to create a functional task abstraction.

A careful analysis of the environment in this case has led the authors to the DIN8580 standard, which is (numerically) followed by other more specific standards. The application categories represent sub-levels of the manufacturing main categories joining, forming, etc. presented in DIN 8580. It is noted that the presented manufacturing classification considers physical manufacturing operations only. Supporting operations related to the perception mechanisms (visual perception, haptic feedback) are not covered in the related classification.

Hence, the authors have extended the existing manufacturing classification standards with perception mechanisms. A combination of research by Groover (2007) with Lederman and Klatzky (1987) informed the classification scheme. The first adapted part by Groover presents a categorisation of visual perception mechanisms for robotic automation. The second part incorporated the research by Lederman et al. focuses on the tactile perception of the humans. In accordance with their findings, a classification containing multiple perception attributes for a specific operation has been developed. The result is a combination of tactile and visual perception senses as a decision criterion to identify the required sensorial requirements (see Fig. 4). Examples of resulting operation attributes are *Visual Perception Object Shape* or *Tactile Perception Temperature.*

Table 4 presents the detailed developed classification used for this work. As it can be seen it demonstrates both percep-

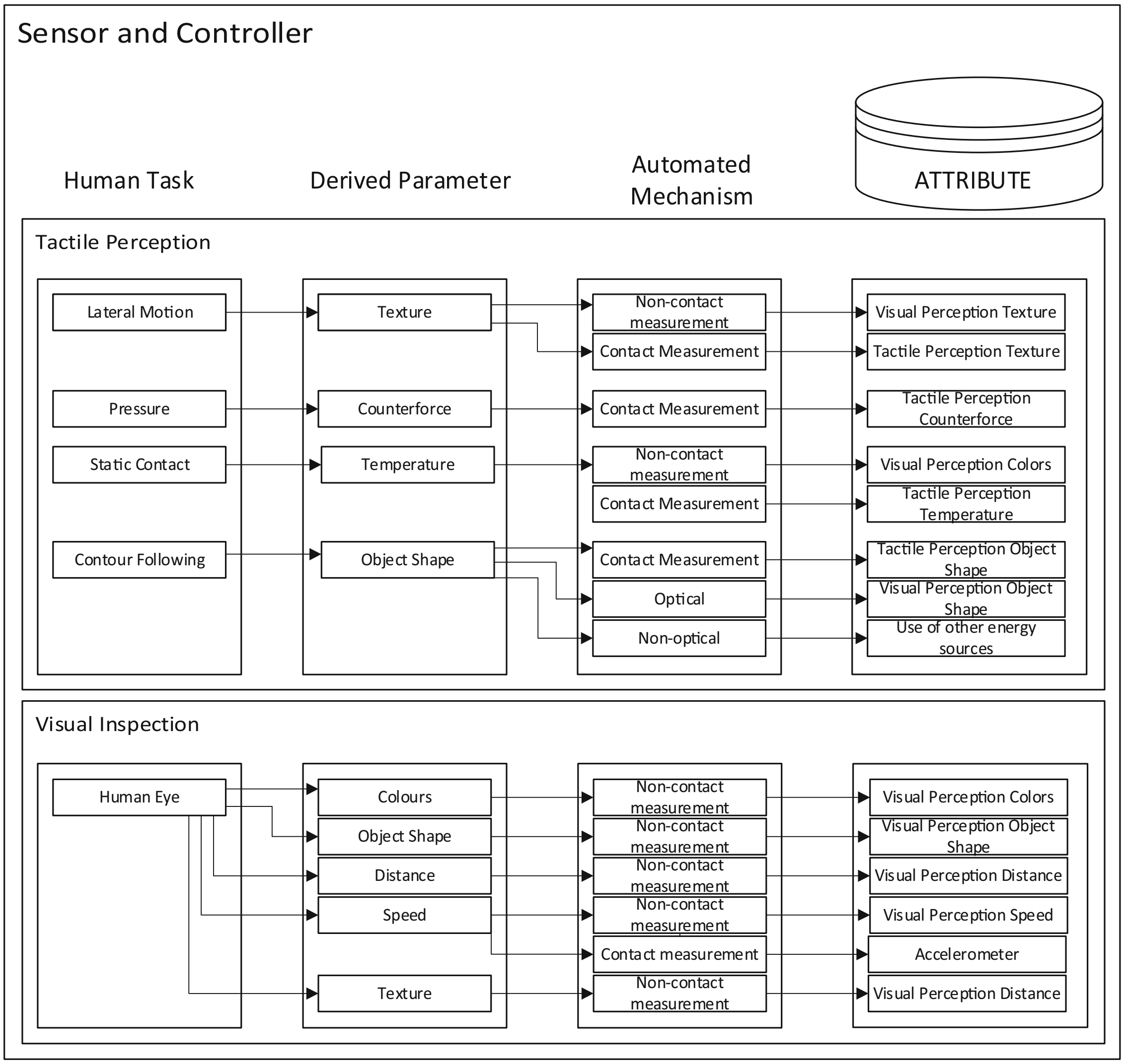


Fig. 4 Tactile and visual perception senses as extension for DIN8580

tional and operational classification for the identification of automation functions.

Clustering for automated function identification

The manufacturing classification against the operation as many-to-many relationship is expressed by binary variables. One operation can have multiple attributes and one attribute can be connected to multiple operations. Consequently, a database is created. The user will identify the attributes for every operation as depicted in Table 5 for welding case study (see "Results" section).

The user will enter as many different attributes to the manufacturing operations as the HTA requires. Every operation must be fully determined with the physical and perceptional attributes defined in the classification scheme. One operation can require multiple attributes for an operation. An example is *grinding tip of the electrode* operation (Operation 1.2.2) performed by an operator. In reality, the grinding operation is not just determined by an attribute responsible for grinding (*cutting with geometrically undefined cutting edges)*, but also requires human-haptic feedback to control the production process (*auxiliary operation*).

For every process operation *i* recorded via the HTA analysis, the *process attribute* a_{ij} related to the *manufacturing classification attribute j* is represented as a binary value.

$$Process\ Attribute\, a_{i,j} \quad a_{i,j} \in \{0, 1\} \tag{1}$$

The binary value expresses, whether the specific process step incorporates operations that fulfil the criteria/pattern of a specific distribution of attributes. The different process attributes result in an attribute matrix A, which can be created as a result of the previous Eq. (1):

Table 4 Selection of classification categories based on standards around DIN8580

Attribute	Attribute	Standard
Changing material characteristics through transfer of particle	a1	DIN 8580
Changing material characteristics through particle screening out	a2	DIN 8580
Changing material characteristics through particle insertion	a3	DIN 8580
Coating from a gaseous or vaporous state	a4	DIN 8580
Coating from a liquid or mushy state	a5	DIN 8580
Coating from ionised state through electrolytic or chemical separation	a6	DIN 8580
Coating from a solid or powdery state	a7	DIN 8580
Pick and place	a8	DIN 8593-1
Filling (e.g. impregnating)	a9	DIN 8593-2
Pressing in and on (e.g. screwing/riveting)	a10	DIN 8593-3
Joining through primary shaping (e.g. grouting)	a11	DIN 8593-4
Joining through forming (e.g. seaming)	a12	DIN 8593-5
Joining through welding (e.g. Laser-, WIG—welding)	a13	DIN 8593-6
Joining through soldering	a14	DIN 8593-7
Gluing	a15	DIN 8593-8
Textile joining	a16	DIN 8593-9
Severing	a17	DIN 8588
Cutting with geometrically defined cutting edges	a18	DIN 8589
Cutting with geometrically undefined cutting edges	a19	DIN 8580
Removal operations	a20	DIN 8590
Disassembling	a21	DIN 8590
Cleaning	a22	DIN 8592
Forming under compressive conditions	a23	DIN 8583
Forming under compressive and tensile conditions	a24	DIN 8584
Forming under tensile conditions	a25	DIN 8585
Forming by bending	a26	DIN 8586
Forming under shearing conditions	a27	DIN 8587
Primary shaping from liquid state	a28	DIN 8581
Primary shaping from plastic state	a29	DIN 8581
Primary shaping from mushy state	a30	DIN 8581
Primary shaping from powdery or granular state	a31	DIN 8581
Primary shaping from fibrous or filamentary state	a32	DIN 8581
Primary shaping from gaseous or vaporous state	a33	DIN 8581
Primary shaping from ionised state	a34	DIN 8581
Tactile perception texture	a35	EXTENSION
Tactile perception counterforce	a36	EXTENSION
Tactile perception temperature	a37	EXTENSION
Tactile perception object shape	a38	EXTENSION
Visual perception colours	a39	EXTENSION
Visual perception object shape	a40	EXTENSION
Visual perception distance	a41	EXTENSION
Visual perception speed	a42	EXTENSION
Visual perception texture	a43	EXTENSION
Tool changing and setup	a44	EXTENSION
Labeling	a45	EXTENSION

$$Attribute\ Matrix\ A \quad A_{dim(i,j)} = \begin{bmatrix} a_{1,1} & \dots & a_{1,j} \\ \vdots & \ddots & \vdots \\ a_{i,1} & \dots & a_{i,j} \end{bmatrix} \tag{2}$$

One process step can have multiple cluster features. The collected process operations are used by the clustering algorithm as a collection of many cases to determine the optimal number of clusters related to the distance measurement between certain clusters with the initial aim to maximise the jump between n-cluster and $(n-1)$-cluster solutions.

The attribute matrix A represents the matrix of the analysed operations and will further be used for the abstraction process. The algorithm aims to divide n operations into k different clusters appending every observation (operation) to a cluster centre (so-called centroid) with the closest mean (Macqueen 1967). The closest mean is related to the distance of the contained clustering attributes from the centroid attributes. K-means clustering is considered difficult from a computational perspective, however, many algorithms converge quickly to an acceptable local optimum (Mahajan et al. 2012).

The generic K-means algorithm is as follows.

A set of observations $(x_1, x_2, \ldots, x_n)$ has an m-dimensional real vector. K-means clustering divides the n observations into k subsets $S = \{S_1, S_2, \ldots, S_k\}$ to minimise the sum of squared distances (Macqueen 1967).

$$k\text{-}means\ Clustering\ Algorithm \quad \underset{s}{arg\,min} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \tag{3}$$

Starting the K-mean clustering with randomised values limited only by the max/min sample value throughout every operation is generally possible. The assumption at this point is, that the patterns will translate into categorical data or attributes carrying binary values. In case the attribute values

Table 5 Process attribute matrix for human operation analysis—welding

Operation name; attributes →	#	Joining through welding	Cutting with geometrically undefined cutting edges	Pick and place	Tool changing and setup	Visual inspection	Visual perception distance
1.1 Select filler rod	1	0	0	0	1	0	0
1.2.1 Select electrode	2	0	0	0	1	0	0
1.2.2 Grind tip of the electrode	3	0	1	0	0	0	0
1.2.3 Select collet and ceramic nozzle	4	0	0	0	1	0	0
1.2.4 Assemble torch	5	0	0	0	1	0	0
1.3.1 Remove grinding leftovers	6	0	1	0	0	0	0
1.3.2.1 Place based on holder on bench	7	0	0	1	0	0	0
1.3.2.2 Attach gas supply	8	0	0	0	1	0	0
1.3.2.3 Secure welding piece	9	0	0	1	0	0	0
2.1 Place foot on foot pedal, and depress	10	1	0	0	0	0	0
2.2 Put on gloves	11	1	0	0	0	0	0
2.3 Hold torch in right hand using pen grip	12	1	0	0	0	0	0
2.4 Hold filler rod in left hand	13	1	0	0	0	0	0
2.5 Move torch and filler rod	14	1	0	0	0	0	0
2.6 Adjust equipment position	15	0	0	0	0	0	1
2.7 Remove objects impeding movement	16	0	0	1	0	0	0
3.1.1 Set and turn on power at the welding set	17	1	0	0	0	0	0
3.1.2 Turn on gas at the gas cylinder	18	1	0	0	0	0	0
3.1.3 Put on welding mask (visor raised)	19	1	0	0	0	0	0
3.2.1 Position torch at tack location	20	1	0	0	0	0	1
3.2.2 Pull down visor	21	1	0	0	0	0	0
3.2.3 Pick up and position filler rod	22	1	0	0	0	0	0
3.2.4 Fully depress foot pedal	23	1	0	0	0	0	0
3.2.5 Dip filler rod in centre of the weld pool	24	1	0	0	0	0	0
3.2.6 Remove rod	25	1	0	0	0	0	0
3.2.7 Gradually release foot pedal	26	1	0	0	0	0	0
4.1 Position torch at weld start	27	1	0	0	0	0	1
4.2 Pick up and position filler rod	28	1	0	0	0	0	0
4.3 Fully depress foot pedal	29	1	0	0	0	0	0
4.4.1 Stroke filler rod in and out of weld pool	30	1	0	0	0	0	0
4.4.2 Feed filler rod through the fingers	31	1	0	0	1	0	0

Table 5 continued

Operation name; attributes →	#	Joining through welding	Cutting with geometrically undefined cutting edges	Pick and place	Tool changing and setup	Visual inspection	Visual perception distance
4.5 Control torch movement	32	1	0	0	0	0	1
4.6 Modulate current	33	1	0	0	0	0	0
4.7 Control foot pedal	34	1	0	0	0	0	0
5.1 Take off equipment	35	1	0	0	1	0	0
5.2 Turn off power and gas supply	36	0	0	0	1	0	0
5.3 Remove welding plates from piece holder	37	0	0	1	0	0	0
5.4.1 Visually inspect top surface of weld	38	0	0	0	0	0	1
5.4.2 Visually inspect under surface of weld	39	0	0	0	0	0	1

are all binary, the identity matrix $I_{j,n}$ can be used to represent the starting centroids for the clustering process to advance the centroid handling algorithm explained in detail in the following paragraph.

$$Centroid\ Matrix\ C \quad C_{dim(j,n)} = \begin{bmatrix} 1\,0\,0 & & \\ 0\,1\,0 & \ldots & 0 \\ 0\,0\,1 & & \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix} = I_{j,n} \tag{4}$$

K-means minimises the distance between the centroids $c_{j,n}$ and the attribute matrix by manipulating the centroid matrix C to reduce the distance vector D_{opt} values over all distances. The following functions show a detailed description of the steps needed to achieve the aim of Eq. (3).

$$Distance\ d_{i,n} \quad d_{i,k} = \left(\sum^{i} (a_{i,k} - c_{j,k})^2 \right)^{1/2} \tag{5}$$

The distance matrix D can be expressed according to the following equation:

$$Distance\ Matrix\ D \quad D = \begin{bmatrix} d_{1,1} & \ldots & d_{1,k} \\ \vdots & \ddots & \vdots \\ d_{i,1} & \ldots & d_{i,k} \end{bmatrix} \tag{6}$$

Table 6 depicts an example of a distance matrix for the welding case study.

The created distance matrix can now be optimised in a way that the distances are being minimised for different sizes of k. The value k represents the number of different cluster-centres used. The optimal solution creates a distance vector D_{opt}, which can be minimised using the sum of distances. The optimised distances in the previous table are marked by a (*). The distance vector represents the smallest distance of every column distance ($d_{i,1}$, …, $d_{i,n}$) according to the following equation.

$$Minimum\ Distance\ min\ D_{opt} \quad min\ D_{opt} = \sum_{i=1}^{n} \left(\min_{1 \le k \le n} d_{i,k} \right) \tag{7}$$

The results of the equation are the minimum distances of different centroids. Table 7 shows different accumulated differences for specific k.

Five different centroids are used to cluster the existing sample. An increase of the cluster number k leads to an overrepresentation of centroids as the distances converge to zero. A comparison of the distances is an indication of the k-effectiveness. Once the distances are all zero for k, the centroids are purely a representation of all the single cases available (in terms of attribute distribution) and did not achieve the goal of reducing the dimension of the operation.

A possible solution to address this issue is a selection of an optimal k via an investigation of the distances between $min\ D_{opt}$. As seen in Table 7, the optimal distances can be summarised to understand how well a specific number of centroids k covers the attribute vectors of the created attribute matrix A. Two criteria should be respected for the evaluation of a suitable cluster number k:

- Firstly, k cannot be chosen in a way that allows the production of a trivial solution. A trivial solution means the selection of different cluster centroids k reproducing the original operations. Such an approach would not effectively reduce or cluster the operations.

Table 6 Distance matrix ($k=5$)—example welding

Distances	d1	d2	d3	d4	d5
1.1 Select filler rod	1.4142	1.4142	1.4142	0*	1.4142
1.2.1 Select electrode	1.4142	1.4142	1.4142	0*	1.4142
1.2.2 Grind tip off the electrode	1.4142	0*	1.4142	1.4142	1.4142
…	1.4142	1.4142	1.4142	0*	1.4142
…	1.4142	1.4142	1.4142	0*	1.4142
	1*	1	1	1	1
	1.4142	1.4142	1.4142	0*	1.4142
	1.7321	1.7321	1*	1.7321	1
	0*	1.4142	1.4142	1.4142	1.4142
…	0*	1.4142	1.4142	1.4142	1.4142
5.4.2 Visually inspect under surface	0*	1.4142	1.4142	1.4142	1.4142

Table 7 Minimum distance matrix D_{Opt}—welding case study

Minimum distance	Min2	Min3	Min4	Min5	…	…	Min n
1.1 Select filler rod	1.4142	1.4142	0	0	…	0	0
1.2.1 Select electrode	1.4142	1.4142	0	0	…	0	0
1.2.2 Grind tip of the electrode	0	0	0	0	…	0	0
…	1.4142	1.4142	0	0	…	0	0
…	1.4142	1.4142	0	0	…	0	0
…	1	1	1	1	…	1	0
…	1.4142	0	0	0	…	0	0
…	1.4142	1.4142	0	0	…	0	0
…	1.7321	1	1	1	…	1	0
…	0	0	0	0	…	0	0
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	1.4142	1.4142	1.4142	1.4142	…	0	0
…	1.4142	0	0	0	…	0	0
…	0	0	0	0	…	0	0
…	0	0	0	0	…	0	0
…	1.4142	1.4142	1.4142	1.4142	…	1.4142	0
…	…	…	…	…	…	…	…
Sum	25.945	20.971	13.899*	12.485	…	6.8284	0

- Secondly, k should be pointing out the biggest 'jump' in the sum of optimal distances related to the chosen attribute matrix and cluster number k.

As seen in Table 7, the accumulated optimal distances decrease with a growing k. However, a rapid decrease of the optimised distance vector at a specific time is noticeable. This step points to several clusters significantly reducing the distance to the dataset's attribute distribution. The centroids indicate the main characteristics of the dataset. The next step considers the biggest jump in the optimised solution.

In this case, the biggest jump occurs for the optimal number of cluster centroids $k = 4$ (Table 7, the largest difference between $min\ D_{opt}$ from $k = 3$ to $k = 4$). Particularly the determination of k is a research field itself. The authors have modified the Bayes Information Criterion (BIC) idea to find a possible solution to the depicted problem. The presented solution shows sufficient results for the application. Different reasons are responsible for that:

- Firstly, an HTA generally has a dataset length (number of operations) considerably smaller than what is considered a large dataset in the data analysis community.

Table 8 Hierarchical task structure and allocated centroid—welding example

HTA structure	Allocated centroid	Sequential helper
1.1 Select filler rod	4	Keep
1.2.1 Select electrode	4	Keep
1.2.2 Grind tip of the electrode	2	Keep
1.2.3 Select collet and ceramic nozzle	4	Keep
…	…	…
2.4 Hold filler rod in left hand	1	Keep
2.6 Adjust equipment position	1	Keep
2.7 Remove objects impeding movement	3	Keep
…	…	…

- Secondly, the attribute values are binary ($a_{i,j} = \{0,1\}$) and, therefore, the created distances will be in similar dimensions and do not require a manipulation of the dataset.
- Thirdly, the number of different attributes accumulated within operations is limited.

A combination of those factors reduces the number of cases in all dimensions of the dataset and the minimum distances significantly. For the case studies, the presented criterion was proven to deliver sufficient results, as reported in "Results".

The determination of a specific k, representing the number of centroids, enables the allocation of specific attribute distributions to a specific cluster centre. Based on this distribution, a table (see for example Table 10) can be created presenting the percentage distribution of manufacturing attributes within the clusters. The clusters identified will be assigned as a process function.

The deliverable for this part of the work is a functional task abstraction. Based on the functional task abstraction, where specific manufacturing operation attributes have been allocated to specific centroids.

Before the results of the clusters can be finally demonstrated, the user must also determine whether the specific processes can be allocated in a specific cluster considering sequential criteria. If a process must be performed after another process (for instance grinding on different scales), sequential information must be provided. Therefore, the user is asked to answer for every process step, whether specific operations should be allocated in a stand-alone function due to a sequential importance or it should be kept in the existing cluster ("Keep", see Table 8). This step allows the tool to show the final process function considering sequential constraints.

Results

The proposed approach was tested on five industry processes, detailed descriptions of the cases studies can be found in the cited literature in Table 9. All of them are currently manual processes that are being considered for automation by the companies motivated by efficiency, skill shortage, flexibility and quality. The advanced automation solutions require sensors and intelligence to deal with process variabilities due to the highly skilled nature of the tasks. All studies were recorded and the HTA and IDEF0 representations were produced by the researchers for those processes.

IDEF0 results

First, the IDEF0 results are presented (see Fig. 5) before the results from the clustering algorithms are displayed in the next section. The IDEF0 have been produced by researchers

Table 9 Advanced automation case studies requiring in-depth task analysis

Case study	Description	DIN 8580	Main investigator
Welding	MIG welding	Joining through welding	Sanchez-Salas (2016)
Grinding	Grinding and polishing of complex-shaped surfaces	Cutting with geometrically undefined cutting edges	Kalt et al. (2016) and Kalt (2016)
Beater winding	Production process of drum beaters	Textile joining	Zhao et al. (2016)
Threaded fastener assembly	Automated freeform assembly of threaded fasteners	Assembly	Dharmaraj (2015)
Deburring	Removing defects/burrs from manufactured parts	Cutting with geometrically undefined cutting edges	Sanchez-Salas (2016)

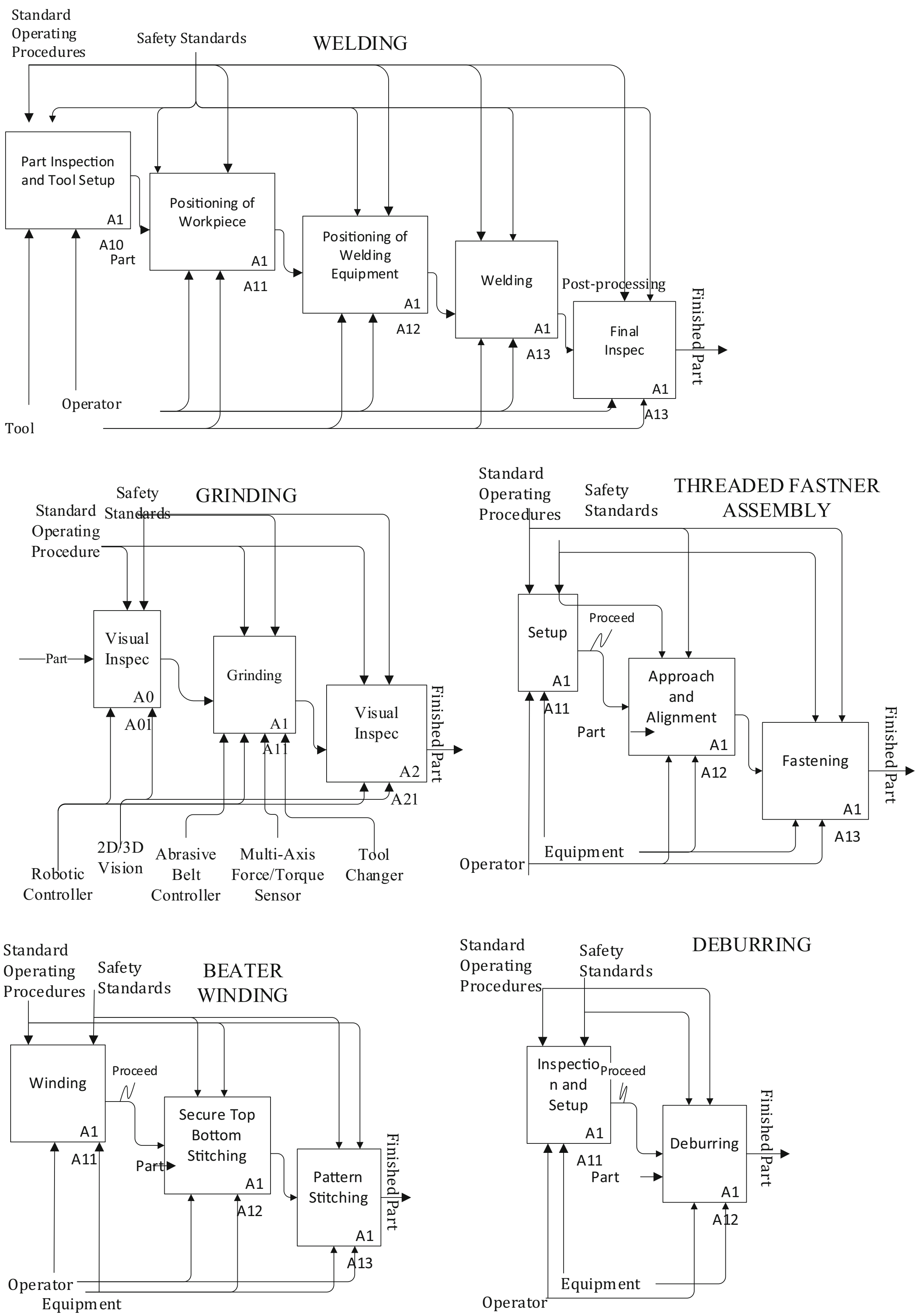


Fig. 5 SADT/IDEF0 results from experts

Table 10 Clustering results—welding

Process function	Joining through welding (%)	Cutting with geometrically undefined cutting edge (%)	Pick and place (%)	Tool changing and setup (%)	Visual perception texture (%)	Visual perception distance (%)	Automation function
1	100	0	0	25	100	100	Welding + inspection
2	0	100	0	0	0	0	Grinding
3	0	0	100	0	0	0	Pick and place
4	0	0	0	75	0	0	Tool changer

Table 11 Clustering results—grinding

Process function	Cutting with geometrically undefined cutting edge (%)	Visual perception texture (%)	Pick and place (%)	Tactile perception (%)	Tool changing and setup (%)	Automation function
1	100	0	0	100	20	Grinding
2	0	100	0	0	0	Visual inspection
3	0	0	100	0	0	Object orientation
4	0	0	0	0	80	Tool changer

Table 12 Clustering results—beater winding

Process function	Textile joining (%)	Pick and place (%)	Tool changing (%)	Visual perception texture (%)	Tactile perception (%)	Cutting with geometrically defined cutting edge (%)	Automation function
1	100	0	0	100	0	100	Sewing
2	0	100	0	0	100	0	Thread winding
3	0	0	100	0	0	0	Tool changer

Table 13 Clustering results—threaded fastener assembly

Process Function	Pressing in and on (%)	Pick and place (%)	Tool changing and setup (%)	Visual perception distance (%)	Visual perception object shape (%)	Automation function
1	100	0	0	50	20	Fastening
2	0	100	0	50	80	Pick and place
3	0	0	100	0	0	Tool changer

who studied the processes for automation, which is the current and widely adopted approach.

Clustering results

The clustering results are presented as a percentage distribution of individual attributes among different process functions for the operations identified in the HTA. In Tables 10, 11, 12, 13 and 14, the process functions (identified clusters) are displayed on the left-hand side and the manufacturing process attributes on top of the table. The manufacturing process attributes are as defined in Table 4, which could be expanded for customised attributes as appropriate to the process.

Analysis and validation

Table 15 compares the functional abstraction with IDEF0 produced by the experts, clustering results and with the actual solutions implemented in the original case studies. The results for each case study are discussed in turn next.

Table 14 Clustering results—deburring

Process function	Cutting with geometrically undefined cutting edges (%)	Tactile perception texture (%)	Tactile perception object shape (%)	Visual perception object shape (%)	Visual perception texture (%)	Visual perception distance (%)	Tool changing and setup (%)	Cleaning (%)	Automation function
1	100	0	0	20	20	100	0	100	Grinding
2	0	100	100	40	40	0	0	0	Visual–tactile control
3	0	0	0	40	40	0	0	0	Visual control
4	0	0	0	0	0	0	100	0	Tool changer

Table 15 Comparison of clustering and IDEF0—results summary

Process	Manual abstraction (IDEF0)	Clustering	Actual solution
Welding	5 functions (preparation = tool setup, positioning, positioning 2, welding, inspection)	4 functions (welding + inspection, grinding, pick and place, tool changer)	4 functions (welding, inspection, tool setup, pick and place)
Grinding	3 functions (part geometry following + visual detection, belt feed rate control + grinding & force/torque, visual inspection)	4 functions (grinding with force/torque sensor, visual inspection, object orientation, tool changer)	2 function (auto-grinding + with manipulator force/torque sensor and gripper, part inspection)
Beater winding	3 functions (winding, secure top bottom stitching, pattern stitching)	3 functions (stitching, customised process = winding, tool changing	3 functions (stitching, winding, tool changing)
Threaded fastener assembly	3 functions (approach and alignment, fastener insertion, torque control)	3 functions (auto-fastening, pick and place, tool changer)	3 functions (auto-fastening, pick and place, tool changer)
Deburring	2 functions (selection of tool = tool setup, removing = deburring)	4 functions (grinding, visual–haptic process control, visual inspection, tool changer)	Not-automated (–)

Welding

The welding process was divided by the automation expert into 5 different functions (Fig. 5), whereas the clustering algorithm has identified 4 key functions (Table 10). The algorithm clustered *welding* (100%) and *inspection* (100% for texture and distance) into one function. A grinding process was identified as a function (Function 2) because in the manual process (and the HTA) the tip of the welding tool was ground by the operator. This step does not occur in the actual automation solution. Despite this, the 4 functions identified from clustering algorithm were accurate. The IDEF0 method divided the process functions in a repetitive pattern of *tool preparation and setup* but *visual inspection* and *grinding the welding tip* were neglected by the experts.

Grinding

The grinding process was manually abstracted into 5 different functions (Fig. 5), in contrast to 4 according to the clustering algorithm (Table 11). The clustering algorithm combines *cutting with a geometrically undefined cutting edge* (100%) with a *tactile force perception* (100%) and a *tool changing* attribute (20%). The second function contains a *visual perception of the part surface/texture* (100%). The turning of the workpiece in between the grinding processes was attributed to the pick and place function (100%). The remaining function is related to a *tool changing* attribute (80%). The automation system implemented consists of a grinding application (abrasive belt) and a separate gripper containing the force/torque sensor element. In addition to that, a visual inspection system was suggested although not implemented. The tool changer was not needed as the actual solution does not require a change of the abrasive belt, due to a different type of component and it was only a prototype solution. The *tactile feedback of force and torque* was allocated to the gripping system since commercial grippers typically come with force and torque sensors.

Beater winding

The beater winding process was manually translated into 3 different functions (Fig. 5). The clustering algorithm also produces 3 functions (Table 12). The first function contains three attributes: *textile joining* (100%), *visual perception of the texture* (100%), and *cutting with a geometrically defined cutting edge* for the thread (100%). This function can be interpreted as the sewing function of the process. The second function contains the attribute for *pick and place* (100%) extended by a measuring device for the counterforce based on tactile measurements. This function represents the winding of the thread. After the winding process is finished, the operator used to cover winding gaps in the pattern. This results in a *visual inspection* to identify the texture and correct the errors. To switch between both functions, a *tool changer* has been identified in function 3. However, the clustering algorithm has attributed the *visual inspection* to the sewing function.

Threaded fastener assembly

The results for threaded fastener assembly process were similar from both the expert (Fig. 5) and the algorithm (Table 13). The expert identified *approach and alignment*, which was identified by the clustering algorithm as the *pick and place* function. The process function included 100% of the *pick and place* attribute as well as 50% of *visual determination of a distance* and 80% of the *visual perception to recognise an object shape*. The fastening function (Function 1) accumulated the remaining visual shares (distance and object shape) and 100% of the '*pressing in and on'* attributes. The remaining percentages were connected to the tool changer. This tool changer is used to switch from a pick and place to a fastening process function. The auto-fastening function include both insertion and torque control, which were identified by the experts as separate functions.

Deburring

The deburring process has not been automated. A possible reason was due to the complexity of the automated solution. This complexity is not specifically indicated by the manual process abstraction (Fig. 5). The manual process abstraction identified three different functions. The functions are a selection of the appropriate tool and the deburring process. The clustering algorithm results (Table 14), however, indicate that a complex tool is needed requiring a visual–haptic process control and a decoupled visual inspection process after that. Those two automation functions require an in-depth knowledge and indicate a high complexity from a programming perspective.

Discussion and conclusions

Two major limitations associated with the current approaches in automation requirements engineering were identified. First of all, approaches related to manual task analysis processes have been criticised throughout the current literature as unreliable (Olsen and Shorrock 2010) and highly influenced by the level of expertise of the analyst (Sheperd 2005). Secondly, the way a task is fulfilled by a human operator might differ highly from the way that the automation system performs the task. Consequently, the method proposed in this paper assumes that a comparison and mapping must take place on a functional level leading to the research aim to provide an approach to *bridge the gap between task analysis and the automation system design based on the identified task functions*.

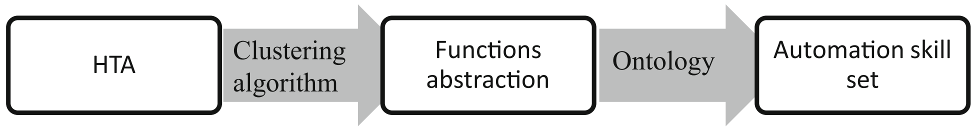
The clustering algorithm overcomes the influence of an HTA hierarchy as perceived by the expert, which overcomes the chronological structure and sequential dependencies unless indicated of the task performance. This means that an attribute of the manufacturing process could be allocated, following the determination of its nearest centroid, independent of the HTA hierarchy thus reducing the expert's influence in the function allocation process. The results indicate that the clustering algorithm achieves the goal of a functional task abstraction and, in some of the cases, the functions identified through the clustering algorithm are closer to the optimal functions required in the actual automation solution.

Based on this functional task abstraction, the issue whereby the functions performed by automation may differ from the manual process is addressed by separating the mapping of HTA to functional task abstraction to automation requirements (Fig. 6). A mapping between the functional abstraction and requirements engineering can be produced to complete the requirements engineering, which is a subject for future work. One possible approach is using an ontology to match functional requirements with the skill set of a specific automation system, see Lohse (2006).

These findings are comparable with the earlier findings, for example Everitt et al. (2015) highlighting the functional approach (Bullock et al. 2013) related to a robotic manipulator as very practical. More detail, in contrast to that, increases the chance of human deviation and, therefore, decreases the repeatability and quality of a task analysis.

Due to the complexity of the individual case studies, only key parts of the results have been presented in this paper. Therefore, the HTA being used in the case studies presented here are reduced for testing the clustering algorithm and it is possible that some iterations were not being captured. An important basis for the method to be transitioned into industrial applications would involve elaborate testing with multiple observations to avoid missing key process information.

Fig. 6 Mapping from HTA to functions to requirements engineering



Acknowledgements The authors acknowledge support from the EPSRC Centre for Innovative Manufacturing in Intelligent Automation in undertaking this research work under Grant Reference No. EP/IO33467/1.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest is arising from the submitted publication.

References

Annet, J., & Stanton, N. A. (2004). Task analysis. In J. Annett & N. A. Stanton (Eds.), *Taylor & Francis E-Library 2004*. London: Taylor & Francis.

Bailey, N. R., & Scerbo, M. W. (2007). Automation-induced complacency for monitoring highly reliable systems: The role of task complexity, system experience, and operator trust. *Theoretical Issues in Ergonomics Science, 8*(4), 321–348. https://doi.org/10.1080/14639220500535301.

Bainbridge, L. (1983). Ironies of automation. *Automatica, 19*(6), 775–779. https://doi.org/10.1016/0005-1098(83)90046-8.

Boschi, F., Zanetti, C., Tavola, G., & Taisch, M. (2016). Functional requirements for reconfigurable and flexible cyber-physical system. In *IECON proceedings (industrial electronics conference)* (pp. 5717–5722). https://doi.org/10.1109/IECON.2016.7794018.

Bullock, I. M., Ma, R. R., & Dollar, A. M. (2013). A hand-centric classification of human and robot dexterous manipulation. *IEEE Transactions on Haptics, 6*(2), 129–144. https://doi.org/10.1109/TOH.2012.53.

Caird-Daley, A., Fletcher, S., & Baker, W. (2013). Automating human skills: Preliminary development of a human factors methodology to capture tacit cognitive skills. In *11th International Conference on Manufacturing Research (ICMR2013)* (pp. 288–295). Cranfield University Press.

Chuck, C., Laskey, M., Krishnan, S., Joshi, R., Fox, R., & Goldberg, K. (2017). Statistical data cleaning for deep learning of automation tasks from demonstrations. In *IEEE international conference on automation science and engineering* (pp. 1142–1149). Xi'an, China: IEEE. https://doi.org/10.1109/COASE.2017.8256258.

Chung, Y. K., & Hwang, S.-M. (2007). Software testing for intelligent robots. In *2007 international conference on control, automation and systems* (pp. 2344–2349). https://doi.org/10.1109/ICCAS.2007.4406752.

Clarkson, P. J., & Hamilton, J. R. (2000). Design 'signposting', A parameter-driven task-based model of the design process. *Research in Engineering Design, 12*(1), 18–38.

Crystal, A., & Ellington, B. (2004). Task analysis and human-computer interaction: Approaches, techniques, and levels of analysis. In *AMCIS 2004 Proceedings* (vol. 391).

de Winter, J. C. F., & Dodou, D. (2014). Why the Fitts list has persisted throughout the history of function allocation. *Cognition, Technology & Work, 16*(1), 1–11. https://doi.org/10.1007/s10111-011-0188-1.

Dharmaraj, K. (2015). *Automated freeform assembly of threaded fasteners*. Loughborough: Loughborough University.

Doltsinis, S., Ferreira, P., & Lohse, N. (2014). An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44*(9), 1125–1138. https://doi.org/10.1109/TSMC.2013.2294155.

Elmaghraby, S. E., & Kamburowski, J. (1992). The analysis of activity networks under generalized precedence relations (GPRs). *Management Science, 38,* 1245–1263. https://doi.org/10.1287/mnsc.38.9.1245.

Everitt, J., & Fletcher, S. (2016). Theoretical issues in ergonomics science task analysis of discrete and continuous skills: A dual methodology approach to human skills capture for automation. *Theoretical Issues in Ergonomics Science, 16*(5), 513–532. https://doi.org/10.1080/1463922X.2015.1028508.

Everitt, J., Fletcher, S., & Caird-Daley, A. (2015). Task analysis of discrete and continuous skills: A dual methodology approach to human skills capture for automation. *Theoretical Issues in Ergonomics Science, 16*(5), 513–532. https://doi.org/10.1080/1463922X.2015.1028508.

Fasth-Berglund, Å., & Stahre, J. (2013). Cognitive automation strategy for reconfigurable and sustainable assembly systems. *Assembly Automation, 33*(3), 294–303. https://doi.org/10.1108/AA-12-2013-036.

Foresight. (2013). *The future of manufacturing: A new era of opportunity and challenge for the UK Summary report*. The Government Office for Science, Project Report, London.

Forrester, J. W. (1997). Industrial dynamics. *Journal of the Operational Research Society, 48*(10), 1037–1041. https://doi.org/10.1057/palgrave.jors.2600946.

Goodrich, M. A., & Boer, E. R. (2003). Model-based human-centered task automation: A case study in ACC system design. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 33*(3), 325–336. https://doi.org/10.1109/TSMCA.2003.817040.

Green, F., Felstead, A., Gallie, D., & Henseke, G. (2016). Skills and work organisation in Britain: A quarter century of change. *J Labour Market Res*, *49*, 121–132. https://doi.org/10.1007/s12651-016-0197-x.

Groover, M. P. (2007). *Automation, production systems, and computer integrated manufacturing*. Edited by Pearson Education India. Prentice Hall. Tsinghua: Pearson Education India. https://doi.org/10.1108/aa.2009.03329dae.001.

Hanai, R., Saito, H., Nakabo, Y., Fujiwara, K., Ogure, T., Mizuguchi, D., et al. (2012). RT-component based integration for IEC61508 ready system using SysML and IEC61499 function blocks. In *2012 IEEE/SICE international symposium on system integration, SII 2012* (pp. 105–110). https://doi.org/10.1109/SII.2012.6426952.

Hanai, R., Suzuki, H., Nakabo, Y., & Harada, K. (2016). Modeling development process of skill-based system for human-like manipulation robot. *Advanced Robotics, 30*(10), 676–690. https://doi.org/10.1080/01691864.2016.1156575.

Hassan, A., Siadat, A., Dantan, J., & Martin, P. (2010). Robotics and computer-integrated manufacturing conceptual process planning—An improvement approach using QFD, FMEA, and ABC methods. *Robotics and Computer Integrated Manufacturing, 26*(4), 392–401. https://doi.org/10.1016/j.rcim.2009.12.002.

Hegenberg, J., Cramar, L., & Schmidt, L. (2012). Task- and user-centered design of a human–robot system for gas leak detection: from requirements analysis to prototypical realization. In *IFAC Proceedings Volumes (IFAC-PapersOnline)* (Vol. 45). IFAC. https://doi.org/10.3182/20120905-3-HR-2030.00076.

Hunten, K. A., Feeney, A. B., & Srinivasan, V. (2013). Recent advances in sharing standardized STEP composite structure design and manufacturing information. *CAD Computer Aided Design, 45*(10), 1215–1221. https://doi.org/10.1016/j.cad.2013.05.001.

Johnson, P., Johnson, H., Waddington, R., & Shouls, A. (1988). Task-related knowledge structures : Analysis, modelling and application. In *BCS HCI* 2001 (pp. 35–62).

Kaindl, H., Arnautovic, E., Ertl, D., & Falb, J. (2009). Iterative requirements engineering and architecting in systems engineering. In *Proceedings of the 4th international conference on systems, ICONS 2009* (pp. 216–221). https://doi.org/10.1109/ICONS.2009.33.

Kalt, E. (2016). *Development of an integrated robotic polishing system*. Loughborough: Loughborough University. https://doi.org/10.1115/1.2757629.

Kalt, E., Monfarad, R. P., & Jackson, M. R. (2016). Towards an automated polishing system—Capturing manual polishing operations. *International Journal of Research in Engineering and Technology, 05*(07), 182–192.

Kelley Jr, J. E, & Walker, M. R. (1959). Critical-path planning and scheduling. *Papers Presented at the December 1–3, 1959, Eastern Joint IRE-AIEE-ACM computer conference* (pp. 160–173). https://doi.org/10.1145/1460299.1460318.

Ketipi, M. K., Koulouriotis, D. E., & Karakasis, E. G. (2014). Robot evaluation and selection part B: A comparative analysis. *International Journal of Advanced Manufacturing Technology, 71*(5–8), 1395–1417. https://doi.org/10.1007/s00170-013-5526-4.

Kong, X. T. R., Luo, H., Huang, G. Q., & Yang, X. (2019). Industrial wearable system: The human-centric empowering technology in industry 4.0. *Journal of Intelligent Manufacturing, 30*(8), 2853–2869. https://doi.org/10.1007/s10845-018-1416-9.

Koulouriotis, D. E., & Ketipi, M. K. (2014). Robot evaluation and selection part A: An integrated review and annotated taxonomy. *International Journal of Advanced Manufacturing Technology, 71*(5–8), 1371–1394. https://doi.org/10.1007/s00170-013-5525-5.

Krishnan, V., Eppinger, S. D., Whitney, D. E., Krishnan, V., Eppinger, S. D., & Whitney, D. E. (1997). A model-based framework to overlap product development activities. *Management Science, 43*(4), 437–451.

Kuutti, K. (1995). Activity theory as a potential framework for human–computer interaction research. *Context and Consciousness: Activity Theory and Human-Computer Interaction*, 17–44.

Lederman, S. J., & Klatzky, R. L. (1987). Hand movements: A window into haptic object recognition. *Cognitive Psychology, 19*(3), 342–368. https://doi.org/10.1016/0010-0285(87)90008-9.

Lohse, N. (2006). Towards an ontology framework for the integrated design of modular assembly systems. *System*.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297). Berkeley: University of California Press.

Mahajan, M., Nimbhorkar, P., & Varadarajan, K. (2012). The planar—means problem is NP-Hard. *Theoretical Computer Science, 442,* 13–21. https://doi.org/10.1016/j.tcs.2010.05.034.

Mayer, R. J., Chuan Jun, S., & Keen, A. K. (1992). An Integrated Manufacturing Planning Assistant—IMPA. *Journal of Intelligent Manufacturing, 3*(2), 109–122. https://doi.org/10.1007/BF01474750.

Oduor, K. F., & Wiebe, E. N. (2008). The effects of automated decision algorithm modality and transparency on reported trust and task performance. In *Proceedings of the 52nd human factors and ergonomics society annual meeting* (pp. 302–306).

Olsen, N. S., & Shorrock, S. T. (2010). Evaluation of the HFACS-ADF safety classification system: Inter-coder consensus and intra-coder consistency. *Accident Analysis and Prevention, 42*(2), 437–444. https://doi.org/10.1016/j.aap.2009.09.005.

Ormerod, T. C., Richardson, J., & Shepherd, A. (1998). Enhancing the usability of a task analysis method: A notation and environment for requirements specification. *Ergonomics, 41*(11), 1642–1663. https://doi.org/10.1080/001401398186117.

Park, J. L., Jin, H. S., Kim, K. H., & Kim, J. J. (2009). A study on real-time progress management system modeling using goal & scenario based requirements engineering. In *ICCIT 2009—4th international conference on computer sciences and convergence information technology* (pp. 262–265). https://doi.org/10.1109/ICCIT.2009.281.

Pedersen, M. R., Nalpantidis, L., Andersen, R. S., Schou, C., Krueger, V., & Madsen, O. (2016). Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing, 37,* 282–291. https://doi.org/10.1016/j.rcim.2015.04.002.

Perry, A., & Helmschrott, S. (2014). *User guide for the German PIAAC Scientific Use File* (Vol. 23).

Phipps, D. L., Meakin, G. H., & Beatty, P. C. W. (2011). Extending hierarchical task analysis to identify cognitive demands and information design requirements. *Applied Ergonomics, 42*(5), 741–748. https://doi.org/10.1016/j.apergo.2010.11.009.

Pritsker, B. (1966). *Graphical evaluation and review technique*. RM-4973-NASA (Vol. RM-4973-NA). Washington DC.

Radice, R. A., Roth, N. K., O'Hara, A. C., & Ciarfella, W. A. (1985). A programming process architecture. *IBM Systems Journal, 24*(2), 79–90.

Ragazzini, J. R., & Bergen, A. R. (1954). A mathematical technique for the analysis of linear systems. *Proceedings of the IRE, 71,* 1645–1651.

Reason, J. (1987). Cognitive aids in process environments: Prostheses or tools? *International Journal of Man-Machine Studies, 27*(5–6), 463–470. https://doi.org/10.1016/S0020-7373(87)80010-X.

Ross, D. T., & Schoman, K. E. (1977). Structured analysis for requirements definition. *IEEE Transactions on Software Engineering SE, 3*(1), 6–15. https://doi.org/10.1109/TSE.1977.229899.

Roy, B. (1968). Classement et Choix En Présence de Points de Vue Multiples. *RAIRO Operations Research Recherche Opérationnelle, 2*(V1), 57–75.

Rude, D. J., Adams, S., & Beling, P. A. (2018). Task recognition from joint tracking data in an operational manufacturing cell. *Journal of Intelligent Manufacturing, 29*(6), 1203–1217. https://doi.org/10.1007/s10845-015-1168-8.

Salmon, P., Jenkins, D., Stanton, N., & Walker, G. (2010). Hierarchical task analysis vs. cognitive work analysis: Comparison of theory, methodology and contribution to system design. *Theoretical Issues in Ergonomics Science, 11*(6), 504–531. https://doi.org/10.1080/14639220903165169.

Sanchez-Salas, A. (2016). *A framework to support automation in manufacturing through the study of process variability*. Loughborough: Loughborough University.

Sanders, A., Elangeswaran, C., & Wulfsberg, J. (2016). Industry 4.0 implies lean manufacturing: Research activities in industry 4.0 function as enablers for lean manufacturing. *Journal of Industrial Engineering and Management, 9*(3), 811–833. https://doi.org/10.3926/jiem.1940.

Sheperd, A. (2005). Hierarchical task analysis. In *Taylor & Francis E-Library 2005*. Vol. e-Book. London: Taylor & Francis. https://doi.org/10.1017/CBO9781107415324.004.

Sitte, J., & Winzer, P. (2007). Methodic design of robot vision systems. In *Proceedings of the 2007 IEEE international conference on mechatronics and automation, ICMA 2007* (pp. 1758–1763). https://doi.org/10.1109/ICMA.2007.4303816.

Stanton, N. A. (2006). Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics, 37*(1 SPEC. ISS.), 55–79. https://doi.org/10.1016/j.apergo.2005.06.003.

Sungur, C. T., Breitenbücher, U., Leymann, F., & Wieland, M. (2016). Context-sensitive adaptive production processes. *Procedia CIRP, 41,* 147–152. https://doi.org/10.1016/j.procir.2015.12.076.

Wantia, N., Esen, M., Hengstebeck, A., Heinze, F., Rossmann, J., Deuse, J., & Kuhlenkoetter, B. (2016). Task planning for human robot interactive processes. In *IEEE international conference on emerging technologies and factory automation, 2016* (Vol. 21, pp. 1–8). IEEE.

White, S., & I B M Corp. (2005). Using BPMN to model a BPEL process. *Business, 3*(March), 1–18.

Wong, C. Y., & Seet, G. (2017). Workload, awareness and automation in multiple-robot supervision. *International Journal of Advanced Robotic Systems, 14*(3), 1–16. https://doi.org/10.1177/1729881417710463.

Xiao, Y., Milgram, P., & John Doyle, D. (1997). Planning behavior and its functional role in interactions with complex systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 27*(3), 313–324. https://doi.org/10.1109/3468.568740.

Yao, X., Zhou, J., Lin, Y., Li, Y., Yu, H., & Liu, Y. (2019). Smart manufacturing based on cyber-physical systems and beyond. *Journal of Intelligent Manufacturing, 30*(8), 2805–2817. https://doi.org/10.1007/s10845-017-1384-5.

Zhao, Y. C., Al-Yacoub, A., Goh, Y. M., Justham, L., Lohse, N., & Jackson, M. R. (2016). Human skill capture: A hidden Markov model of force and torque data in peg-in-a-hole assembly process. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 000655–000660). IEEE. https://doi.org/10.1109/SMC.2016.7844315.

Zhao, Y. C, Johson, T., Goh, Y. M., & Mike, J. (2015). A sensor design and data analysis for automatic drum beater winding. In *International conference on engineering design, IDED 2015* (pp. 1–10).

Zhong, R. Y., Xun, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: A review. *Engineering, 3*(5), 616–630. https://doi.org/10.1016/J.ENG.2017.05.015.

Zhou, M. (1995). *Petri nets in flexible and agile automation*. Boston, MA: Springer. https://doi.org/10.1007/978-1-4615-2231-7.