







Automating the Generation of MBD-Driven Assembly Work Instruction Documentation for Aircraft Components

Aikaterini R. Papadaki¹, Konstantinos Bacharoudis¹ , David Bainbridge¹,
Nick Burbage², Alison Turner¹, David Sanderson¹ , Atanas A. Popov¹ ,
and Svetan M. Ratchev¹ 

¹ Institute for Advanced Manufacturing, Faculty of Engineering, University of Nottingham,
Jubilee Campus, Nottingham NG8 1BB, UK

David.Sanderson@nottingham.ac.uk

² Hamble Aerostructures Ltd, Kings Avenue, Hamble-le-Rice, Southampton SO31 4NF, UK

Abstract. The classical approach to the creation of assembly work instructions for high value, complex products is time-consuming and prone to error. It requires a process engineer to write the work instructions step-by-step and manually insert specific technical information, using an encompassing document of manufacturing parameters or life cycle management software. The latter offers synchronisation to design changes through updateable parameters, however major design modifications still require significant manual work to modify the text contents and structure of the work instructions. This leaves the work instruction documentation vulnerable to human error, as well as making the process time-consuming to fully synchronise. A methodology was therefore developed to resolve these issues, utilising JavaScript and VBA for Office to create a simple interface for rapid content generation for work instructions including text, MBD extracted parameters, images and formatting. The overall methodology speeds up the creation of assembly work instructions and reduces errors by implementing automatic insertion of parameters from an MBD model. The implementation and effectiveness of the suggested approach is demonstrated on a case study for the assembly of the joined wing configuration of the RACER helicopter, the latest generation of compound helicopters of Airbus Helicopter.

Keywords: Assembly work instructions · Model-based definition · Joined wing · Digital manufacturing

1 Introduction

Assembly work instructions play a significant role in assembly quality and lead times, capturing the necessary assembly activities and relevant technical information that engineering technicians must read, understand, and apply in order to successfully assemble the product under development. For high value, low volume, complex products such as aircraft components, production of work instructions is particularly elaborate as they

require a vast number of processes. Many of these processes are also repetitive, occurring at several locations in the aircraft structure, e.g. drilling pilot holes, countersinking, reaming, etc.

The classical approach for creating work instruction documentation first requires a process engineer to retrieve the necessary technical information scattered in CAD models (2D drawings), in spreadsheet documents or through the tacit knowledge of process and design engineers. This technical information must be accumulated and then accurately captured into a document or software. Typical cycle times for the design process in the aerospace field is approximately 5 years [1], therefore retrieving the information accumulated in this period can be a very time-consuming activity, prone to many errors. An inability to create flawless assembly work instructions can jeopardise the entire design activity and confuse or frustrate operators, resulting in poor quality products [2].

Nowadays an increasing number of aerospace companies, such as Boeing and Airbus, are adopting the model-based definition approach [3] to alleviate the problem of managing large amounts of data. Many works discuss and implement MBD technology in the field of manufacturing, e.g. Quintana et al. [4] and Geng et al. [5], however few consider the implementation of MBD to enable product design synchronisation with work instructions. Furthermore, despite fast progression in this field, current research on MBD is limited, particularly for assembly work instructions [3].

A range of modern software packages for product lifecycle management (PLM) are available; one example is Siemens Teamcenter, which offers the extraction of digital work instructions and documentation from 3D models. The specific functionality offered by Teamcenter is the synchronisation of design changes to digital work instructions, and visualisation through interactive 3D assembly sequences. However, the majority of such software available present, including Teamcenter, suffer from the same issues: they can be time consuming to learn and use, and the primary focus is the generation of illustrations for use in technical documentation, not the instructional text contents which are equally vital for shop floor operators.

Alongside these PLM software packages, augmented reality (AR) is one of several technologies that have been devised to optimise the visual representation of these work instructions, thereby improving clarity of their contents (see, for example, Gattullo et al. [6]). This technology is an alternative to traditional documentation and provides a wide range of potential benefits to assembly lines, such as shorter learning time for operators, reduced training costs and overall product quality improvement as seen in various studies [7, 8]. Zauner et al. [9] demonstrates a highly intuitive AR tool that guides the user through simple assembly steps for furniture, with potential for application on more complex assemblies in the future. Although considerable progress has been made in recent years, there are still significant obstacles preventing the incorporation of AR in the production of aircraft structures. These include limitations on object detection using markers for small objects (such as screws, bolts, etc.), high computational requirements, and relatively large upfront investments for equipment installation and training. Furthermore, the majority of AR solutions are offline and therefore any consensus leading to redesign of the product and the assembly processes requires a considerable amount of

time and effort to amend the work instruction documentation, since they cannot synchronise to design changes and must be rebuilt again to produce updated instructions [10, 11]. Few works tackle this issue in the field of augmented reality.

Alternate solutions for the automation and improved visual representation of work instructions have also been developed. Geng et al. [5] developed a lightweight tool that is easily accessible by the end user, utilising automatically generated, interactive CAD images embedded in PDFs for improved cognition of work instructions. However, there is no instructional text generated and it is acknowledged that the creation of the work instructions and synchronising design changes are time consuming with this tool, despite its link to MBD, which reaffirms the current gap. Gors et al. [12] also demonstrates a method to determine assembly sequences and generate images to produce work instructions, however there are minimal text instructions for each step. It is therefore clear that few works target the automation of the text content of assembly work instruction documentation, especially when considering the concept of updatable links to manufacturing parameters using MBD.

Based on the presented literature and the everyday industrial practice, there is a clear need for more automatic document generation of assembly work instructions with an updateable link to MBD models, particularly whilst more novel technologies such as AR remain in the experimental stages of implementation. This paper presents a functioning solution to this problem; the methodology is presented in detail in Sect. 2, followed by an exemplary case study in Sect. 3. The selected case study is the joined wing configuration of the RACER (Rapid And Cost Effective Rotorcraft) demonstrator by Airbus [13]. Details of the specific implementation are discussed in Sect. 4. Outcomes of the case study are discussed in Sect. 5, and finally useful conclusions are presented in Sect. 6.

2 Proposed Methodology

This paper suggests a methodology for rapid generation of work instruction documentation, exploiting repetitive assembly processes to automate the process and applying an MBD approach to collate assembly information and error-proof parameter insertion.

Firstly, to enable insertion of manufacturing parameters within the documentation, this methodology presumes the existence of an MBD model in which geometric data and assembly information is captured in a structured approach. The MBD model must be logically structured such that data is extracted and stored in a file format such that data can be identified and retrieved logically. In this way, an updatable and robust link is made between the stored dataset and the MBD. This link ensures major design changes to the product will not compromise the structure of the dataset and therefore the methodology presented can remain synchronised with the product design.

Secondly, a series of process templates for repetitive assembly processes are created. Each template represents the generalised procedure for a particular process, containing user commands that indicate how the instructions will be structured, such as excluding or repeating sections depending on the parameters of the targeted assembly features that the procedure will be carried out on. Placeholders for variables are also used in conjunction with commands to enable MBD parameter insertion during the generation of content. Therefore, a set of chosen process templates must still be created to initiate

this methodology, however, the need for manual insertion of parameters is eliminated and the time consumed creating work instructions can be reduced. It is important to note that the process templates only have to be created once when setting up the methodology. Modifications to the process templates are only required if the processes change, which is likely to be infrequent due to the certification requirements in the aerospace industry.

Lastly, two algorithms are used to create the documentation. The first algorithm systematically retrieves process templates based on user selection and links the template placeholders to the MBD parameters of the selected feature. The second algorithm processes the documentation through a series of tasks, such as opening individual process templates, inserting parameters into placeholders, creating repetition or exclusion of text, inserting images, and finally exporting to a final document. The overall structure is summarised in Fig. 1.

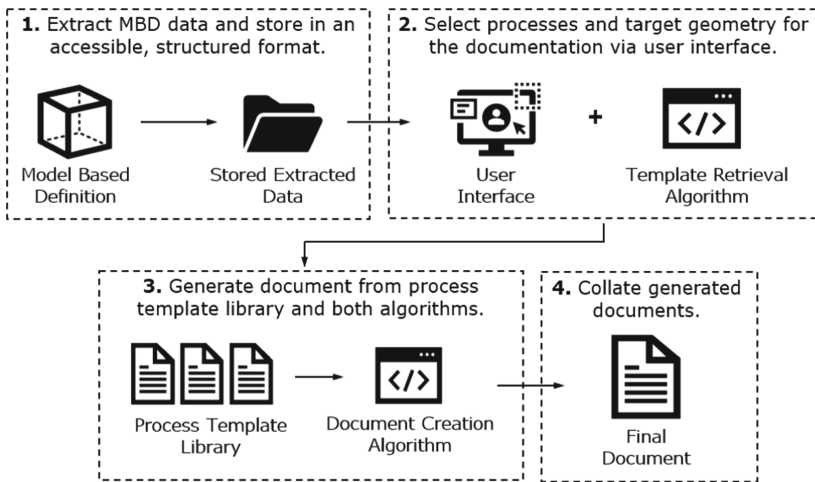


Fig. 1. Flow chart detailing the structure of the methodology.

3 Case Study

The case study selected to demonstrate the developed methodology was the joined wing configuration of the RACER demonstrator [13] shown in Fig. 2.

The specific case study has been selected because sufficient information was present, the assembly processes and assembly sequences associated were relatively complex, and subsequently a successful implementation would mean high applicability to a variety of products. The overall assembly sequence and procedure for this joined wing configuration can be found in [14] and is distinct to a conventional wing assembly, due to the strict and difficult geometric constraints with respect to the location of the wing and the wing interface.

The MBD model of RACER was developed in Dassault's 3DEXperience software, using the concept of the joint definition structure. A joint definition is defined by the

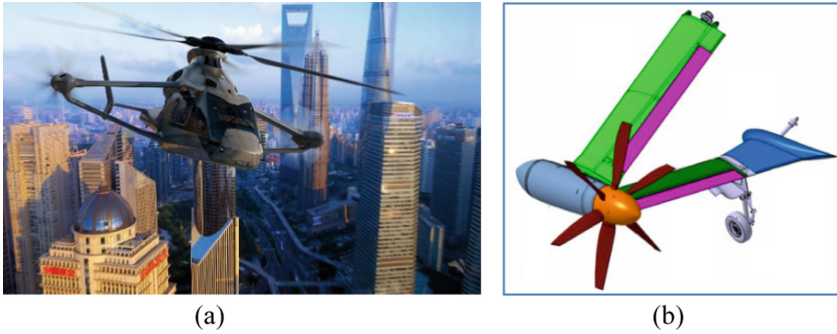


Fig. 2. Images depicting the RACER demonstrator; (a) shows the conceptual image of the rotorcraft and (b) shows the geometry of the jointed wing configuration [13].

contact of two or more interfacing surfaces (an interface group), where multiple joint definitions may be assigned to a single interface group.

An example of an interface group and the contained joint definitions are captured in Fig. 3. Given the interface group “Upper Cover to Mid Rib”, there are three joint definitions associated with the lower wing of the joined wing configuration. Joint definition 14 consists of 11 holes with particular manufacturing parameters, and joint definition 15 and 16 each consist of one hole of differing manufacturing parameters. Together, these three joint definitions make up the “Upper Cover to Mid Rib” interface group.

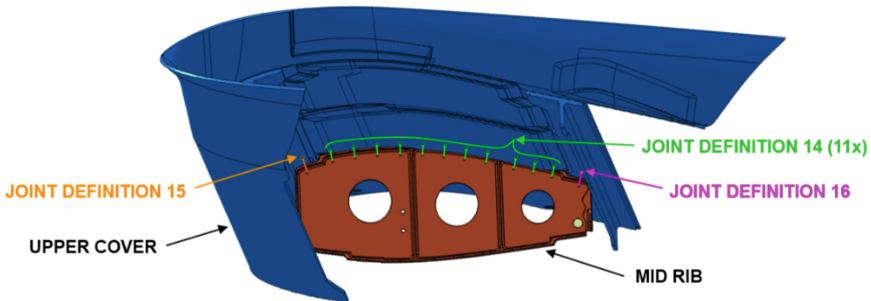


Fig. 3. Joint definition structure displayed within RACER’s MBD model

4 Implementation of Methodology

To implement the methodology using the case study of Sect. 3, a select number of process templates were produced by examining the most common processes in the RACER assembly work instructions, namely pilot drill, full size drill, deburr, shim and final fasten.

Pilot drill, full size drill and final fasten were the most repetitive processes in the sample RACER work instructions, and these processes utilised very similar structures

such that they can be generalised into the pseudocode shown in Fig. 4. With reference to Fig. 4, italic text corresponds to the printed content of the document, bold text to commands, and plain text to other pseudocode contents.

As can be seen in Fig. 4, a for-loop first iterates through each of the selected interface groups, with following lines defining variables such as the name of each interface within the current interface group. Next, an if-statement determines if the current interface group contains two or three components and inserts text accordingly. Additional logic can be employed according to the needs of the process template format. Finally, embedded within the assembly steps of the defined process, a for-loop is used to iterate through the joint definitions of the current interface group. Parameters extracted from the dataset can be inserted throughout the document, including images.

Generic Process Template Logic

Print title of assembly process

let thirdSurfaceNameColumn, jointDefNameColumn, jointDefImageColumn ... = user-defined column indices based on dataset

for each interfaceGroup **in** interfaceGroups: *// interfaceGroup is the index of the row defining the group*

let thirdSurfaceName = dataset[thirdSurfaceNameColumn, interfaceGroup]

if thirdSurfaceName = "n/a" **then**

Print specific text for two surfaces forming the surface group.

else:

Print specific text for three surfaces forming the surface group.

end

print Text of generic numbered steps for the assembly process

let jointDefinitions = A, B, C...

// A list of user-defined row indices (extracted from interface)

for each jointDefinition **in** jointDefinitions: *// jointDefinition is the index of the row defining the group*

let jointDefName = dataset[jointDefNameColumn, jointDefinition]

let jointDefImagePath = dataset[jointDefImageColumn, jointDefinition]

Print text describing the assembly steps with formatting, inserting names, specific manufacturing parameters and images for each jointDefinition, e.g. using jointDefName and JointDefImagePath extracted from the dataset.

next

Fig. 4. Pseudocode demonstrating logic of generalised process template.

The deburr and shim processes appeared less frequently within the example RACER work instructions, however they contained elements that would benefit from creating a process template with commands. For example, the deburr process template requires the total number of holes for a target surface (e.g., Upper Cover) and a breakdown table of these holes per joint definition. To complete this manually, the process engineer must check every interface group that includes the target surface, find each joint definition accordingly, create the table and then sum the total number of holes.

Commands within the process template can be used to automatically extract the total number of holes for a target surface and produce a table that lists every associated joint definition with the number of holes to be deburred. This demonstrates the advantage of the methodology even for less repetitive processes, as it reduces the time needed for document creation by also automating the accumulation and manipulation of parameters.

A user interface has been developed using a JavaScript add-in for Office Excel, with embedded VBA coding to enable document generation. The interface contains a layout

for the user to input the assembly sequence row by row, using dropdown selection. The dropdown selection is linked and updated with the joint definitions and interface groups using the MBD dataset file. This is shown in Columns D, E, F and G of Fig. 5. The developed tool provides great flexibility; if the assembly strategy changes, it is a matter of few minutes for the user to define a new order of assembly processes, selecting specific interface group and joint definitions for each process. It is assumed that a process template already exists for each process. If not, then the user needs to set up the process template once for each new process and further use them through the interface shown in Fig. 5.

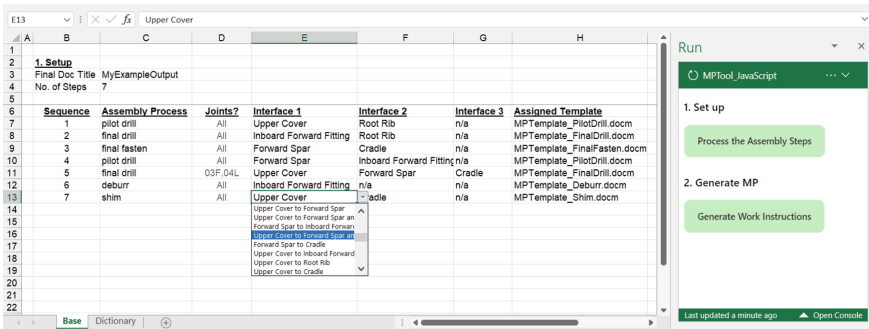


Fig. 5. Image of the user interface (along with the sidebar) in Office Excel.

5 Discussion of Results

A pilot drill process document was produced and formatted according to pre-existing RACER assembly work instructions. The template and final output are shown in Fig. 6(a) and (b) respectively, noting minor post formatting was required such as addition of page numbering and headers. Though these formatting actions can be implemented into the tool, they are typically semi-automated in common word processing programs.

The tool was tested for 10 processes, namely a sequence of four pilot drill processes, two final fasten processes, two final drill processes, one deburr process and one shim process, each with different interface group selections. The approximate time to generate the combined document was 3 min 58 s, containing 23 pages and 6059 words, which included a total of 57 joints. To create approximately 23 pages of work instructions with a more classical approach, an estimate of a few days to a week would be expected. Therefore, it can be seen the timeframe for creating documentation using this methodology is reduced from days, or weeks, to minutes.

Direct comparison to common industrial tools, such as Teamcenter and DELMIA, could not be carried out due to limited access. The literature suggests these tools typically automate image creation, introduce interactive 3D models to the workshop floor and update parameters in real time [15, 16]. However, they require manual text content creation, extensive user training and costly licenses. Therefore, while alternative tools

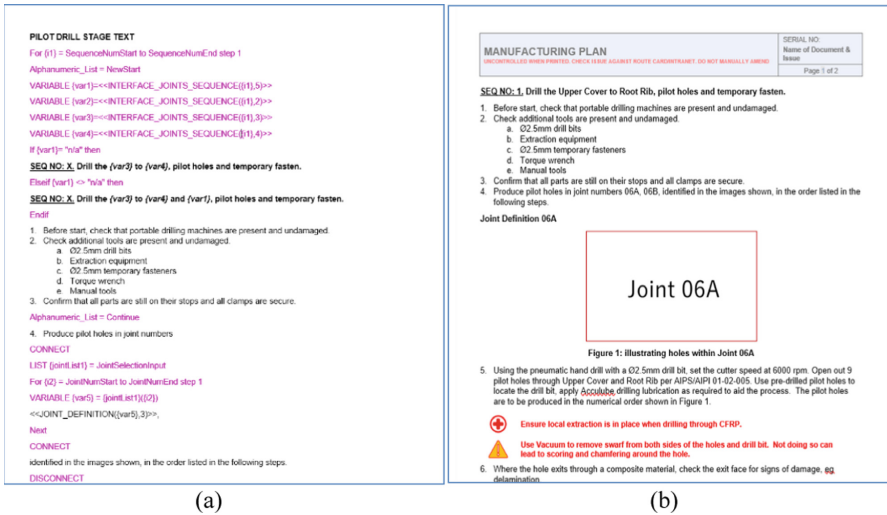


Fig. 6. Images of (a) the pilot drill template, and (b) the resulting document output from the tool.

offer advantages for high quality visuals and cohesive integration with 3D models within the same software package, the demonstrated methodology addresses the shortcomings of these tools. Overall, the methodology provides a low-cost solution for rapid generation of text content for work instructions with updateable, embedded parameters.

6 Conclusions and Future Work

The presented methodology demonstrates the automatic generation of documentation for work instructions with updateable links to MBD data, satisfying the gap identified in relevant literature. Once the methodology is set up then only the user-input instructions are necessary for modifying the assembly sequence or updating the MBD dataset after product design changes. The methodology can be set up by specifying process templates with minimum programming knowledge due to the developed pseudocode whilst still retaining all necessary features and formatting for minimum post-processing. This approach offers the potential to eliminate errors and reduce the time consumed creating documentation by programmatically generating the instructions of assembly processes. This was then verified by the case study implementation, in which the timeframe for instruction creation was shortened significantly.

The methodology also leaves room for future work, such as greater overall automation and collaboration with other methodologies. For example, the implementation of automatic MBD dataset extraction would further speed up the process of assembly work instruction generation. Automatic image extraction and automatic assembly sequence generation from the MBD file could also be implemented for near-complete automation, as demonstrated by Gors et al. [12].

It is also important to note the methodology is not being presented as a complete solution. It generates text for specific processes identified as significantly repetitive, and

not the work instructions in their entirety. However, this tool provides a solution to the issues with the conventional method of assembly instruction production in lieu of more novel approaches that are not yet fully matured.

Acknowledgements. This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement number CSJU-CS2-GAM-AIR-2014-2015. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union.

The tool presented here is available on request; please contact the authors directly to discuss how this can be arranged.

References

1. Shekar, B., Venkataram, R., Satish, B.M.: Managing complexity in aircraft design using design structure matrix. *Concurr. Eng. Res. Appl.* **19**, 283–294 (2011)
2. Claeys, A., Hoedt, S., Schamp, M., et al.: Intelligent authoring and management system for assembly instructions. *Procedia Manuf.* **39**, 1921–1928 (2019)
3. Goher, K., Shehab, E., Al-Ashaab, A.: Model-based definition and enterprise: state-of-the-art and future trends. *Proc. Inst. Mech. Eng. B. J. Eng. Manuf.* **235**(14), 2288–2299 (2020)
4. Quintana, V., Rivest, L., Pellerin, R., et al.: Will model-based definition replace engineering drawings throughout the product lifecycle? A global perspective from aerospace industry. *Comp. Ind.* **61**, 497–508 (2010)
5. Geng, J., Zhang, S., Yang, B.: A publishing method of lightweight three-dimensional assembly instruction for complex products. *J. Comput. Inf. Sci. Eng.* **15**(3), 1–12 (2015)
6. Gattullo, M., Scurati, G.W., Fiorentino, M., et al.: Towards augmented reality manuals for industry 4.0: a methodology. *Robot Comput. Integr. Manuf.* **56**, 276–286 (2019)
7. Frigo, M., da Silva, E.C.C., Barbosa, G.: Augmented reality in aerospace manufacturing: a review. *J. Ind. Intell. Inf.* **4**(2), 125–130 (2016)
8. Doshi, A., Smith, R.T., Thomas, B.H., Bouras, C.: Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing. *Int. J. Adv. Manuf. Technol.* **89**(5–8), 1279–1293 (2016). <https://doi.org/10.1007/s00170-016-9164-5>
9. Zauner, J., Haller, M., Brandl, A.: Authoring of a mixed reality assembly instructor for hierarchical structures. In: *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, pp. 237–246. IEEE, New York (2003)
10. Goher, K., Shehab, E., Al-Ashaab, A.: Trends in model-based definition based assembly information for high-value manufacturing. In: Hedburg, T. Jr., Carlisle, M. (eds.) *Proceedings of the 11th Model-Based Enterprise Summit, MBE*, pp. 99–103. NIST, Maryland (2020)
11. Lavric, T., Bricard, E., Preda, M., et al.: An AR work instructions authoring tool for human-operated industrial assembly lines. In: *Proceedings of the IEEE International Conference on Artificial Intelligence and Virtual Reality IEEE AIVR*, vol. 1, pp. 174–183. IEEE CPS, New Jersey
12. Gors, D., Put, J., Vanherle, B., et al.: Semi-automatic extraction of digital work instructions from CAD models. *Procedia CIRP* **97**, 39–44 (2021)
13. Blacha, M., Fink, A., Eglin, P., et al.: Clean Sky 2: Exploring New Rotorcraft High Speed Configurations. In: *Proceedings of the 43rd European Rotorcraft Forum, ERF*, pp. 989–1000. Curran Associates Inc., New York (2017)

14. Bainbridge, D., Bacharoudis, K. C., Cini, A., et al.: Advanced assembly Solutions for the Airbus RACER Joined-Wing Configuration. SAE Technical Paper 2019-01-1884 (2019)
15. Siemens PLM Teamcenter: <https://www.plm.automation.siemens.com/global/en/products/teamcenter>. Accessed 27 Apr 2022, last accessed 2022/04/27
16. Dassault Systems DELMIA. <https://www.3ds.com/products-services/delmia>. Accessed 27 Apr 2022

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

