Designing Shoestring Solutions: An approach for designing low-cost digital solutions for manufacturing

Gregory Hawkridge, Duncan McFarlane, Jan Kaiser, Lavindra de Silva, German Terrazas

University of Cambridge

Abstract. This paper examines the design of low-cost digital solutions for manufacturing. A set of criteria are established that take into account the limited designer experience and limited time budget that accompany a low-cost project. Alternatives are assessed and a design approach is proposed that addresses these criteria using a set of identified features. Development of the proposed approach is not yet complete, however it already provides a simple, accessible, and streamlined method for implementing low-cost digital solutions.

Keywords: Low Cost · Digital Manufacturing · System Design

1 Introduction

This paper consider an approach for designing pragmatic digital systems that address routine challenges faced by manufacturing companies in a low-cost manner. This design approach forms part of a broader "end-to-end" framework which has been developed to support the digital needs of small manufacturers [1, 2].

Digital tools and systems have the potential to improve efficiency and growth for companies within the manufacturing sector and across the business spectrum [3, 4]. Companies wanting to achieve these objectives tend to do so through large, wide-ranging projects [5, 6]. [5] suggest that "20% of companies plan to invest more than 10% of annual revenues [on Industry 4.0 transformation] in the next five years" based on their 2016 survey. It has been noted that many manufacturers (especially small and medium sized manufacturers) may not benefit from digital transformation to the same extent as others [5, 7, 8, 9]. The organizational and operational changes triggered by digitalization can be significant for such companies. Companies are therefore cautious about the risk and uncertainty of digitalization and often view it as too expensive, too complex and/or oversized [5, 10].

Within this context, there is a need for simple, practical, low-cost digital solutions that can be used to address day-to-day challenges in a low-risk manner [1]. This paper considers the design of such solutions (which is also required to be simple and low in cost). This work forms part of a research program aimed at developing an approach to digital transformation which addresses a common concern that recent developments in digital manufacturing are unlikely to be accessible to SMEs [11, 12, 13, 14]. Although this research has focused on SMEs, larger companies have indicated that there

adfa, p. 1, 2011.

are simple practical problems within their operations where such an approach could also be beneficial as their existing digital transformation processes are overly complex and onerous for such problems.

This paper examines the design and development requirements for low cost digital solution implementation within manufacturing and assesses existing options for the solution structure and design/development process. The deployment and integration of such solutions will be examined in a future paper [15]. An initial approach is then proposed and evaluated against the identified criteria.

The paper is structured as follows: it starts by considering the challenges and context of low-cost solution developments and proposes criteria that a design approach should meet. Target features for the design approach are identified and options are considered. The proposed design approach is then presented along with an illustration of its functioning. The design approach is evaluated and then conclusions are given.

2 Design requirements for industrial low-cost digital solutions

This section considers the context of low-cost solution design and proposes criteria that a design approach for low-cost digital solution development should meet to operate effectively within that context.

2.1 Challenges and context of low-cost digital solution implementation

For the implementation of low-cost digital solutions, the fundamental challenge is that the cost of the solution is not governed by component costs alone, but also includes the cost of time spent designing, developing and later deploying the solution. This challenge is not unique to low-cost solutions, however there are several aspects that are made more significant by the low-cost nature of the project. These features can be grouped under the areas of *project resourcing*, *component selection*, and *solution performance*.

Project Resourcing.

For project resourcing, the limited budget means that it is often hard to justify allocating expert resource (whether in-house or outsourced), and if it can be justified it is typically for short, well understood and time-constrained tasks. As a result, low-cost digital solution projects will usually be carried out by personnel with limited skills or limited experience. A further complication is that the people conducting these projects usually have other roles and responsibilities so developing a low-cost digital solution is seldom their priority. A possible way of alleviating these issues is by leveraging interns, student projects and/or apprentices (an example of how the authors have used this strategy for the development of proof-of-concept systems can be found in [16]).

Component Selection.

For component selection, targeting low-cost hardware and low-cost / open source software components has several effects. To begin with, many of the available options exist within commodity markets outside of conventional manufacturing automation or digitalization markets. In this space, market variations due to new competitors, improved offerings, or new technology occur with greater frequency. This complicates keeping track of available options. Furthermore, low-cost options may not offer the same level of completeness as more expensive industrial alternatives and may need to be combined with other components to be comparable (i.e. hardware components may not come with cables, connectors or enclosures and software components may not provide the same interfaces, or data transfer mechanisms). Similarly, low-cost options may not offer the same levels of ruggedness, reliability, validation or security. Finally, some low-cost options may have incomplete or missing datasheets, manuals, or documentation. Although this assessment of low-cost alternative may suggest that they are not suitable for industrial usage, the key is selecting appropriate applications. Low-cost digitalization is better suited to peripheral, non-safety-critical applications that do not require deep integration with exiting digital infrastructure, typically in the areas of sensing, decision support and staff guidance [14].

A further consideration which relates to both component selection and project resourcing is that, in the context of a low-cost project, the time overhead spent making design decisions can more readily overtake any potential savings to be had from those decisions. A contributing factor to this analysis overhead is the breadth of the search space when considering low-cost components. This can be exacerbated by a lack of information availability, which is not unique to off the shelf-components, but also plays out in the pricing opaqueness of many digital manufacturing solution vendors.

Solution Performance.

Design inexperience and low-cost componentry are sources of concern and uncertainty about solution performance. One concern is whether inexperienced designers will be able to sufficiently understand the problem, capture the requirements and realize a solution. Secondly, once a solution is designed, will the ordered components function as expected, or is their low price point an indicator of inconsistent quality and reduced reliability. A further challenge with low-cost components is that they may require certain expertise and knowledge to utilize them to their full potential.

All the factors discussed in this section show that the implementing low-cost digital solutions can be a non-trivial task especially when carried out by employees with limited skills and experience.

2.2 Criteria for a low-cost solution design approach

To facilitate design within the context discussed in the previous section, we propose the following criteria that a low-cost design approach would need to satisfy:

- **Accessible** – The design approach should allow both novice and experienced designers to develop functional solutions. (The solution developed by a novice may only satisfy certain of a company's requirements or offer limited performance, but it should work.)

- **Simple and structured** – The design process should be easily understood by inexperienced designers. It should follow a logical process and ensure the designer always has a clear next step.
- **Streamlined** – The design approach should accelerate the development process by facilitating quick design decisions, while ensuring that decisions are informed.
- **Versatile** – The design approach should be able to incorporate a large variety of available components, easily interchange similar/equivalent components, and incorporate new components that are developed.

3 Solution design approach

This section proposes a candidate design approach. It starts by identifying target features and then considers alternatives before presenting the proposed approach along with an illustration.

3.1 Target Features

To satisfy the proposed criteria, a number of important features for a design approach were identified. **Table 1** summarises these features and maps their expected contributions. To provide structure, the features have been grouped under the broad areas to which they apply. Where possible these features have been used to guide the low-cost (shoestring) solution design approach.

The first set of features relate to the process that a designer follows when designing a solution. To be accessible and useful for designer of differing capabilities, the design process should offer different levels of detail. Similarly, to be simple and structured, it should be procedural; following a predefined, logical set of steps. Design iterations should be limited or optional. Where, typical design processes tend to be iterative in order to improve the design, however iterations may leave novice designers disheartened. The design process should also assume a high degree of uncertainty and assist novice designers in clarifying the issues they are trying to address.

The second feature set covers tools that assist designers. The first tool is a mechanism for checking component compatibility in the design. This may be an automated tool, or a manual check applied to the design. The second tool checks that the design is complete, which can conversely inform designers about what still needs to be completed. The third tool identifies or estimates the expected level of performance. The final tool is a mechanism for capturing "good design" so that it can be re-used. Many companies have similar low-cost digital needs [12] therefore, it is likely that many designs will be fully or partially similar.

The third feature set relate to the components that form the design. The first feature is a set of common interfaces and data formats that should be prioritised when designing. This relates to the second feature, a classification of interchangeable components for each aspect of a design. Third is a mechanism for including future technological developments. And finally, the approach should not depend on specific technology or infrastructure as this limits the scope of what can be designed and may prevent utilisation of more familiar technology.

Table 1. Feature contributions to criteria (Key: ● Significant ◐ Partial ○ Insignificant)

	Accessible	Simple and Structured	Stream-lined	Versatile
Process				
Variable level of detail	●	◐	●	○
Accommodating of uncertainty	●	○	◐	○
Procedural (design process)	●	●	○	○
Limited / optional iterations (design process)	◐	●	●	○
Tools				
Compatibility verification	●	●	●	●
Completeness verification	●	●	●	○
Performance indications	◐	○	●	◐
Capture and re-use of good design	●	◐	●	○
Components				
Standardised/Common interfaces	○	●	●	●
Classification of interchangeable components	○	◐	●	●
Pathway for including new components	○	○	◐	●
Technology and infrastructure neutral	●	○	○	●

3.2 Design Options

A design approach is unavoidably coupled to the structure of the solution that is being designed and the wider development model within which the design process is conducted. This section discusses alternative development models and structure and how they can facilitate the identified features.

Development Model.

A development model provides a framework to plan and execute the tasks required to deliver a project. Prominent examples include the Waterfall model, the Spiral model, and Agile. A key consideration here is the how these models handle uncertainty. Linear models, such as the Waterfall model, focus on fully understanding requirements and developing a complete specification before designing and implementing a fully-featured system. These all-in-one approaches can be effective for small projects and may require less development effort, however they increase the criticality of the design stage as the design needs to be correct first time.

In contrast, iterative approaches, such as Agile and the Spiral model, separate development into incremental phases which are conducted sequentially to develop an increasingly complete system. These iterative approaches are better suited to low-cost digital solution design as they reduce the criticality of the design stage since uncertainty can be mitigated by using the outputs from previous increments to refine requirements. Additionally, changes and corrections can be addressed in future iterations. A further benefit is that a basic version of the solution is quickly available for use and can start to generate value to end-users. For companies implementing a digital solution, the gradual nature of iterative development can mitigate risks around operational change and resistance to said change.

Structure.

Throughout this paper there has been an underlying assumption that the structure of the low-cost digital solutions being considered will have a modular nature, this is exhibited through the use of terms like component and interface. Modularity has many benefits from a lifecycle perspective as damaged or technically obsolete components can be replaced/upgraded. This is particularly pertinent for low-cost technologies which tend to have shorter lifespans and are subject to more rapid innovation. Further, Modularity is a common method for managing complexity within engineering design, an example of which is the system, subsystem, and component decomposition in the traditional systems engineering V model. The use of such a composition hierarchy is necessary to manage the cognitive load of considering 100s, if not 1000s, of components in large systems. However, such an approach needs to be justified for small, simple systems such as those considered here.

A digital solution will clearly be made from components, however the question is whether an intermediary (subsystem) layer is required. Inexperienced designer will likely find it easier to individually design several subsystems, with few components, than a single system with many components. However, defining and developing subsystem interfaces adds complication and cost. This balance between cognitive load and additional effort swings either way depending on the low-cost application.

The target features for this paper's design method swing the balance towards using a subsystem layer. The use of subsystems enables designers to work at subsystem or a component levels, facilitating the desired feature for offering different interaction levels. Further, subsystems facilitate the capture and re-use of partial solution designs.

3.3 Design approach

This section presents the proposed design approach for shoestring solutions developed using the target features in section 3.1. At the start of the design approach, it is assumed that the designer has identified what type of solution they are developing (In the authors' work this is termed the solution area, see [12] for an overview), and the high level function requirements for that solution using a tool like user stories. This section starts by discussing the structure of the designed solutions, before proposing the approach that is used to design them.

Solution Structure.

Designed solutions use a three tier composition hierarchy (as shown in **Fig. 1**), at the bottom of the hierarchy are building blocks. Building blocks offer basic functionality and are combined to form service modules (the middle layer) which provide high level functionality at a network level. The functionalities provided by a collection of service modules is then composed to realize the required solution. The following section illustrates the design of a solution and includes examples of these different layers.

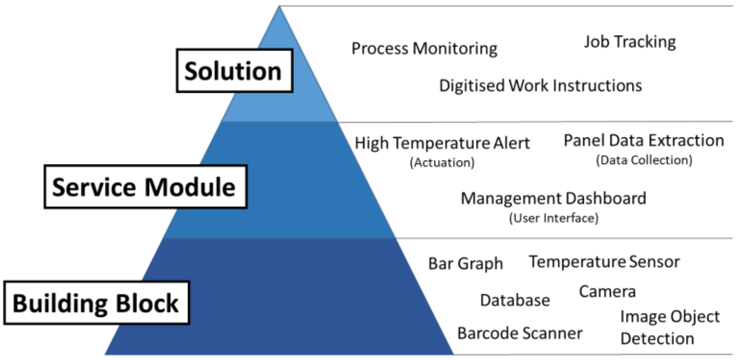

Fig. 1. Solution composition hierarchy (with examples on the right)

Building blocks are modular elements with base functionalities and well defined interfaces. The building block concept helps to address some of the target features related to components. Each building block has two primary attributes: function and interface. Building blocks with the same function can be swapped without affecting the functional capabilities of the design (e.g. temperature sensor, camera, or database). Further, building blocks with the same function and interface can be swapped without affecting compatibility with other parts of the design. In this manner building blocks wrap off-the-shelf products so that designers need not consider their internal complexity. This wrapping can either be conceptual or it can include hardware or software snippets needed to achieve compatibility. For example, a thermocouple probe on its own is not a building block as it does not provide a reliable, easily readable analogue signal, however a thermocouple probe could be combined with cold-junction-compensation and amplification to make a building block.

Service modules operate at a network level; the elementary functions provided by their constituent building blocks are amalgamated into high level functionalities that are made available using the selected service level communication mechanism. The compound functionality provided by each service module falls into one of six categories: Data collection, Analysis, User Interface, State and Data Storage, Actuation, or Decision Making.

Service modules provide the necessary subsystem level discussed in section 3.2 along with the associated benefits presented there, however they also serve as the primary unit of distribution. A simple solution can start with all service modules co-located at the initial deployment location and then certain service modules can be replicated and distributed as required by the expansion of the solution.

Design Process.

The design process has two phases, abstract design and instantiation. In the abstract phase, the designer decides what type of element is needed based on the needed functionality and in the instantiation phase they select the components that will be used to implement that functionality. An overview of the process is shown in **Fig. 2**. The process has three primary levels of detail with a fourth optional “detailed” level.

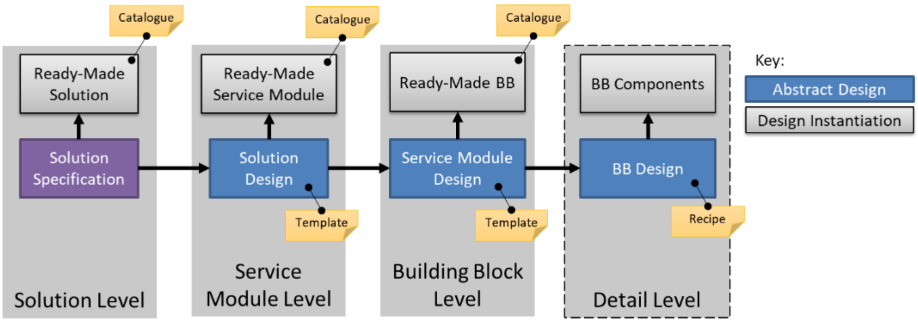

Fig. 2. Design Process

The process starts at the solution level with the solution specification (a high level functional specification, possibly including a set of user stories). If there is a ready-made solution that matches the specification, then it is selected and implemented. If a satisfactory option is not found, the designer can choose to consider the solution at the service module level. Alternatively if the designer lacks the confidence or skills to move to the service module level, they may choose to implement a ready-made solution that only partially meets their needs.

At the service module level, the designer creates a solution design by decomposing the solution into a set of service modules according to the functionality that is required. Then for each service module in the design, the readily available options are considered and selected if appropriate.

Once again, if none are suitable, the designer may choose to move to the building block level where they create service module designs through decomposition into building blocks. Ready-made building blocks can then be selected to instantiate the design. If none exist, as a last ditch effort, the designer may descend to the detail level where they make a custom building block using off-the-shelf components.

Once the designer has instantiated their design at whichever level is necessary/comfortable, they have what is required to move forward to deployment, however this process description has not yet covered two key issues, how decomposition is performed and where the ready-made options come from.

To facilitate decomposition in a manner that is simple and accessible to inexperienced designer a template-based approach is used. Each template has a title, a description of its intended use-case, and a list of its constituent elements. Templates are abstract, so a solution template will detail what types of service modules are required and their functionality. Similarly, a service module template describes the required building blocks in terms of their functionality.

Ready-made options are maintained in catalogues, entries include off-the-shelf products, services, or systems as well as captured instantiations at the next level down. For example, the service module catalogue contains compatible off-the-shelf systems that operate at a network level as well as complete instantiated designs (possibly based on templates) made from building blocks which have been captured and packaged as a whole. Together these two mechanisms allow for the crowdsourcing of good abstract designs (templates) and good instantiated designs (ready-made options).

3.4 Illustrative Design

This section provides a short illustration for how the design approach can be used to develop a low-cost digital solution in the area of job tracking. (A job refers to a piece of work in discrete manufacturing – typically the production of a single item/product.) Consider a designer trying to satisfy two primary user stories: As an operations manager, I want to see were all jobs are in the factory so that I have visibility of the state of production; and, as a shop floor manager, I want to see how long a job has been at a station so that I can ensure no jobs get lost or misplaced.

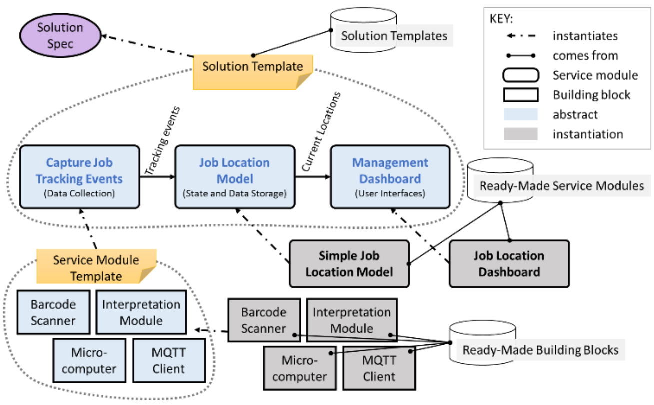

Fig. 3. Illustrative Design

The process followed by the designer is as follows (depicted in **Fig. 3**). The designer starts at the solution level. None of the available ready-made designs are satisfactory so they move to the service module level where a template intended for basic job tracking is selected which includes three service modules: a data collection service module that captures and transmits job tracking events, a state and data storage service module that maintains the jobs' locations, and a user interface service module that provides a tracking dashboard to management. They first consider the state and data storage service module and instantiate it with a ready-made option that listens for tracking events using MQTT and updates an internal state model that can be queried using a REST API. Next, the user interface is similarly instantiated with a ready-made option that will query the state model using its API and display the current job locations. Next the data collection service module is considered. No suitable ready-made options are found so the designer moves to the BB level. They consider templates that offer a variety of approaches and settle for one that uses barcode scanners and output over MQTT. The instantiate the design with ready-made building blocks from the catalogue including a USB barcode scanner. Expansion to further locations can be achieved by adding replicas of the data collection service module or by adding barcode scanners to existing data collection service modules for nearby locations.

4 Evaluation

This section evaluates the proposed design approach. First the identified criteria are considers, then the testing approach is evaluates, finally the outstanding needs are discussed.

4.1 Meeting requirements

Although the design approach is still under development, it is useful to assess to what extent it has met the feature set in 3.1:

- **Variable level of detail** – The design approach can operate at a solution level, a service module level, or a building block level to cater to different abilities.
- **Accommodating of uncertainty** – The design approach operates within an iterative development process to mitigate the risk of uncertainty and provide opportunities for refinement.
- **Procedural (design process)** – The design approach has a clear structure that can be followed end-to-end.
- **Limited / optional iterations (design process)** – The design approach does not have optimisation iterations. Improvements are made in subsequent revisions.
- **Capture and re-use of good design** – The design approach uses templates and catalogues of ready-made elements to capture and re-use good design.
- **Standardised/Common interfaces** – The building block and service module concepts facilitate the standardisation of interfaces.
- **Classification of interchangeable components** – The use classification of building block functionalities and interfaces facilitates the interchange of components.
- **Technology and infrastructure neutral** – The design approach and solution structure do not mandate any specific technologies.

These features contribute to a design that meet the established criteria of being simple, accessible, streamlined and versatile as discussed in section 3.1.

4.2 Testing the approach

The presented approach has been used in five pilots with small manufacturers at the time of writing, with plans for further pilots. The typical progression of these pilots follows three increments: an initial single deployment trial, a small scale trial across a couple of locations, and wide scale deployment in the company; with improvements made after each increment. The approach has also been applied to several demonstrators and student developments. These are noted here as templates and ready-made elements from these developments have been reused in some pilots. This testing has been limited to a small number of solution areas, although it has focussed on solutions identified as priorities for UK SMEs [1, 2, 12]. This testing is evidence that the approach can be used to develop low-cost solutions, however further a/b style testing is being considered to show that it is more effective that without the approach.

4.3 Outstanding needs

The presented approach does not discuss design tools for ensuring completeness or checking compatibility. The approach facilitates these objectives. The cumulative functionalities provided by building blocks and service modules along with whether or not they are all instantiated can be used to determine completeness. Similarly, the well-defined interfaces of these elements can be used to map compatibility. However both of these mechanisms are tedious and well suited to automation by digital tools. This automation is future work, and is dependent on ongoing work looking at a digital environment to support the development (and deployment) of these low-cost solutions [17].

A further target feature that has not yet been considered is that of performance determination or estimation within the context of the presented approach. The existence of such a feature would have a significant impact on the risks of low-cost design being conducted by inexperienced designers, however the practicality or even possibility of such a feature when considering low-cost components is not yet known.

5 Conclusions

In this paper, the design requirements for low cost digital solutions within manufacturing were examined. Alternative solution structures and design procedures were assessed and a design approach was proposed based on a set of features that were identified using the established criteria. The proposed design approach uses a three tiered modular structure to enable designers to operate at different levels of details, to handle the wide variety of low-cost components, and to facilitate interchangeability of those components. The approach is still under development and does not yet implement all the identified features, and future work will add these features as well as expand the catalogues of templates and ready-made elements. In conclusion, even though the presented approach is not yet complete, it already provides a simple, accessible, streamlined and versatile method for implementing low-cost digital solutions.

This work was supported by the Engineering and Physical Sciences Research Council [grant number EPSRC: EP/R032777/1].

6 References

1. McFarlane, D., Ratchev, S., Thorne, A., Parlikad, A.K., De Silva, L., Schönfuß, B., Hawkridge, G., Terrazas, G. and Tlegenov, Y., 2019, October. Digital manufacturing on a shoestring: Low cost digital solutions for SMEs. In *SOHOMA* (pp. 40-51). Springer, Cham.
2. McFarlane, D., Ratchev, S., de Silva, L., Hawkridge, G., Schönfuβ, B., Terrazas, G., 2021. Digitalisation for SME Manufacturers: A Framework and a Low Cost Approach. submitted to SOHOMA 2021.
3. Lu, Y., 2017. "Industry 4.0: A survey on technologies, applications and open research issues". Journal of industrial information integration, 6, pp.1-10.
4. Westerman, G., Bonnet, D. and McAfee, A., 2014. "The nine elements of digital transformation". MIT Sloan Management Review, 55(3), pp.1-6.
5. Geissbauer, R., Vedso, J., Schrauf, S., 2016. "Industry 4.0: Building the digital enterprise", Global Industry 4.0 Survey, PwC
6. R. Davies, 2015. "Industry 4.0: Digitalisation for productivity and growth". European Parliamentary Research Service. Briefing.
7. Horváth, D. and Szabó, R.Z., 2019. Driving forces and barriers of Industry 4.0: Do multinational and small and medium-sized companies have equal opportunities?. Technological Forecasting and Social Change, 146, pp.119-132.
8. Stentoft, J., Adsbøll Wickstrøm,Philipsen, K. and Haug, A., 2020. Drivers and barriers for Industry 4.0 readiness and practice: empirical evidence from small and medium-sized manufacturers. Production Planning & Control, pp.1-18.
9. Mittal, S., Khan, M.A., Romero, D. and Wuest, T., 2018. A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs). *Journal of manufacturing systems*, *49*, pp.194-214.
10. Masood, T. and Sonntag, P., 2020. Industry 4.0: Adoption challenges and benefits for SMEs. Computers in Industry, 121, p.103261.
11. McFarlane, D., Ratchev, S., 2018, *Digital Manufacturing on a Shoestring*, UK Engineering & Physical Sciences Research Council, Research Grant EP/ R032777/1
12. Schönfuß, B., McFarlane, D., Hawkridge, G., Salter, L., Athanassopoulou, N., de Silva, L., 2021. *A Catalogue of Digital Solution Areas for Prioritising the Needs of Manufacturing SMEs.* Computers in industry.
13. De Silva, L., Hawkridge, G., Terrazas, G., Perez Hernandez, M., Thorne, A., McFarlane, D. and Tlegenov, Y., 2020, March. *An Application of HRI in Low-Cost Digital Manufacturing*. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction pp.71-71.
14. Hawkridge, G., Mukherjee, A., McFarlane, D., Tlegenov, Y., Parlikad, A.K., Reyner, N.J. and Thorne, A., 2021. Monitoring on a shoestring: Low cost solutions for digital manufacturing. *Annual Reviews in Control*.
15. Hawkridge, G., McFarlane, D., Mukherjee, A., de Silva, L., Terrazas, G., Ling, Z., 2021. Deploying Low Cost Digital Solutions (manuscript under preparation)
16. Terrazas, G., Hawkridge, G., McNally, M., McFarlane, D., Ling, Z., Lau, J., 2021. Hackathons as a Method to Accelerate the Development of Low Cost Digital Solutions. submitted to SOHOMA 2021.
17. Ling, Z., de Silva, L., Hawkridge, G., McFarlane, D., Martinez-Arellano, G., Schoenfuss, B., Thorne, A., 2021. A Visual Environment for Developing Affordable Digital Manufacturing Solutions for SMEs. submitted to SOHOMA 2021.