Group Report Template

Alex Valentiner *, Max Sandberg Hansen*, Mikkel Priisholm Larsen*, Student 4*, Student 5*,
University of Southern Denmark, SDU Software Engineering, Odense, Denmark
Email: * {alval20, maxha20, mikkl20,student4,student5}@student.sdu.dk

***Abstract*—**

***Index Terms*—Keyword1, Keyword2, Keyword3, Keyword4, Keyword5**

I. Introduction and Motivation

The structure of the paper is as follows. Section II outlines the research question and the research approach. Section III describes similar work in the field and how our contribution fits the field. Section IV-A presents a production reconfiguration use case. The use case serves as input to specify a reconfigurability QA requirement in Section IV-C. Section V introduces the proposed reconfigurable middleware software architecture design. Section VI evaluates the proposed middleware on realistic equipment in the I4.0 lab and analyzes the results against the stated QA requirement.

II. Problem and Approach

Problem.

Research questions:

1)
2)

Approach. The following steps are taken to answer this paper's research questions:

1)

III. Related work

This Section addresses existing contributions by examining xxx in the I4.0 domain. In total, x papers are investigated.

In [1], experiences are elaborated on a three-layer architecture of a reconfigurable smart factory for drug packing in healthcare I4.0.

The paper [2] proposes an ontology agent-based architecture for inferring new configurations to adapt to changes in manufacturing requirements and/or environment.

In [3], [4] an architecture for a reconfigurable production system is specified. Two objectives for reconfiguration and how they can be reached are described.

Several papers [5]–[7] describe reconfigurable manufacturing systems that are cost-effective and responsive to market changes.

All contributions provide valuable knowledge about reconfiguration but lack a study of the software architecture perspective that specifies a quantifiable reconfigurability architectural requirement, a software architecture that adopts the architectural requirements, and evaluates the architectural requirement.

IV. Use Case and Quality Attribute Scenario

Assumption 1: The company is a manufacturing drones. Therefore the company has to develop hardware (drones) and software for the drones.

Assumption 2: The company has multiple departments: Software Development, Part production, Drone assembly, Test, and Shipping.

Assumption 3: The company has a size of 150 workers which are spread out between all the departments.

A. *Use cases - Alex, Max, Mikkel*

This Section introduces the use case and the specified x QASes. The QASes are developed based on the use case.

Use case 1: Add component to drone
Actor: Customer/Owner of a drone
Preconditions: Customer has bought a drone
Steps:

1) Customer attach new hardware to the drone.
2) Customer extends the drones software by creating a new software component to handle this new hardware.
   a) Customer plug the drone into his computer and open the official software provided by us.
   b) Customer selects add new component in the software.
   c) Customer adds the software component that uses the provided API's.
   d) Customer clicks on a button that uploads the new software to the drone.

Postconditions: The drone now has the new software components provided by the user.

Use case 2: New robot
Actor: Robot
Preconditions: The production line needs a new robot to replace an old robot.
Steps:

1) The old robot is removed from the production line
2) The new robot is added to the production line
3) The interface between the robot and the production system needs to be changed
4) The interface between the robot and the monitoring system needs to be updated

Post-conditions: The new robot has added to the production line

Use case 3: New drone type
Actor: Software development, part production, drone assembly, and test department
Preconditions: A new type of drone has been designed
Steps:

1) Departments implements new features specific to the new type of drone
   a) Software department implements the new functionality for the specific drone type.
   b) Part production starts producing the new drone parts
2) Drone assembly adds the instructions to assemble the new drone type, to their catalog of drone assembly instructions.
3) Test department creates new tests for the specific drone type.

Postconditions: The new type of drone can be produced.

Use case 4: Drone software update
Actor: Customer Preconditions: Software department has released a new update Steps:

1) Customer plugs drone into his/her computer and open the official software for the drone.
2) Customer clicks on update drone software.
   a) Drone software is being updated.
3) Customer takes the drone out of the computer after the update.

Postconditions: Drone is updated.

B. Systems and subsystems - (Alex, Max, Mikkel

1) Examine the Context: The software delivery for the drones needs to run 24/7 and needs to be continuously deployable. The software delivery system will be used to store software versions, where users can download new software for their drone based on the drone type and software version wanted in the official application. This means that new versions are continuously deployed to the application ready to be installed by users. *(Deployability and Availability* Great API documentation for the users, so it is possible for users to use these to make compatible components for the software, so that they can modify their drones. This implies a very loose coupling between software components, so that components written in other programming languages are compatible with the standard drone software. New updates for the standard/base software should not ruin the components made by users. This means the use of interfaces will be necessary, so the user do not depend directly on implementations. This is some of the measures to address Quality Attributes such as *Interoperability*, *Deployability*, and *Integrability*.
The production system needs to run 24/7 to continuously produce new drones for the customers. The software for the production system therefore needs to run 24/7 to keep the robots etc working on producing drones. This means that the production and testing of the drones should automated so it can keep running continuously without supervision. The architecture needs to have a *Availability* of 99.0% which means that the system will have a maximum downtime of 3 days and 15 hours per year. To support the 99.0% availability, the architecture will need to address fault detection, a way to recover from the fault, and fault preventions. Since not all faults can be predicted, a way to secure a high availability would be to have multiple producition lines, which can keep producing the drones while the faulty production lines can be addressed.
The robots in the production line/system needs to be to be replaced with another robot with ease. This take as little time as possible to easily exchange one robot with another. This therefore addresses the Quality Attributes *Interoperability* and *Availability*

2) Primary system: Production system
Subsystems:

- Assembly system
- Monitoring system

3) Primary system: Software delivery
Subsystems:

- Version control of drone software
- Software testing
- Software updater

4) Primary system: ERP system (storage system)
Subsystems:

- Data analysis system
- Data processing system
- Monitoring system

C. Quality attribute scenarios

V. The solution

This section will describe a proposed design of that aims to achieve the stated QASes stated in the previous section.

VI. Evaluation

This Section describes the evaluation of the proposed design. Section VI-A introduces the design of the experiment to evaluate the system. Section VI-B identifies the measurements in the system for the experiment. Section VI-C describes the pilot test used to compute the number of replication in the actual evaluation. Section VI-D presents the analysis of the results from the experiment.

A. Experiment design

B. Measurements

C. Pilot test

D. Analysis

VII. Future work

VIII. Conclusion

References

[1] J. Wan, S. Tang, D. Li, M. Imran, C. Zhang, C. Liu, and Z. Pang, "Reconfigurable smart factory for drug packing in healthcare industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 507–516, 2019.

[2] Y. Alsafi and V. Vyatkin, “Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 4, pp. 381–391, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584509001239

[3] P. Leitão, J. Barbosa, A. Pereira, J. Barata, and A. W. Colombo, “Specification of the perform architecture for the seamless production system reconfiguration,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Florence, Italy: IEEE, 2016, pp. 5729–5734.

[4] G. Angione, J. Barbosa, F. Gosewehr, P. Leitão, D. Massa, J. Matos, R. S. Peres, A. D. Rocha, and J. Wermann, “Integration and deployment of a distributed and pluggable industrial architecture for the perform project,” *Procedia Manufacturing*, vol. 11, pp. 896–904, 2017, 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978917304018

[5] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, “Reconfigurable manufacturing systems,” *CIRP Annals*, vol. 48, no. 2, pp. 527–540, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850607632326

[6] Y. Koren and M. Shpitalni, “Design of reconfigurable manufacturing systems,” *Journal of Manufacturing Systems*, vol. 29, no. 4, pp. 130–141, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612511000021

[7] M. Bortolini, F. G. Galizia, and C. Mora, “Reconfigurable manufacturing systems: Literature review and research trend,” *Journal of Manufacturing Systems*, vol. 49, pp. 93–106, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612518303650