Abstract

Increasing product variety and shortening product life cycles call for a fast reconfiguration of production systems. To face these challenges one common solution is the encapsulation of subsystems by creating modules. However, modularization raises the initial costs of the production system which is why the optimal degree of modularization must be determined in order to minimize the life cycle costs. The decision on the modularity of the system has to be taken in the early planning phase although the quality of the system data is poor. A method for estimating and evaluating the life cycle costs of a decentralized component-based automation system is presented in this paper. To establish a solid basis for the evaluation the system is divided into cost packages and an estimation method is proposed in order to obtain reliable data on each cost package. Based on these cost packages, the user of the method can easily build up different life cycle scenarios for the production system. Particular attention within this method is paid to the system availability which is a very important criterion for economic success. The result is a thorough analysis of the life cycle costs in order to take decisions concerning the suitable degree of modularization in the early planning phase of a production system.

Keywords: Life cycle costs, Modularization, System architecture, Automation Component

1. INTRODUCTION

Manufacturing is undergoing a profound change. Turbulence creates market fluctuations, the individualization of products results in a great diversity of products in small lot sizes and the shortening life cycles force manufacturers to react faster to the influences of the environment [1][2]. These are the challenges that need to be faced to keep up with the market requirements of the future.

In order to adapt production systems to these requirements flexibility and reconfigurability have to be improved [3] [4]. This means that changes to the structure of a production system must be executable at a minimum of costs and in shortest implementation time possible to avoid downtimes. Modularization is one approach to increase the reconfigurability and thus to meet these requirements. The concept is based on autonomous modules which can be easily combined with other modules [5].

Modularization reduces the complexity by migrating functionality to modules. This increases the quality of the system because each module is realized by experts on the specific area. The main advantage of modularization is the exchangeability and reusability of the different modules. Lean and intuitive interfaces allow modules to be combined and recombed and thus the functionality of the system can be changed in a quick and easy way.

But modularization also presents disadvantages. The creation, encapsulation and implementation of modules causes an increase of hardware costs compared to a component in a centralized architecture.
Considering the statements on modularization mentioned above, it is necessary to identify a suitable degree of modularity in order to minimize the costs of automation solutions.

2. Architecture of production systems

2.1. Classical centralized control architecture

For a long time automation solutions were defined according to a centralized architecture whose levels are represented in the automation pyramid (Fig 1) [6]. The Enterprise Resource Planning (ERP) and the Manufacturing Execution System (MES) are focused on planning and monitoring production resources. These higher levels of the automation pyramid are beyond the scope of this paper.

The PLC level is the centralized platform to control the components. Automation components are directly wired to the programmable logic controller (PLC) and behave according to the signals sent by the PLC. The software is implemented and executed in the central PLC. This involves a high level of specialization of the monolithic code for the current configuration of the production system. Modifications of the configuration cause adaptations to be made to the central PLC code which demands a big effort and a high degree of complexity. Nonetheless, this architecture has advantages in terms of low component costs as they don’t need embedded controllers. Furthermore, the coordination of the components is simplified due to a clear structure.

2.2. Decentralized control architecture

With advances in micro technology more and more embedded intelligence has been integrated into components. This development has meant that components can provide control features, thus blurring the distinction between the PLC and the component level. (Fig 2)

The migration of functionality towards the components allows a new kind of communication. Instead of sending detailed single signals to the components, the controller can now hand over tasks which are executed autonomously by the component controlled by embedded intelligence.

This architecture also allows a new kind of engineering which is based on the skills of the automation components.

2.3. Choice of a suitable control architecture

As both architectures provide advantages the production planner has to decide on this issue in dependency of the application. This decision can be taken independently for each component. This means that components in centralized and decentralized architecture can be used in one production system. Thereby the degree of modularity is not one fix value for the entire system but it can be chosen individually for each component.

The decision on the architecture of the component is driven by the component costs during the life cycle. In this paper a cost estimation method is presented to evaluate the life cycle cost of automation components and to take a decision on the suitable architecture for the component under consideration.

Modularization can be executed at different levels, from the basic components up to subsystems or even whole production cells which are encapsulated in a module.

The modularized and task-oriented approach of engineering reduces the effort for wiring as well as for implementing the functionality during development of the automation system. The disadvantage of this approach is the high costs for the embedded controllers in the distributed architecture and the effort required for encapsulating and implementing the modules.
3. State of the art

Literature describes lots of methods for estimating the costs of a product. Niazi et al. [7] give a detailed review of the state of the art distinguishing between qualitative and quantitative methods.

However, the demand for flexibility and reconfigurability of production systems leads to the need for a life cycle cost estimation for automation components [8].

Thyssen et al. [9] describe the calculation of the costs of modularity with activity-based costing. This method takes into account the costs for each phase and activity in the product life cycle to obtain the overall costs. Zaehef al [10] focus on the costs of modularity during the life cycle using real options and statistical methods. Real options are used to model the uncertainties in the prediction of future production that the planner has to face. Deif and ElMaraghy [11] concentrate on the reconfiguration process in their paper. An economic evaluation is presented of the process modifying the configuration of the production system.

These methods provide solutions which cannot directly be applied to the estimation of automation components. Furthermore they focus on modeling and evaluating the life cycle of the production system but they don’t focus on obtaining the data used in the methods.

When gathering data, the estimation of software costs is an especially critical point. Therefore several methods for analysis are described that provide options for measuring the effort involved in software projects. Boehm et al. [12] provide an overview of software estimation methods. A distinction between model-based, expertise-based, learning-oriented, dynamics-based and regression-based techniques is introduced. These methods are specific to complex software projects in an object-oriented pattern. However, the differences in software between object-orientation and modularization don’t allow the application of the mentioned methods for estimating the software costs of automation components [13].

In summary it can be stated that there is concept to evaluate the costs of automation components over the life cycle in a modular architecture.

4. Component-based estimation method

The purpose of the presented method is to take profound decisions on the architecture of automation components. During the early phases of development, most of the features and costs of the automation solution are fixed although the quality of system information is poor [14]. That is why the production planner needs methodic support to get reliable data on the system which he can base his decision on.

In this paper a method is presented to evaluate the costs of a component during the life cycle. This evaluation can be performed on different degrees of modularity in order to examine the effects of architecture over the life cycle.

For reaching this objective the presented method is divided into two steps. The first step includes the estimation of costs during the life cycle. Therefore generally applicable cost packages are defined (4.1) and an estimation method is introduced to obtain reliable data on these cost packages. (4.2)

The second step, the evaluation of the life cycle, is presented in chapter 5. Deriving from the cost packages different scenarios are built up (5.1). Furthermore, the influence of the architecture on the availability is examined (5.2) in order to get a thorough analysis on the life cycle costs of the component (5.3).

![Fig 3 structure of the method](image)

### 4.1. Decomposition of the life cycle

In order to analyze a component it is necessary to divide the life cycle into smaller sections. The idea is to identify reusable cost packages that occur multiple times over the life cycle. Based on these cost packages the user can build up scenarios to describe the life cycle.

These cost packages are:

- Hardware costs
- Physical attachment
- Infrastructure
- Software
- Commissioning

The cost package Hardware costs contains all those parts that have to be bought in order to operate the component, including wiring, connectors, etc. Physical attachment includes the working hours needed to attach the component to the surroundings. This task consists of mechanically mounting and wiring the component. It also includes the documentation about the wiring and the management of wiring errors. This cost package is completed if all connections are attached and documented properly. The cost package Infrastructure contains all the installations around the component that are necessary for the operation of the component, for example the provision of compressed air or the bus communication. This cost package models the relationships of the component that are necessary to build larger systems based on the components. The implementation of the PLC code is estimated within the cost package Software. Both the functionality of the component and the logical attachment to the surroundings must be implemented. For modularized systems the interfaces are particularly important. The cost package Commissioning describes the final step in the development of the automation solution. Parameterization of the components, teaching of positions, unit tests, system tests and dealing with errors that occur during the tests are included in this cost package.
In order to obtain a high-quality and reliable estimation of each cost package there is a subordinate structure splitting the cost package into smaller parts (Table 1).

### 4.2. Estimation procedure

The next step of the method involves a valuation of each part in order to obtain the value of the cost packages. Depending on the availability of information and the simplicity of the estimation, the user can choose to insert the values in the table either as working hours or as the total costs for this part. As the cost package needs to reflect a monetary value there is a transformation between working hours and costs via the hourly wage rate of the personnel executing the task. The model takes into account the different hourly rates according to the type of the task to be executed and the qualifications required by those executing the task.

The figures inserted in the tables for the cost packages are the basis for the evaluation of the entire production system. That is why a close look needs to be taken at how these values are gathered as the quality of the estimation determines the quality of the life cycle evaluation. Concerning the values that are necessary for the estimation, there are three different types of data:

- **Facts**
- **Estimated values**
- **Application parameters**

The category **Facts** includes all the figures that can be determined objectively and definitely. This category is applicable for example for the hardware costs of a component as they can be looked up in the product catalogue of the manufacturer. These figures are not subject to considerable uncertainty.

The data type **Estimated values** contains all aspects that cannot be directly derived from a database. Especially the working hours required to execute a certain task can be allocated to this category. These figures are subject to fluctuations and subjectivity of the validation. There is therefore no formal way to calculate the working hours or the costs of these tasks. The estimation by experts is the only way to obtain estimations for these figures. However, there is a variation in the estimations by different experts depending on their experience and opinions. In order to increase the quality of the estimation a group of experts has to be consulted in order to statistically balance out the variations between the different opinions of the experts.

| Single costs in this cost package | Time[h] | Cost[€] |
|-----------------------------------|---------|---------|
| Mechanical attachment             | 1       | 90      |
| Wiring the component (power supply, signals, pneumatics, hydraulic, …) | 2.5     | 250     |
| Documentation of the wiring       | 1       | 100     |

**Physical Attachment** 4,5 440

In the first step of the estimation process each expert inserts his estimated values into a structured table for the cost packages described above. After the evaluation and presentation of the results, a meeting with all experts participating in the estimation process is arranged. During the discussion the differences in the estimation can be explained in order to obtain a final consensus between all experts, ensuring a high-quality estimation of the values.

Furthermore, empirical data can be taken into account to improve the estimation process. By comparing the estimation to values from similar systems the experts can verify their opinion and optimize the results to obtain an estimation which is as objective and reliable as possible.

The third type of data is the **Application parameters**. These values are used to describe the application and the different scenarios. For example, hourly cost of a system shutdown is an application parameter. According to the type of the application parameters these values don’t need to be determined but they can be used as variation parameters in the final evaluation which will be presented more precisely in the next step.

### 5. Life cycle evaluation of the production system

Based on the cost packages determined in the previous section, the life cycle can be created and evaluated. The idea is to create scenarios representing different occurrences of a component during its life time. By combining these scenarios according to the assumed application of the production system the life cycle costs can be determined.

#### 5.1. Scenarios during the life cycle

The typical scenarios in the life time of a component are analyzed in this paragraph. The cost estimation of each scenario is based on the cost packages described in the previous section. In the presented method four different scenarios are taken into account:

- Initial ramp up
- Failure of the component
- Change of functionality
- Reuse of the component

The **Initial ramp up** represents the initial construction of a component based production system. The engineering tasks must be executed from the early planning phase until the system is operational.

The **Failure of the component** is a very critical occurrence as it can result in huge downtime costs that need to be considered in the life cycle analysis. The time and effort to repair and restart the production system are examined in this scenario. The likelihood of this scenario depends on the availability of the component.

The scenario **Change of functionality** is applicable to functional modifications of the production system. No physical modifications are executed but the software is adapted to create a modified functionality of the production system. This can be achieved by changing parameters or target positions, for example.
The **Reuse of the component** is the final scenario. This means that the component is not needed anymore in the initial system and can be reused in another configuration.

The costs for each scenario can be calculated based on the cost packages that were presented previously (Fig 4). Depending on the type of scenario, different cost packages need to be considered to obtain the costs of the scenario.

Depending on the type of scenario, different cost packages need to be considered to obtain the costs of the scenario.

![Fig 4 Building up the scenarios based on the cost packages that were determined during the estimation](image)

In Table 2 the structure of the scenario Reuse of the component is presented. The main parts of the evaluation are based on the cost packages defined in the previous section. In order to adapt the calculation to the specific type of scenario, specialized tasks are added to the calculation. These tasks need to be estimated separately. Totaling all items results in the costs for the Reuse of the component under consideration.

| Single costs in this scenario | Time [h] | Costs [€] |
|------------------------------|----------|-----------|
| Separating the connectors    | 0.5      | 40        |
| Dismounting                  | 1        | 80        |
| Check of functionality       |          | 50        |
| Physical attachment          |          | 440       |
| Software                     |          | 700       |
| Commissioning                |          | 350       |

| Reuse of the component       | 1660     |

Having calculated the costs for the different scenarios, the next step is to calculate the life cycle costs.

Three of the scenarios can be determined by the application in the production system: initial ramp up, change of functionality and reuse of the component. The frequency of these scenarios is subject to planning by the manufacturer and can be introduced into the calculation as a parameter.

By contrast, the number of occurrences of the scenario Reuse of the component cannot be considered as a parameter chosen by the user of the method. Instead the availability of the system has to be investigated in order to obtain information on the frequency of this scenario.

5.2. Availability

The argument about the benefit of different architectures cannot be included without discussion of system availability. It is impossible to realize a modularized solution on the shop floor if the system availability is worse than that of the standard centralized automation solution even if there are advantages in terms of flexibility and reconfiguration.

To describe the availability of a system the **Mean Time Between Failures (MTBF)** and the **Mean Time To Repair (MTTR)** need to be evaluated.

MTBF describes the time between two failures when the system runs properly. MTTR measures the downtime of the system after a failure until it is restarted. Availability is commonly defined using the following formula: [15]

\[
\text{Availability} = \frac{\text{MTBF}}{\text{MTTR} + \text{MTBF}}
\]

In line with formula (2), either the MTBF has to be increased or the MTTR has to be decreased in order to improve the system availability.

Considering a component in a modular architecture different effects can be derived.

On the one hand the probability of a failure of the component increases with the addition of a further device, the local control unit. On the other hand the local control unit can monitor the component improving the maintenance cycles and reducing the risk of component failures. That is why no clear trend can be given on the MTBF of modular components.

The MTTR decreases in a modular architecture as the modularity allows the fast exchange of a component after a failure.

All in all, it must be guaranteed that the availability of the distributed modular system is not worse than the state-of-the-art centralized system to achieve a sustainable economic success.

5.3. Life cycle analysis

The final evaluation of the method presented in this paper concerns the life cycle costs of different system configurations. Therefore the different scenarios presented above are combined to obtain the lifetime costs of the system (Fig 5).

![Fig 5 Evaluation of the life cycle costs based on the scenarios](image)
The probability of failures needs to be calculated according to the availability approach. The other three scenarios can be combined according to the boundary conditions and the application of the component. By varying these parameters the user can easily study the influence of different scenarios on the life cycle costs.

Table 3 Life cycle costs of the component under consideration

| No | Scenario                  | Scenario costs [€] | Sum[€] |
|----|---------------------------|--------------------|--------|
| 1  | First ramp up             | 2300               | 2300   |
| 2  | Failure                   | 1880               | 3760   |
| 1  | Change of functionality   | 560                | 560    |
| 3  | Reuse of the component    | 1660               | 4980   |
|    | **Life cycle costs**      |                    | **11600** |

As shown in the example (Table 3), it is very easy to calculate the costs for different applications. During the early development phases the exact usage of a component cannot be predicted precisely but the method presented makes it easy to evaluate different scenarios in order to take a well-informed decision on the suitable architecture of the automation component.

6. Summary and outlook

In this paper a method for estimating and evaluating life cycle costs of decentralized component-based automation solutions is presented. Based on the estimation of cost packages scenarios can be created that occur during the lifetime of a component. The final evaluation combines the different scenarios and illustrates the influences of the application on the examined configuration. The result is a life cycle cost calculation supporting the decision on the degree of modularity for automation components in the early phase of development.

The scenarios presented in this paper point at examining a suitable degree of modularization. In case of other objectives, the portfolio of scenarios can be adapted in order to take into account other influences that might be important for the examination. For example, the energy consumption over the life cycle can be introduced as a scenario in order for this aspect to be evaluated.

7. References

[1] ElMaraghy, H.A., 2005. Flexible and reconfigurable manufacturing systems paradigms. International Journal of Flexible Manufacturing Systems 17 (4), 261–276.
[2] Wiendahl, H.-P., Heger, C.L., 2004. Justifying changeability. A methodical approach to achieving cost effectiveness. Journal for Manufacturing Science and Production 6 (1-2), 33–40.
[3] Jovane, F., Westkämper, E., Williams, D.J., 2009. The ManuFuture road: towards competitive and sustainable high-adding-value manufacturing. Springer.
[4] Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N., Briese, M., 2007. Changeable manufacturing-classification, design and operation. CIRP Annals-Manufacturing Technology 56 (2), 783–809.
[5] Parnas, D., Clements, P., Weiss, D., 1985. The Modular Structure of Complex Systems. IEEE Trans. Software Eng. SE-11 (3), 259–266.
[6] British Standards Institution, 2001. IEC 62264-1. Enterprise-control system integration. Part 1: Models and terminology.
[7] Niazi, A., Dai, J.S., Balabani, S., Seneviratne, L., 2005. Product Cost Estimation: Technique Classification and Methodology Review. Journal of Manufacturing Science and Engineering 128 (2), 563–575.
[8] Gershenson, J.K., Prasad, G.J., Allamnneni, S., 1999. Modular product design: a life-cycle view. Journal of Integrated Design and Process Science 3 (4), 13–26.
[9] Thyssen, J., Israelsen, P., Jørgensen, B., 2006. Activity-based costing as a method for assessing the economics of modularization—A case study and beyond. International journal of production economics 103 (1), 252–270.
[10] Zäh, M.F., Sudhoff, W., Möller, N., Aull, F., 2004. Evaluation of mobile production scenarios based on the Real Option Approach. 4. Chemnitzer Tagung’ Vernetzt Planen und Produzieren’(VPP). Chemnitz 27, 28.
[11] Deif, A.M., ElMaraghy, W., 2006. Effect of reconfiguration costs on planning for capacity scalability in reconfigurable manufacturing systems. International Journal of Flexible Manufacturing Systems 18 (3), 225–238.
[12] Boehm, B., Abts, C., Chulani, S., 2000. Software development cost estimation approaches—A survey. Annals of Software Engineering 10 (1-4), 177–205.
[13] Szyperski, C., 2002. Component software: beyond object-oriented programming. Pearson Education.
[14] Ehrlenspiel, K., Kiewert, A., Lindemann, U., 2007. Cost-efficient design. Springer.
[15] Vargas, E., 2000. High availability fundamentals. Sun Blueprints series.