DECENTRALIZED HANDLING OF CONFLICTS IN MULTI-BRAND ENGINEERING CHANGE MANAGEMENT

Hesselmann, Carsten (1); Kehl, Stefan (1); Stiefel, Patrick (2); Müller, Jörg (1)
1: Clausthal University of Technology, Germany; 2: Volkswagen Group, Germany

Abstract

Changes on components that are commonly used in different products in multi-brand product development, cause various types of conflicts. This paper considers conflicts of interest among individuals that are caused by differing payoffs each brand would realize if a specific change request were accepted. Following an analysis of requirements on multi-brand engineering change management processes, we propose an approach to model these conflicts using methods from the field of agreement technologies. In particular, we propose voting games as a model and mechanism to represent and reconcile opposing interests and conflicting requirements. In this initial paper, we describe such a mechanism from a conceptual perspective and illustrate our model by using an example agreement process, where commonly used components are defined by different brands and changes on them are voted on in order to determine the set of components that combines the different interests at best. A core element of our approach is a micro-meso-macro coordination architecture that allows to express and reconcile different value systems (e.g. from corporate group, from different brands, down to individual designers).

Keywords: Decision making, Platform strategies, Requirements, Agreement technologies, Engineering change management

Contact:
Stefan Kehl
Clausthal University of Technology
Informatics
Germany
stefan.kehl@tu-clausthal.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 4: Design Methods and Tools, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

Industrial product and process complexity is increasing (Durão et al., 2015) due to rising expectations from customers and the resulting higher variety (Kehl et al., 2016). A good example for this development is the automotive industry. To meet customer requirements, new products consistently include new functions (e.g. driver assistance system) and a wider range of variety (customizability). This trend leads to a higher number of components per product and therefore to higher complexity of the overall product (Cheng and Chu, 2012). (In this paper, by *component* we denote a single part, an assembly or combination of those; each component consists of several elements (BOM nodes, geometries, etc.) as defined e.g. in (Kehl et al., 2016)).

It is a company's goal to meet the rising consumer expectations and individual needs by offering a large (customizable) variety within their product families while maintaining (nearly) mass production effectiveness (Tseng and Jiao, 2007). The downside of a large variety is that it leads to a high number of dependencies between individual components. In a corporate group, where different brands share the same components, these dependencies multiply according to the number of products and brands (with potential conflictive goals).

Considering that product development is not a linear process, but a continuous sequence of changes on existing components (Bucciarelli, 1994; McMahon, 1994; Cross, 2008; Cheng and Chu, 2012; Kehl et al., 2015), each *change* on a component may lead to changes on other components (even in other products of different brands). (Cheng and Chu, 2012) refer to this as *avalanches* of changes.

To maintain an effective product development process in a cooperate group, it is necessary to decide which changes (proposed by different brands) will be accepted or declined. Solid decision making regarding upcoming changes includes: (i) deciding which of alternate changes to realise and (ii) keeping track of dependencies so that resulting changes are predictable.

(Clarkson et al., 2004) address this issue by predicting and managing change propagations to an existing product caused by new requirements or from faults. Changes are predicted with the help of a product risk matrix on a case study of rotorcraft design for helicopters. The dimensions of the matrix are the product (with dependency matrices and predictive matrices) and the change requirement (with the identification of initial and predicted changes).

(Cheng and Chu, 2012) present three indices for changeability in products with weighted networks theory. Connections between components are analysed to model the product network. Subsequently the assessment of change propagations is developed based on weighted networks theory, where a component's coupling strength (to other components) defines the component's change impact.

Creating and maintaining product data (especially if changes occur) in multiple-brand product development is a largely decentralized process. Therefore (Bender et al., 2015) propose different agent-based architectures to support the decentralized stakeholders by (partly) automating the creation and alignment of logistical and geometric information of shared components.

The above authors analyse change operations and how their consequences can be predicted, but they do not consider social aspects (such as brand-specific design or customer demands) regarding the payoff each brand would achieve if a specific change were realized. The approach presented in this paper takes into account both the differing interests of different brands and their resulting requirements and the exploit of cost saving potentials across the group (e.g. via the usage of platforms, i.e. a collection of components used in different products by multiple brands) aiming to keep multiple change requests consistent. In practice, initial changes on platform components often lead to additional components for the proposing brands, diluting the platform and causing unsatisfactory *usage*. The platform usage is decreasing along the product development process as change requests are not kept consistent with the platform's components. We expect that engineering costs across the group can be reduced, because conflicts are truly solved by combining the different interests at best and the use of defined platform components is more stable.

In this concept paper, we introduce an initial approach to support the process of platform definition and agreeing on platform components based on agreement technologies (Ossowski, 2013). Agreement Technologies (also sometimes referred to as Computational Social Choice (Brandt et al., 2016)) denote models, mechanisms, and tools for algorithmic collective decision-making, including auctions, voting, and games (Nisan, 2008). By having the brands vote over proposed components while rules and regulations are specified by the corporate group, we aim to reconcile both the brands' and the corporate group's

goals to preserve a high percentage of shared components. According to (Hamraz et al., 2013) this game theoretical approach is a pre-change step in the field of engineering change management to reduce the overall number of changes.
As mentioned above, the "agreement technologies toolbox" offers a wide arsenal of methods for collective decision-making; in this paper, we study voting to determine a platform scope as the set of components that best (according to pre-defined utility functions) combines the different interests of all players (group and individual brands). Clearly, such an agreement will involve negotiation, compromise, and pareto-optimality.
The structure of this paper is as follows: first, we describe a corporate group and its brands (focusing on elements related to product development), as socio-technical systems and model the occurring conflicts in a three-level architecture. Afterwards we propose a voting game model (embedded in the three-level architecture), for defining and coordinating a platform, and we provide and illustrate a specific example scenario to analyse the scope of options in the multi-brand platform game. We conclude the paper with a summary and future work.

2 MULTI-BRAND CHANGE MANAGEMENT: A SOCIO-TECHNICAL SYSTEM

A corporate group in the automotive industry and its brands can be conceived as socio-technical systems since according to (Saurin and Gonzalez, 2013); they feature (i) a large number of elements which interact dynamically, (ii) a wide diversity among these elements according to multiple criteria, (iii) an unanticipated variability due to product variety or randomness and (iv) the system's resilience, which describes the sustainability under expected and unexpected conditions.
In addition, some elements of a socio-technical system will form groups (organizations) that correspond to a set of common goals. In the given scenario, such a group can relate to a component, assembly, platform or the like where every individual involved in its development is part of the same group. At the same time, due to stakeholders' diversity, conflicts of interests (such as incompatible change requests) are likely to occur. Such can be among any group and between multiple stakeholders.
In the following section, we describe the concept of a three-level architecture, originally proposed by (Sanderson, 2013), and transfer it into the multi-brand change management domain to model the resulting socio-technical system with the occurring conflicts.

2.1 3M: micro-meso-macro architecture for socio-technical systems

The 3M architecture consists of three hierarchically structured levels (see Figure 1 for an illustration with examples for micro and meso elements in the context of a company in the automotive industry):

- **Micro level**: This is the lowest level of the architecture. Every individual belongs to this level. Individuals are heterogeneous and differ by e.g. attributes, type, and workspace. Individuals have their own preference structures and objectives (captured e.g. by payoff functions).
- **Meso level**: The individuals belonging to the micro level form groups on the meso level. Thereby it is possible for them to organise themselves according to e.g. a common goal. Furthermore, an individual is not limited to being in one group only.

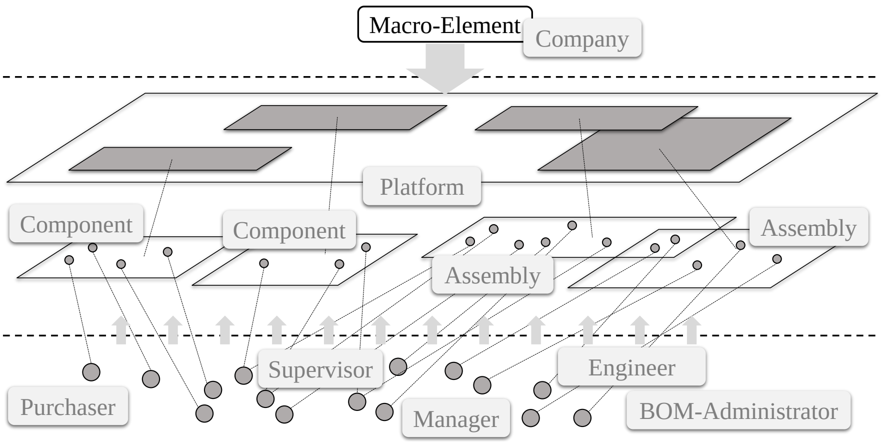

Figure 1. Micro-meso-macro architecture for an automotive company

- **Macro level**: This level represents the superordinated governance or "societal" perspective. It defines appropriate properties, value systems, and payoff functions to define system optima.

Both micro and macro level influence the groups on the meso level. This is due to individual goals, which each member of the micro level wants to achieve, and contrary to that the macro level by defining rules, which may encourage or inhibit the clusters on the meso level (Sanderson, 2013). This architecture is useful to relate the correlations taking effect in causing conflicts and to originate the opposing interests.

2.2 Three level architecture in the context of brands and corporate group

Considering the corporate group as well leads to a different architecture. Due to the reason that a corporate group is hierarchical above all companies, the companies are on the micro level in this architecture. Figure 2 shows the correlation between the corporate group and the companies. Here the corporate group is on the macro level whereas the elements on the micro level represent a company from Figure 1. The groups at the meso level can represent a joint work on components, assemblies, whole products or a platform, which is used for a specific vehicle class.
Moreover, in this platform conflicts of interest are likely to occur. This can be caused by multiple reasons like (i) one company wants to make changes on a platform component which would cause conflicts for other companies' products, (ii) a change on a component is requested because it is incompatible with components from outside the platform for a company, or (iii) multiple companies prefer to change a platform component but the changes are inconsistent (due to changed material or geometry). Reasons like these can result of requirements for cost savings, performance, used material or different target groups the companies want to address with their products that base on the same platform.

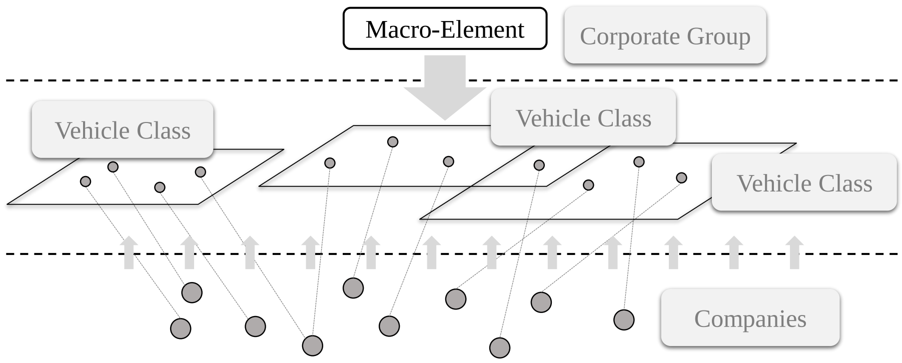

Figure 2. Micro-meso-macro architecture for a corporate group

We present an approach based on agreement technologies, i.e. game theory and voting, in the following section to address the described conflicts occurring while defining and coordinating a platform.

3 A MULTI-BRAND PLATFORM GAME

In the initial situation, there are multiple brands and the game is about finding an agreement on components that are used by all brands. The game models the coordination of a platform whereas the voting mechanism is used to choose one of multiple, alternative components (or change requests on a component) as part of the platform. The game ends when an agreement on the platform is found, in other words when a vote over the defined scope of components was successful. In reference to the architecture, one game represents an element of the meso level in Figure 2. This meso element is referred to as one game instance.
Before defining the voting protocol, the necessary background for game theory and the game elements are applied to the scenario of multi-brand product development. In game theory, a game consists of (i) a set of players, (ii) a set of possible strategies, which consist of actions a player can perform, (iii) a function determining the utility/payoff for each player, and (iv) a set of rules defining how the game is played (Bartholomae and Wiens, 2016, p.31-55). Furthermore, a game is classified by the sequence in which the game is played and what information is available for the players in which state of the game. In the context of multi-brand product development, the players are the involved companies.

3.1 Utility function and strategic goal weight

When conceptualizing a game, the main question to consider is: What is the strategic goal a player wants to achieve (Bartholomae and Wiens, 2016)? Here each player (company) tries to maximise the payoff from using the components, which they are voting on. For that reason, the utility function needs to determine the payoff of every component for each player. One way to do this is to base the utility function on the three following aspects:

- **Characteristics** which sum up the range of requirements.
- **Weights** for every brand for each characteristic to determine how important this requirement is. This reflects the requirements and preferences (for each brand) in reference to the products in which the component shall be used.
- **Ratings** for every component according to the given characteristics. By combining ratings and weights, the payoff can be determined for each player.

From the micro perspective, the strategic goal is to maximise the payoff of all components used. Whereas from the macro perspective the goal is to maximise the overall payoff, which is for example definable as: (i) the average of all payoffs, (ii) the maximisation of the smallest payoff, or (iii) the sum of every individual payoff. Depending on the definition of overall payoff, the preferred result from the macro perspective can be different from what the players prefer. The rules (that define how the game is played) enable the macro element to influence the game's process to achieve the maximisation of the overall payoff.

Thereby each game instance can have a different set of rules that can be used to enforce the strategic goal of the macro element. With this, the scale of diverse micro elements for each group at the meso level (representing a vehicle class using one common platform) is incorporated. Rules give many possibilities to influence the game's procedure and the players' behaviour. How the players can achieve their strategic goal is a question of available actions, which are discussed in the following section.

3.2 Scope of actions and roles

The players are influencing the component's development by proposing changes according to their individual preferences and requirements. A reaction to that is to accept/decline a request or start negotiating about the change and offer a compensation if a change causes payoff losses. So the strategic goal is achievable by making sure that the developed components are designed in a way that they fulfil the own requirements as good as possible.

In the game, the micro elements are homogeneous players and have the same roles and rights. In total, there are four different roles within the game. These are voter, proposer, chair and monitor as defined in (Pitt, 2005). Being a voter enables a player to take part in a vote. The role proposer can make propositions for a vote. The chair supervises the vote and the monitor is responsible for counting the votes and making sure all votes are cast correctly (Pitt, 2005).

Table 1. Types of players and assigned roles

Type of Element	Roles in the game	Roles given in scenario
Micro element	Voter and proposer	Company
Macro element	Chair and monitor	Corporative group

Table 1 displays how the roles are assigned in relation to the elements in Figure 2. It enables the collaboration partners to propose changes and at the same time take part in the vote. With this, the proposition of components (or changes on components) becomes an interactive process. Thereby the macro element, representing a corporate group, is responsible for determining each vote's outcome as monitor and has the chair. Latter opens the possibility to take influence on the vote if necessary by adjusting the vote in case the vote is stuck and no majority exists. With this, the chance all players or a majority agrees on one proposal is increased. How such adjustments can look like is discussed in Section 3.3.

Before the game can begin, following terms have to be met and information has to be on hand: (i) the roles are assigned, (ii) an initial set of components which are voted on and the dependencies between them are at hand, (iii) the characteristics which are used to evaluate the components are defined, (iv) each player has given a weight for every characteristic, (v) all change propositions by the players have been made, (vi) the propositions are evaluated according to the specified characteristics and (vii) the voting protocol is defined.

In the next section, we exemplarily describe the aspects of the voting protocol before illustrating the game's sequence and rules on basis of one instance of the game.

3.3 Voting protocol

The voting protocol determines the vote's exact sequence and defines when and in what way adjustments are possible for the chair. Thereby the voting protocol includes the following aspects:

- The order in which the components are voted on,
- whether votes are hold parallel or sequential,
- whether the set of components is fix or variable,
- how many times one component can be voted on,
- how an agreement is defined,
- the set of possible adjustments if no agreement is found and
- if special rights (like veto rights) are permitted.

The order: Because a component can be related to one another it is possible to visualise these relations as groups (e.g. an assembly) in a tree structure (Figure 3). In addition, this representation gives an overview of all components being part of the game and displays the current state of the game by reflecting the finished votes. Here a depth-first search defines the order in which the components are voted on. This incorporates the difficulty of voting over technically related components. In Figure 3 the players have voted over two independent components and one with technical dependencies. The next votes have to be over the three remaining components of that assembly before continuing with the rest.
Parallel or sequential: In relation to the voting order the question whether votes are hold parallel or sequential has to be answered. Parallel voting gives the possibility to accelerate the whole process whereat sequential votes are better at incorporating the dependencies between different components. This is important because a vote's outcome (previous round of the voting game) can influence the next vote (round) (Wooldridge, 2009, p.240). For this reason, sequential voting is used here. It gives the players the opportunity to consider prior outcomes when casting votes. The components' representation in groups in Figure 3 illustrates the described issue.
Set of components: When the number of components is defined as variable, it enables the players to add new ones during the game. Firstly, this means this component is meant to be developed together or used by all collaboration partners. Secondly, the proposed component has to be evaluated according to the defined characteristics. With that, the players can determine the payoff from using this component, compare it to the one currently used and make change propositions. Lastly, the component needs to be included in the voting order. Due to reasons of simplicity the number of components will be set to invariable as this is a first draft of a voting protocol which can later be extended in terms of complexity. A variable number could not only extend the game several times by adding new votes but also put the game on hold as the component needs to be evaluated and identifying possible change propositions.

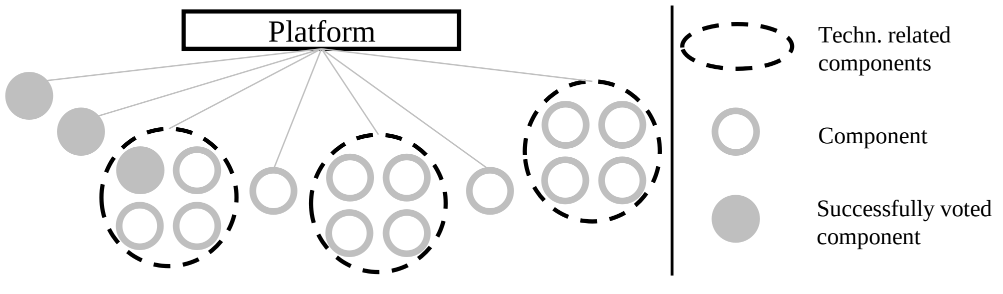

Figure 3. Relations between components as tree structure

Number of votes on one component: It is possible that the vote continues for a long time if there is no winner and the number of voting rounds is not limited. This can be handled in two possible ways. The first option is to set a maximum number of voting rounds. Thus, the vote terminates in either case but the outcome is uncertain. However, even after reaching the maximum number of rounds there might still be no winner. The second option is to make vote adjustments to increase the chance the players agree on one proposal:
Vote agreement and adjustments: Only the role chair can make adjustments. There are different ways on how an adjustment can be realised. Despite the differences, they are all made for the same reason; to

increase the chance the players find an agreement. This can be for example a unanimous vote or an absolute majority. In order to make adjustments work the players need to have an incentive to change their decision or make a different proposal in the first place. To achieve this it is necessary to affect the player's payoff for it is the main decision criterion in the game. For this, three alternatives are described here:

- **Compensation:** Offering a compensation to the players, who rejected a proposal, which was accepted by others. If the compensation is high enough it can make the rejecting players change their vote and accept the proposal. This can only work if the offered payoff is at least big enough to balance the losses that were the reason for the rejection. Taking reference to the given scenario such a compensation can be of financial kind.
- **Exchange:** Involving the players, who gain payoff from a proposal, in the mentioned compensation. In this case, the compensation is not carried solely by the company or corporate group, which has the role chair and is represented by the macro element, but also by some of the other players. This can be described as an exchange where players who benefit from using the chosen proposal compensate the losses of others. Therefore, players might give their approval despite the loss of payoff. In addition, players might change their proposals in a way that they cause less payoff losses. In that case, less compensation is necessary. This can be accomplished by linking the compensation's extend with the payoff losses caused by the proposal.
- **Majority change:** If no such adjustment leads to a valid outcome, the definition of agreement can be adjusted as well. For example, by changing a vote's winning condition from an absolute majority to a simple majority. This does not take influence on the players' behaviour like a compensation but simplifies the vote's termination condition. Doing so can lead to an agreement without investing time and money on negotiations.

Special rights: In voting procedures veto rights or abstention are a valid way to enhance a player's possibilities within the vote. For this first version of a voting protocol, those special rights are not taking into account. The reason for this is that in the given scenario abstention is no valid option for a collaboration partner. As for veto rights, every partner is equal wherefore they are not included here. However, a later version might include these aspects.
In summary, the protocol covers these aspects as shown in Table 2.

Table 2. Parameter of the voting protocol

Voting order	deep-first; breadth-first; random	The order is defined as a depth-first search whereat components are grouped if there are technical dependencies between them. Once a vote on a group-component was cast, the group has to be completed before continuing on other components.
Sequence	parallel; sequential	Votes are hold sequential to incorporate prior outcome for future votes.
Set of components	variable; fix	Is defined as fix, which means no components can be added during the game. The players can only propose changes on components that are given at the start of the game.
Number of votes	unlimited; limited	Each vote can have a maximum of two rounds. If no agreement was found in the first, the last round gives the possibility to find an agreement via adjustments.
Agreement	unanimous; absolute maj.; simple maj.	The vote was successful if all players agree on one proposition.
Adjustment	compensation; exchange; maj. change	If players vote against a proposition due to losses in payoff, their loss will be compensated by the vote's chair.
Special rights	veto; abstention; exit	No veto rights are intended. The same goes for exiting the game.

3.4 Exemplary game instance

With the voting game defined a possible sequence for the vote of one component is described in the following: Firstly, the roles are assigned to the all players. In addition, each player defines the own preferences (shown as an exemplary four-dimensional matrix for each player) with which propositions are evaluated. In the next step, the players have the possibility to make propositions for the component, which is voted on. In this example player 3 makes the first proposition. The players evaluate this proposition according their individual preferences. Figure 4 displays this situation and the resulting payoffs each player expects after evaluating the proposal.

Player 2 makes the second proposition, which has different specifications. For this reason, every player evaluates a different payoff (Figure 5 shows the payoffs for both propositions). Based on these evaluations the players decide for which component they want to vote. As there are no further propositions the voting can commence and results in a 3 to 1 vote as proposition 1 is favoured by three players and proposition 2 by one player.

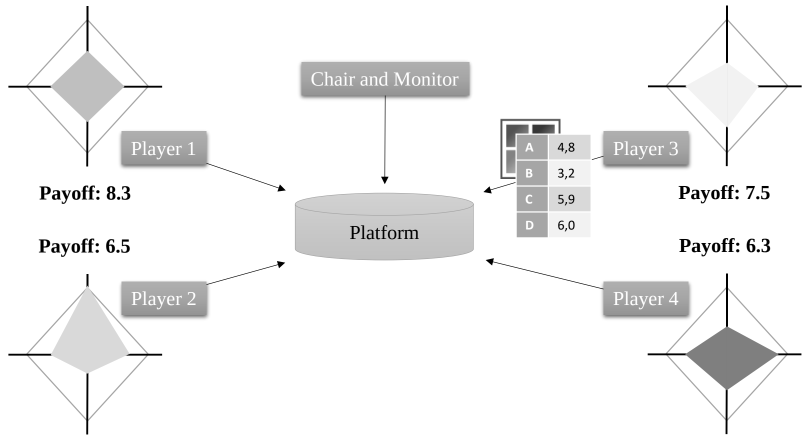

Figure 4. Every player evaluates the proposal

Since the result is not unanimous, the vote's chair can make adjustments. In this example, the chair chooses the *exchange* adjustment. Each player has to give some of the gained value from choosing the first proposition in form of money to the chair. This reduces the payoff for each player who favours proposition 1 but only to a degree where it remains beneficial. The vote's chair then compensates the loss the player 2 would have if the first proposition were chosen (Figure 6).

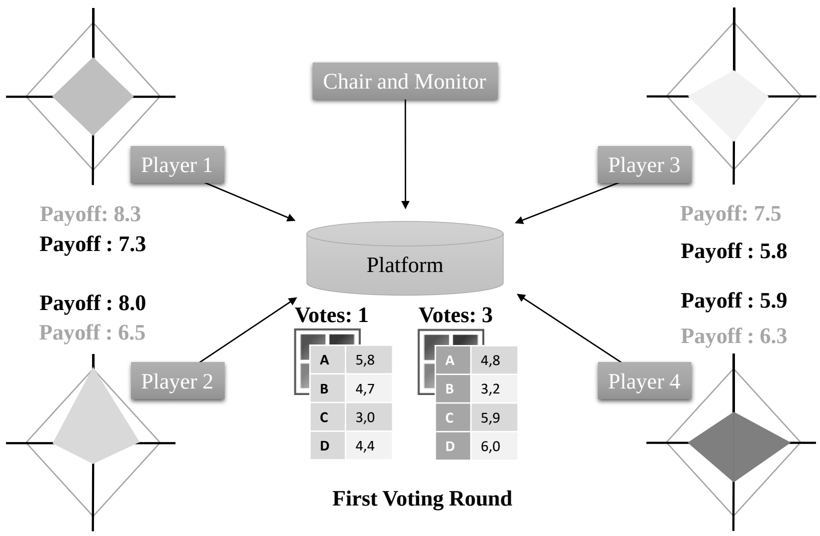

Figure 5. First voting round starts since there are no further proposals

Since player 2 no longer loses payoff from voting for the first proposition thanks to the compensation, the second voting round has a unanimous result (Figure 6). For that reason, the vote over this component was successful and the voting game can continue with collecting propositions for the next component.

With the presented approach, the issue of diluting the platform is addressed. The voting mechanism allows every player to follow the own goals by giving a vote and expressing their specifications via proposing components or change requests. The possibility of setting rules and making adjustments enables the vote's chair to pursue the goal of keeping a certain degree of shared components, if desired, with the methods like compensation or exchange.

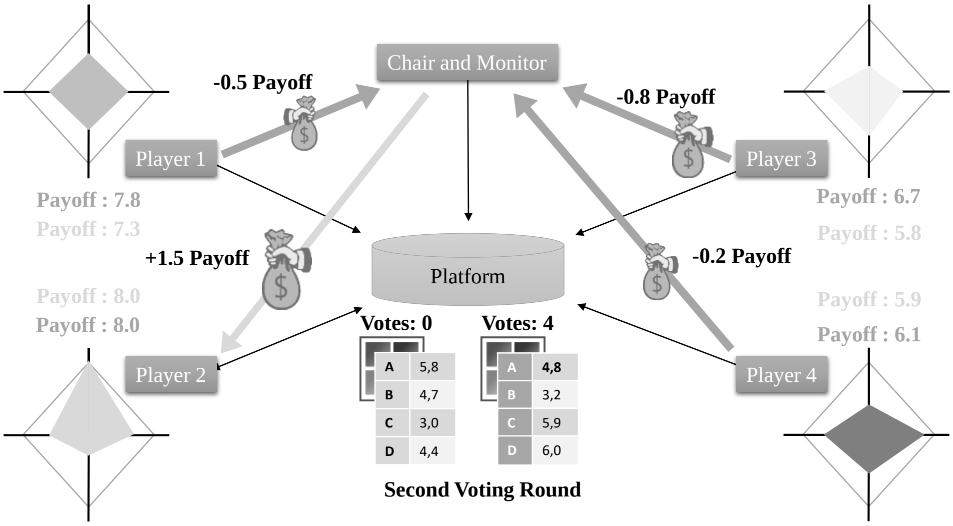

Figure 6. Since the result is not unanimous, the exchange-adjustment takes effect for the second voting round

4 SUMMARY AND FUTURE WORK

In this paper, we describe an initial conceptual approach towards using agreement technologies to handle conflicts in multi-brand product development occurring while coordinating a common platform. First, we describe a corporate group in the automotive industry and its companies as socio-technical systems. These systems and its elements are modelled in a three-level architecture. The macro, meso and micro levels express the relationships between elements as well as conflicts of interest. We identified conflicts occurring during the definition and coordination of a common platform and presented an approach based on voting and game theory to address these conflicts.

The main contributions of this paper are: (i) using game theory as a first approach and discussing the scope of modelling options (Section 3), (ii) identifying the necessary information to enable the modelling of conflicts in multi-brand product development (Section 3.1 and 3.2) and (iii) the voting protocol defining the properties and sequence of the multi-brand platform game (Section 3.3).

With future work, we will proceed towards a formal voting model, which will allow us to analyse properties of different protocols, and analyse different strategy pairs with respect to properties such as pareto-efficiency, stability with respect to equilibria, and voter satisfaction. Besides, the current game-theoretic setting needs to be extended to be able to consider the problem of platform definition as an optimization problem. When defining the game and its calculation of payoffs the questions of "optimal" payoff was already posed. An optimal overall payoff can be defined in various ways. For example, as (i) the sum of all individual payoffs, (ii) the calculated average of all payoffs or (iii) maximisation of the lowest individual payoff, to give some examples. These approaches will be compared and surveyed. It is also necessary to focus on the micro perspective. It is important to keep in mind and reflect the fact that optimality criteria at macro-level (system optimum) and micro-level (user optimum) will differ, and trade-offs will be made regarding the overall outcome. A key topic of future research will be, how the macro level (corporate group) can inject incentives into the agreement mechanisms to nudge the meso- and micro level behaviour into accepting solutions, which are closer to system optimum. This includes analysing approaches different to financial compensation.

When changing a component, the consequences of possible follow-up changes have to be taken into account. Within the proposed game, the players vote over components that they want to share in their products. When a player uses a new component due to a votes outcome this might lead to problems within the related product. If product specific components are incompatible with the newly voted component, the result are mandatory changes. Those can in turn reduce the payoff, which was gained by including the new component into the product in the first place. Considering this comprehensive aspect in the calculation of payoffs requires an in-depth survey.

In our approach the categories used to rate components and the rating itself are assumed pre-existing. How a catalogue of categories, which describe and rate components, can be specified will be part of future work. Once this is researched the subsequent question of how to rate a component according to these categories needs to be answered as well.

Lastly, the factor of cost savings due to joint procurement will be scope of future work. The question whether or not cost-savings can justify using an evaluated component is difficult to answer and requires a comprehensive analysis.

Studying engineering product development processes to model them using game theory is the goal of current work. The processes relate to the development of singe components and include coordination between different engineers. The approach is closely attached to the industrial practice to outline the benefits of our game theory approach.

REFERENCES

Bartholomae, F.W. and Wiens, M. (2016), *Spieltheorie: Ein anwendungsorientiertes Lehrbuch*, Springer Gabler, Wiesbaden.

Bender, J., Kehl, S. and Müller, J.P. (2015), "A Comparison of Agent-Based Coordination Architecture Variants for Automotive Product Change Management", in Müller, J.P. (Ed.), *Multiagent system technologies: 13th German Conference MATES 2015 Cottbus Germany September 28 - 30 2015 revised selected papers: MATES*, *Lecture Notes in Computer Science*, Vol. 9433, 1st ed., Springer International Publishing, Cham, pp. 249–267.

Brandt, F., Conitzer, V., Endriss, U., Lang, J. and Procaccia, A.D. (Eds.) (2016), *Handbook of computational social choice*, Cambridge University Press, New York, NY, USA.

Bucciarelli, L.L. (1994), *Designing Engineers*, *Inside technology*, INSTITUTE OF TECHNOLOGY.

Cheng, H. and Chu, X. (2012), "A network-based assessment approach for change impacts on complex product", *Journal of Intelligent Manufacturing*, Vol. 23 No. 4, pp. 1419–1431.

Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting Change Propagation in Complex Design", *Journal of Mechanical Design*, Vol. 126 No. 5, p. 788.

Cross, N. (2008), *Engineering design methods: Strategies for product design*, 4. ed., Wiley, Chichester.

Durão, L.F.C.S., Eichhorn, H., Anderl, R., Schützer, K. and Zancul, E.d.S. (2015), "Integrated Component Data Model based on UML for Smart Components Lifecycle Management: A Conceptual Approach", in Bouras, A., Eynard, B., Foufou, S. and Thoben, K.-D. (Eds.), *Product Lifecycle Management in the Era of Internet of Things: 12th IFIP WG 5.1 International Conference PLM 2015 Doha Qatar October 19-21 2015 Revised Selected Papers*, *IFIP Advances in Information and Communication Technology*, Vol. 467, Springer International Publishing, Cham, pp. 13–22.

Hamraz, B., Caldwell, N.H.M. and Clarkson, P.J. (2013), "A Holistic Categorization Framework for Literature on Engineering Change Management", *Systems Engineering*, Vol. 16 No. 4, pp. 473–505.

Kehl, S., Hesselmann, C., Stiefel, P. and Müller, J.P. (2016), "Static Product Structures: An Industrial Standard on the Wane", in Harik, R., Rivest, L., Bernard, A., Eynard, B. and Bouras, A. (Eds.), *Product Lifecycle Management for Digital Transformation of Industries*, Springer International Publishing, Cham.

Kehl, S., Stiefel, P. and Mueller, J.P. (2015), "Changes on changes: towards an agent-based approach for managing complexity in decentralized product development", in Weber, C., Husung, S., Cascini, G., Cantamessa, M. and Marjanović, D. (Eds.), *The 20th International Conference on Engineering Design (ICED15): 27th-30th July 2015, Politecnico di Milano, Italy: Proceedings of ICED15: Design organisation and management*, *Proceedings of the 20th International Conference on Engineering Design*, Vol. 80,3, Design Society, Glasgow, Scotland, pp. 219–228.

McMahon, C.A. (1994), "Observations on Modes of Incremental Change in Design", *Journal of Engineering Design*, Vol. 5 No. 3, pp. 195–209.

Nisan, N. (Ed.) (2008), *Algorithmic game theory*, Repr., [Nachdr.], Cambridge Univ. Press, Cambridge.

Ossowski, S. (2013), *Agreement Technologies*, Springer Netherlands, Dordrecht.

Pitt, J. (2005), "Voting in Multi-Agent Systems", *The Computer Journal*, Vol. 49 No. 2, pp. 156–170.

Sanderson, D.W. (2013), "Adaptation Strategies for Self-Organising Electronic Institutions", PhD-Thesis, Imperial College London, 1 January.

Saurin, T.A. and Gonzalez, S.S. (2013), "Assessing the compatibility of the management of standardized procedures with the complexity of a sociotechnical system: case study of a control room in an oil refinery", *Applied ergonomics*, Vol. 44 No. 5, pp. 811–823.

Tseng, M.M. and Jiao, J. (2007), "Mass Customization", in *Handbook of Industrial Engineering*, John Wiley & Sons, Inc, pp. 684–709.

Wooldridge, M. (2009), *An introduction to multiagent systems*, 2. ed., Wiley, Chichester.