University of Groningen

Utility-based Decision Making in Collective Adaptive Systems

Andrikopoulos, Vasilios; Bitsaki, Marina; Saez, Santiago Gomez; Karastoyanova, Dimka; Nikolaou, Christos; Psycharaki, Alina

Published in:
CLOSER 2014

DOI:
10.5220/0004937403080314

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2014

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Andrikopoulos, V., Bitsaki, M., Saez, S. G., Karastoyanova, D., Nikolaou, C., & Psycharaki, A. (2014). Utility-based Decision Making in Collective Adaptive Systems. In M. Helfert, F. Desprez, D. Ferguson, F. Leymann, & V. M. Muñoz (Eds.), *CLOSER 2014 : Proceedings of the 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain, April 3-5, 2014* (pp. 308-314). SciTePress. https://doi.org/10.5220/0004937403080314

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 21-08-2022

Utility-based Decision Making in Collective Adaptive Systems

Vasilios Andrikopoulos[1], Marina Bitsaki[2], Santiago Gómez Sáez[1], Dimka Karastoyanova[1],
Christos Nikolaou[2] and Alina Psycharaki[2]
[1] *Institute of Architecture of Application Systems, University of Stuttgart, Stuttgart, Germany*
[2] *Transformation Services Laboratory, University of Crete, Heraklion, Greece*

Keywords: Collective Adaptive Systems, Utility, Decision Making, Choreography.

Abstract: Large-scale systems comprising of multiple heterogeneous entities are directly influenced by the interactions of their participating entities. Such entities, both physical and virtual, attempt to satisfy their objectives by dynamically collaborating with each other, and thus forming collective adaptive systems. These systems are subject to the dynamicity of the entities' objectives, and to changes to the environment. In this work we focus on the latter, i.e. on providing the means for entities in such systems to model, monitor and evaluate their perceived utility by participating in the system. This allows for them to make informed decisions about their interactions with other entities in the system. For this purpose we propose a utility-based approach for decision making, as well as an architecture that allows for the support of this approach.

1 INTRODUCTION

Collective systems comprise heterogeneous entities collaborating towards the achievement of their own objectives, and the overall objective of the collective (Andrikopoulos et al., 2013). Such systems are usually large scale, typically consisting of both physical and virtual entities distributed both organizationally and geographically. Entities in these systems actively influence the operation of the system by interacting with other entities. Their behavior is guided by their individual objectives and the decisions they make in order to satisfy them. In this respect, support for decision making on the level of entities is an integral part of the realization of such systems.

For this purpose, in the EU ALLOW Ensembles project[1], we use the concept of Collective Adaptive System (CAS) (Kernbach et al., 2011), and define the underpinning concepts for modeling, execution and adaptation of CAS entities and their interactions. In particular, we propose to model and manage entities as collections of cells encapsulating their functionalities. Entities collaborate with each other to achieve their objectives in the context of ensembles describing the interactions among them. As a way to measure entity objectives' achievement, we use the concept of *utility* as a function over its preferences, context, state and interactions with other entities.

The contributions of this work can be summarized as follows:

1. We propose a utility-based approach for the purpose of decision making in CAS systems based on the conceptual model introduced in the project.
2. We extend an existing architecture in order to enable the modeling and execution of enabling mechanisms, and discuss their implementation based on well-established technologies.

The remaining of the paper is structured as follows: starting from a motivating scenario in Section 2, we then introduce our proposal on how to deal with decision making in CAS systems (Section 3). Consequently, in Section 4 we extend the architecture introduced in (Andrikopoulos et al., 2013) in order to enable the modeling and execution of the mechanisms supporting our proposal as distributed, large scale, pervasive systems; we also discuss the implementation of these concepts based on well-established technologies. The paper closes with a summary of related work (Section 5), and concludes with an outline of research challenges and future work (Section 6).

---

[1] ALLOW Ensembles: http://www.allow-ensembles.eu

Andrikopoulos V., Bitsaki M., Goméz Sáez S., Karastoyanova D., Nikolaou C. and Psycharaki A..
Utility-based Decision Making in Collective Adaptive Systems.
DOI: 10.5220/0004937403080314
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 308-314
ISBN: 978-989-758-019-2

2 MOTIVATION

Supporting citizens mobility within the urban environment is a priority for municipalities worldwide. As part of the effort to offer "smart services" to citizens, in the ALLOW Ensembles project the case of an urban mobility scenario is used as a CAS for demonstration and evaluation purposes (Andrikopoulos et al., 2013). The FlexiBus service is a special type of bus in this scenario that operates a flexible route set by passenger needs, allowing the advance booking of pickup points along a (dynamically) calculated route to a set of predefined destinations (i.e., city center).

Using the FlexiBus system, each *Passenger* can request a trip to one of the predefined destinations in the system by communicating with the *Route Planner*, asking to start at a certain time and from a preferred pickup point. Each passenger can pay their trip directly in the bus (cash, with a credit card or a monthly pass) or through the FlexiBus company web site. Furthermore, during the route execution, each passenger waiting for a bus can be notified for problems on a selected route (e.g. bus delays, accidents, etc.). Each *Bus Driver* is assigned by the *Route Manager* to a precise but dynamically calculated by the Route Planner route to execute, including the list of passengers assigned to it, and a unique final destination. Bus drivers communicate with an assigned Route Manager which monitors and ensures the correct execution of the route, in order to ask for the next pick-up point, and to communicate information like passengers' check-in. During the route realization, a bus on route can also accept passengers that have not previously booked if there are available seats and the passenger pick-up point is along the route.

The Route Planner attempts to define routes in a manner that reduces the wasteful overlap that occurs with many individual passengers traveling to the same destination. This means carrying more people in one route to radically reduce the number of FlexiBuses on the road, fuel usage, CO_2 emissions, traffic congestion, etc. The passengers on the other hand have concrete requirements on e.g. their arrival time, and they attempt to optimize across changing dimensions e.g. fare cost or time spent on the bus. The FlexiBus system is therefore required to support the interaction of different entities (software or physical) striving to achieve potentially conflicting and changing over time goals. In the following sections we present our proposal for using utility as the means for allowing these entities to co-exist and reconciliate their goals in the system.

3 APPROACH

3.1 The ALLOW Ensembles Model

In large-scale collective adaptive systems, entities which are heterogeneous actors of both human and software actors, interact in various ways to form a distributed and dynamic CAS system that evolves continuously to achieve a goal in a given context. For this purpose, in ALLOW Ensembles we model both types of actors as *entities* aggregating different functionalities, e.g. the check-in on a bus route, or the payment of the fare, as reusable *cells*. Cells encapsulate some functionality that the entity offers to, or requires from the system. Cells from different entities interact with each other as part of their predefined functionality.

The outcome of these interactions are *ensembles* that are created to fulfill specific goals initiated by the entities. Even though each entity has its own selfish interests, the cooperation with others entails an increase in each one's satisfaction expressed in utility terms. For this purpose, in the scope of the project we propose to model the economic perspective of systems of ensembles. In particular, we consider that entities take part in games according to various criteria in order to achieve a set of objectives. Entities have potentially conflicting interests within the system and strive to achieve individual objectives, thus formulating *non-cooperative games* (Osborne and Rubinstein, 1994). On the other hand, entities cooperate to achieve a common task and fulfill a set of group objectives, a situation that can modeled as *cooperative games* (Wiese, 2010). We incorporate both types of games to model entities' behavior according to the goals needed to be achieved.

3.2 Utility & Strategic Ensembles

Our position for dealing with such CAS systems as the ones discussed above is to analyze economic models that assign *utility* (Norstad, 1999) to entities according to their properties and impose constraints according to the desired goals. The utility of an entity is a measure of satisfaction experienced by the entity for using a service or consuming a good. Entities make choices so as to maximize their utility. A utility function is a way to quantify utility, assigning a number to every possible choice of the entity such that more preferred choices get assigned larger numbers than less preferred ones (Varian, 2010). We consider that the cells of an entity that are involved in the selected ensemble operate in order to satisfy the goal that is fulfilled by the ensemble and contribute to the improvement of the entity's utility. If the entity participates in another ensemble (this is determined by the interactions that the entity decides to get involved to), the same cells operate in a new context that attributes a new level of utility to the entity. Game theory can be used to model the selection from a set of candidate ensembles based on the criterion of *utility maximization* (Varian, 2010): each entity chooses the ensemble that maximizes its utility taking into account the actions taken by the other entities too. Once an ensemble is being executed, it can be evaluated and compared to competitors or former ensembles through the *collective utility* that aggregates the individual utilities of all entities involved.

In order to model the economic behavior of our system, we define the concept of *meta-cells* that represent the economic characteristics of functional cells. We distinguish different activities that are performed by meta-cells according to the goals assigned to the respective functional cells. More specifically, they calculate the utility of an entity when participating in a given ensemble, as well as the collective utility of an ensemble derived by the various entities that interact to form the ensemble, they communicate with other cells and take decisions/compute strategies of entities, they collect data and perform measurements (resource consumption, satisfaction, costs, delays, prices, etc.), and they run optimization algorithms to improve the performance of ensembles.

Interactions of functional cells initiated by the entities result in the creation of functional ensembles (or simply, ensembles) in order to fulfill specific goals. Simultaneously, strategic utility-based interactions trigger the initialization of meta-cells that are managed by a new structure called the *strategic ensemble* in order to handle decision making on the level of interaction between entities. The objectives of a strategic ensemble include the following:

- to impose constraints according to entity goals in order to reduce the various choices of entities,
- to evaluate ensembles from the perspective of one entity according to the entity's preferences dynamically,
- to assign utility to each entity when participating in an ensemble in order to make the optimal choice and manage the negotiation between two entities.

Figure 1 illustrates one of the strategic ensembles in the FlexiBus scenario discussed in the previous sections. The meta-cells of the entities in the FlexiBus scenario are described as follows:

1. Route Evaluation (Route Planner): evaluates each route according to passenger preferences, passenger's previous choices and route's reliability based on historical data.

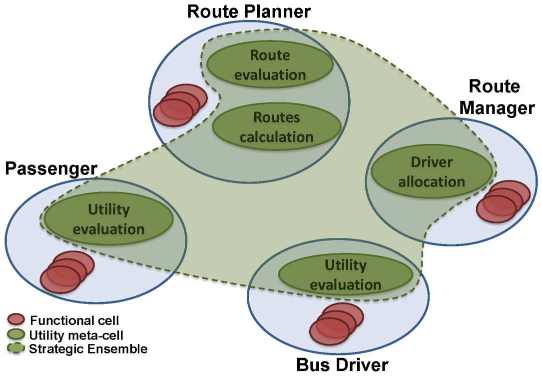

Figure 1: A Strategic Ensemble in the FlexiBus Scenario.

2. Routes Calculation (Route Planner): makes a list of routes according to passenger goals, examines whether the requirements of current passengers are violated (estimate new travel times) for all routes in the list and removes the routes that do not satisfy passengers' goals or negotiates with passengers, evaluates the routes according to passengers' preferences and FlexiBus' benefits.
3. Driver Allocation (Route Manager): negotiates with the FlexiBus drivers to find an appropriate Bus Driver entity to execute a route.
4. Utility Evaluation (Bus Driver): calculates the utility of the driver for the requested route according to his preferences and accepts or rejects the offer of the Route Manager to execute a route.
5. Utility Evaluation (Passenger): calculates a passenger's utility for each route in the list and announces the optimal choice. In Algorithm 1 we provide an example of passenger utility as a function of travel time t, trip cost C and a set of preferences such as payment method, smoking/non-smoking trips, seat preferences, delay tolerance and so on.

The structure of our methodology implies that information flows among the various components of the system. Meta-cells extract information from entities. Goals may impose constraints on meta-cells that solve optimization problems. Entities communicate through meta-cells and use them in order to select appropriate cells to form ensembles or reconfigure ensembles for increasing system's utility.

3.3 Strategic Ensembles Life-cycle

The life-cycle of the strategic ensemble is closely related to that of the execution ensembles. It is created before the execution ensemble, since the meta-cells

Algorithm 1 : Utility Evaluation Algorithm for Passenger Entities.

```
for all routes do calculate $U$:
    $U = \sum_{i=1}^{k} w_i v_i + w_{k+1} e^{-at-bC}$, where
    $v_i = \begin{cases} \frac{(m+1)-j}{(m+1)} & \text{if choice } j \text{ of a total of } m \text{ choices is satisfied} \\ 0 & \text{if preference } i \text{ is not satisfied,} \end{cases}$
    $w_i, i = 1, \ldots, k$ are weights for each of $k$ preferences and $w_{k+1}$ the weight for the component that accounts for travel time and cost ($\sum_{i=1}^{k+1} w_i = 1$), and
    $a$ and $b$ are the impact factors for travel time $t$ and cost $C$ respectively.
end for
return $max\{U\}$
```

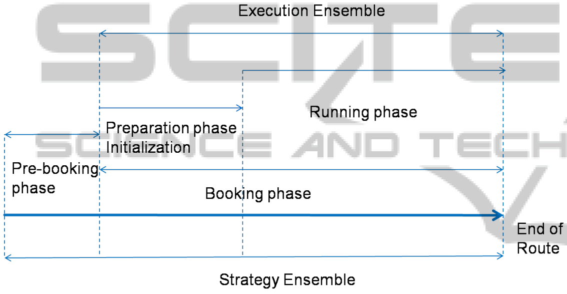

Figure 2: The life-cycle of the Strategic Ensemble.

are responsible for making decisions about the selection of the most beneficial ensembles. Moreover, it runs in parallel to the execution ensembles, since it affects the operations of the execution ensembles until their termination. We consider that each entity has its own set of meta-cells, which interact with other cells of this entity and with cells and meta-cells of other entities as well. Meta-cells of an entity join or leave a strategic ensemble and as a result the related execution ensembles, according to the decisions the entity makes based on her calculated utility.

In particular about the FlexiBus scenario, we consider two phases in the life-cycle of a route (execution ensemble): the *pre-booking phase*, where a route is going to be executed if a certain number of requests is reached until a certain deadline, and *the execution phase*, where the route is bound to start or it has already started. The timeline for the strategic ensemble that runs through this execution ensemble is shown in Figure 2. According to the scenario, the execution cell of the Passenger invokes her meta-cell by sending the trip request. Then, a series of meta-cell interactions start with the aim to make the optimal decision for each entity. The Utility Evaluation meta-cell of the Passenger invokes the Routes Calculation meta-cell of the Route Planner by sending the trip request and waits to receive a list of routes, in order to evaluate them. The Routes Calculation meta-cell interacts with the Route Evaluation meta-cell of the Route Planner, as well as with the active Route Managers in order to identify whether the Passenger can join an existing route. If this is possible, the Route Planner negotiates with the Passengers already in the route, since they may need to re-evaluate their utility from being on this route, if e.g. the arrival time changes due to the additional passenger. The list of routes returned to the Passenger are evaluated by the Passenger using Algorithm 1 and ranked accordingly. The Passenger then joins in an (execution) ensemble with the Route Manager, Bus Driver and other Passengers in the route that she evaluated as optimal.

4 ARCHITECTURE & IMPLEMENTATION

An architecture design capable of supporting the phases of the CAS lifecycle is presented in (Andrikopoulos et al., 2013). However, such architecture does not support the utility-based decision making capabilities proposed in the previous section; for this purpose in this work we enhance it with the necessary components as depicted in Figure 3.

The architecture for the modeling and the execution of CAS systems comprises two major component groups: a *Modeling Tool* and a *Runtime Environment*. The *Modeling Tool* consists of three main components: a *Choreography Modeler* to create choreography models which specify the interactions between cells of functional and strategic ensembles, a *Transformer* to generate process skeletons that can be completed to executable processes, which can be subsequently modified in the *Process Modeler* The *Modeling Tool* is developed as a pair of interoperable Eclipse Graphical Editors that support the specification of choreographies in the BPEL4Chor language (Decker et al., 2008). Both functional cells and meta-cells are specified using the WS-BPEL language, and therefore the process is developed as an extension of the BPEL Eclipse designer described in (Sonntag et al., 2012). The editor is also equipped with monitoring capabilities that provide the user real-time information related to the flow execution.

The *Runtime Environment* comprises the necessary components to enable the execution of both cells and meta-cells. The cell and meta-cell models are deployed on one or more *Execution Engines* and can be instantiated at any time. Cell models may contain abstract activities, and therefore the Execution Engine must be able to start the execution of incomplete pro-

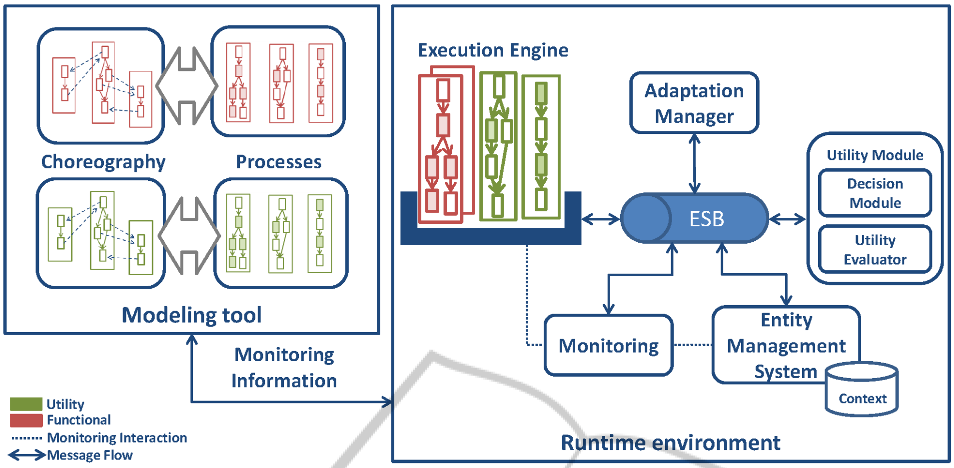

Figure 3: Architecture overview

cesses, allowing the dynamic injection of additional activities retrieved from the *Adaptation Manager* or triggered through the *Modeling Tool*. The realized *Execution Engine* is an extended Apache ODE Engine[2], an open source implementation of BPEL, and presented in (Sonntag et al., 2012). The *Entity Management System (EMS)* deals with all aspects of entity management. When a new entity is created, its available functional and meta-cells are deployed in the *Execution Engine*, and its corresponding properties to its context model. The EMS allows the Adaptation Manager to access the cell models, instances, context properties, and the necessary data for utility evaluation, e.g. user preferences, needed for the utility based planning and adaptation operations.

The interaction of the *Utility Module* with one or multiple meta-cells participating in a strategic ensemble provides the necessary functionality for utility evaluation and decision making support. The *Decision Module* and the *Utility Evaluator* deal with such operations in a generic and domain independent manner. Therefore, meta-cells are responsible for retrieving the necessary data, e.g. context information, user preferences, etc., and orchestrate the multiple services offered in the *Utility Module*. Monitoring information retrieval, e.g. KPIs or WS-BPEL Events (Kopp et al., 2011) functionalities must be wrapped in the *Monitoring* component.

All components should be provided as services and communicate through an *Enterprise Service Bus (ESB)* solution to facilitate their integration. As multiple organizational levels can be present in such systems, we use the ESB^{MT} multi-tenant aware ESB solution, as presented in (Strauch et al., 2013), for ensuring communication isolation between multiple organizations and its users. ESB^{MT} enhances the Apache ServiceMix solution[3] with multi-tenant communication support within service endpoints deployed in the ESB, and multi-tenant aware dynamic endpoint deployment and management capabilities.

5 RELATED WORK

Several studies related to the collective or adaptive aspects of complex systems have been driven in areas such as *Swarm Intelligence* (C. Pinciroli et al., 2011; Levi and Kernbach, 2010), *Autonomic computing* (Bruni et al., 2012; Lewis et al., 2012), and *Multi-agent based systems* (Cabri et al., 2011; Lavinal et al., 2006). These converge in the need for defining interaction rules between participants towards adapting the system under ordinary and extraordinary events. However, there exists a gap towards a general solution definition for all aspects in CAS. In this work we aim to target this problem by proposing the usage and extension of existing modeling and execution approaches towards providing a generic and evolutionary utility-based adaptation in CAS.

Interactions between multiple participants in a choreography can be modeled following the interaction or the interconnection modeling approaches (Barker et al., 2009). In the former, communication between participants is modeled using atomic *interaction activities*. The WS-CDL (Kavantzas et al., 2005) and the Savara[4] project are approaches that support the explicit specification of the *interaction activities*. On the other hand, the interconnection modeling approach consists on interconnecting the communication activities of each partic-

---

[2] Apache ODE: http://ode.apache.org/

[3] Apache ServiceMix: http://servicemix.apache.org

[4] http://www.jboss.org/savara

ipant of the choreography. This approach is supported in the CHOReOS Integrated Development and Runtime Environment[5], in the Open Knowledge European project[6], and in BPEL4Chor (Decker et al., 2007). As BPEL4Chor enables the choreography specification atop of WS-BPEL and decouples the choreography behavior specification from the technical communication details, it is considered as an appropriate extensible point in CAS modeling and specification.

Value is intimately linked to service systems: the latter exist only as long as they can create value for both the providers and the consumers; otherwise they dissolve and disappear. In (Bloch and Jackson, 2007), value is defined as “benefits of an agent accrued by his participation in the network minus any costs involved in setting up the network links directly or indirectly”. A method is proposed in (Caswell et al., 2008), for computing values by taking into account service system partners’ satisfaction and additional value accrued by the relationship levels of the various partners. In (Gordijn et al., 2001), the authors combine IT systems analysis with business modeling to build an e-business model that specifies e-business scenarios rather than defining values.

While various approaches have been proposed to measure service network performance, little experimental testing or theoretical investigation on competing networks has been performed. (Biem and Caswell, 2008) describe building block elements of a value network model and design a network-based strategy for a prescriptive value network analysis. (Allee, 2008) provides a systematic but qualitative way for approaching the dynamics of intangible value realization, inter-convertibility, and creation. Both of the above approaches use qualitative methods to describe value in a service network in contrast to (Caswell et al., 2008) that calculates value in a quantifiable manner. A service system’s performance is studied in (Voskakis et al., 2011) by solving value optimization problems with respect to service prices.

(Van Dinther et al., 2009) studies the strategic behavior of service providers within service value networks and proposes an auction-based mechanism to efficiently match service offers and requests and determine prices, showing that incentive compatibility holds under certain conditions. (Katsamakas, 2005) analyzes industry structure in the presence of value networks and has developed a model dealing with a number of design aspects, such as the supplier number and the partner investments importance. It is shown that industry structure is more likely to shift to competition between value networks in which IT plays an important role. Systems encountering competition between service networks, analyzed with respect to their evolution through well-defined strategies, have received little attention. Initial work is provided in (Bitsaki et al., 2012) on competing service networks by defining and simulating strategies in games of incomplete information. The results show that Nash Equilibria exist when one player represents each service system.

6 CONCLUSIONS AND FUTURE WORK

Collective Adaptive Systems (CAS) are formed by multiple heterogeneous virtual and physical entities that collaborate with each other towards fulfilling their individual objectives (Kernbach et al., 2011). However, such collaboration can change over time, as entities’ objectives can dynamically adapt or change due to external or internal environmental events. Therefore, on the one hand entities should be provided with the necessary mechanisms that will guarantee a successful fulfillment of their objectives, while on the other hand such mechanisms must minimize the impact of entities’ influences on the collective. For this purpose, the concepts of utility and decision making are brought into the cells and ensembles paradigm of CAS in the ALLOW Ensembles project.

In this work we focus on providing the necessary support to entities to facilitate the modeling, monitoring, and evaluation of their perceived utility by participating in the system. Therefore, we describe in a first step a conceptual view on meta-cells and strategic ensembles, and propose the strategic ensemble lifecycle as an emerging negotiation mechanism. A utility evaluation algorithm is then proposed for a concrete entity of the FlexiBus scenario. We plan to extend the approach to model the various interactions among entities by means of games in order to make decisions on which ensembles to choose to participate or which entities to add in an existing ensemble, taking into account future passenger demands or other environmental events.

In a second step, an architectural and a work in progress implementation approach presented in (Andrikopoulos et al., 2013) is enhanced towards supporting the multiple required phases for utility-based decision making in CAS. Future work also focuses on analyzing and utilizing multiple provisioning and management techniques for supporting entities joining and leaving the system, as well as investigating

---

[5] CHOReOS: Large Scale Choreographies for the Future Internet: http://www.choreos.eu/

[6] Open Knowledge: http://www.openk.org/

on different distribution and deployment options of the runtime environment. For these purposes, the previously described components must be extended and integrated in the prototype implementation.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the EU's Seventh Framework Programme (FP7/2007-2013) project ALLOW Ensembles (grant agreement no. 600792), and from the German Research Foundation (DFG) within the Cluster of Excellence (EXC310) in Simulation Technology.

REFERENCES

Allee, V. (2008). Value network analysis and value conversion of tangible and intangible assets. *Journal of Intellectual Capital*, 9(1):5–24.

Andrikopoulos, V., Bucchiarone, A., Sáez, S. G., Karastoyanova, D., and Mezzina, C. A. (2013). Towards Modeling and Execution of Collective Adaptive Systems. In *Proceedings of WESOA'13*, pages 1–12. Springer.

Barker, A., Walton, C. D., and Robertson, D. (2009). Choreographing web services. *IEEE Transactions on Services Computing*, 2(2):152–166.

Biem, A. and Caswell, N. (2008). A value network model for strategic analysis. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 361–361. IEEE.

Bitsaki, M., Nikolaou, C., Voskakis, M., van den Heuvel, W.-J., and Tsikrikas, K. (2012). Performance analysis and strategic interactions in service networks. *International Journal On Advances in Networks and Services*, 4(3 and 4):375–385.

Bloch, F. and Jackson, M. O. (2007). The formation of networks with transfers among players. *Journal of Economic Theory*, 133(1):83–110.

Bruni, R., Corradini, A., Gadducci, F., Lluch-Lafuente, A., and Vandin, A. (2012). A conceptual framework for adaptation. In *FASE*, pages 240–254.

C. Pinciroli et al. (2011). Argos: A modular, multi-engine simulator for heterogeneous swarm robotics. In *IROS*, pages 5027–5034.

Cabri, G., Puviani, M., and Zambonelli, F. (2011). Towards a taxonomy of adaptive agent-based collaboration patterns for autonomic service ensembles. In *CTS*, pages 508–515.

Caswell, N. S., Nikolaou, C., Sairamesh, J., Bitsaki, M., Koutras, G., and Iacovidis, G. (2008). Estimating value in service systems: a case study of a repair service system. *IBM Systems Journal*, 47(1):87–100.

Decker, G., Kopp, O., Leymann, F., Pfitzner, K., and Weske, M. (2008). Modeling Service Choreographies Using BPMN and BPEL4Chor. In *Proceedings of CAiSE'08*, pages 79–93. Springer.

Decker, G., Kopp, O., Leymann, F., and Weske, M. (2007). BPEL4Chor: Extending BPEL for Modeling Choreographies. In *Proceedings of ICWS'07*, pages 296–303, Salt Lake City. IEEE Computer Society.

Gordijn, J., Akkermans, H., and Van Vliet, J. (2001). Designing and evaluating e-business models. *IEEE intelligent Systems*, 16(4):11–17.

Katsamakas, E. (2005). Competing value networks, incomplete contracts and it. In *Proceedings of HICSS'05*, pages 212b–212b. IEEE.

Kavantzas, N., Burdett, D., Ritzinger, G., Fletcher, T., Lafon, Y., and Barreto, C. (2005). Web services choreography description language version 1.0.

Kernbach, S., Schmickl, T., and Timmis, J. (2011). Collective adaptive systems: Challenges beyond evolvability. *ACM Computing Research Repository (CoRR)*.

Kopp, O., Henke, S., Karastoyanova, D., Khalaf, R., Leymann, F., Sonntag, M., Steinmetz, T., Unger, T., and Wetzstein, B. (2011). An Event Model for WS-BPEL 2.0. Technischer Bericht Informatik 2011/07, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany.

Lavinal, E., Desprats, T., and Raynaud, Y. (2006). A generic multi-agent conceptual framework towards self-management. In *NOMS*, pages 394–403.

Levi, P. and Kernbach, S. (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, volume 7. Springer Verlag.

Lewis, P., Platzner, M., and Yao, X. (2012). An outlook for self-awareness in computing systems. *Awareness Magazine*.

Norstad, J. (1999). An introduction to utility theory. Unpublished manuscript at http://homepage. mac. com/j. norstad.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Sonntag, M., Hahn, M., and Karastoyanova, D. (2012). Mayflower - Explorative Modeling of Scientific Workflows with BPEL. In *Proceedings of BPM'12*, pages 1–5. CEUR Workshop Proceedings.

Strauch, S., Andrikopoulos, V., Sáez, S. G., and Leymann, F. (2013). ESB^{MT}: A Multi-tenant Aware Enterprise Service Bus. *International Journal of Next-Generation Computing*, 4(3):230–249.

Van Dinther, C., Blau, B., and Conte, T. (2009). Strategic behavior in service networks under price and service level competition. In *Wirtschaftsinformatik (1)*, pages 599–608. Citeseer.

Varian, H. (2010). *Intermediate microeconomics: a modern approach*. Norton & Company.

Voskakis, M., Nikolaou, C., van den Heuvel, W.-J., and Bitsaki, M. (2011). Service network modeling and performance analysis. In *Proceedings of ICIW'11*, pages 58–63.

Wiese, H. (2010). Applied cooperative game theory. http://www.uni-leipzig.de/ micro.