

# Resource Allocation, Trading and Adaptation in Self-managing Systems

Guglielmo Lulli<sup>1</sup>, Pasqualina Potena<sup>2</sup>, and Claudia Raibulet<sup>1</sup>

<sup>1</sup> Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione, Viale Sarca 336, U14, 20126 Milan, Italy

<sup>2</sup> Università degli Studi di Bergamo, Viale Marconi, 24024, Bergamo, Italy  
*{lulli, raibulet}@disco.unimib.it,*  
*pasqualina.potena@unibg.it*

**Abstract.** The allocation of a limited number of resources among multiple self-interested stakeholders is a challenging issue for many real life systems and applications. Resource management for this type of systems is a challenging task because of the different objectives of the owner of the resources and the stakeholders. The owner aims to an efficient usage of the resources, while stakeholders have self-interested objectives. This paper presents a software framework for resource management based on the integration of dynamic allocation, trading, and self-adaptation mechanisms. Resource allocation and adaptation are performed in a centralized manner, while resource trading is achieved through a decentralized approach. Furthermore, the paper presents the application of the proposed framework in two different domains: aeronautics and Internetworks.

**Keywords:** Resource management, resource allocation, resource trading, self-adaptation, resource-constrained system, ATM, SOA.

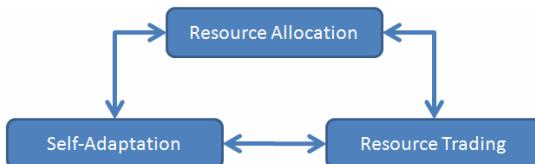
## 1 Introduction

Many real life systems and applications concern the allocation of a limited number of resources among multiple self-interested stakeholders or agents [3, 7, 10, 21, 22]. The allocation of resources may be executed taking into account either the individual objectives of the stakeholders, the global efficiency of the overall system, or both. Often, for this class of systems, the stakeholders are reluctant to disclose their objectives that can also be in contrast with the efficiency of the overall system. This situation may lead to a trade-off between the system and the stakeholder objectives. In addition to the criteria used for the allocation of resources, the allocation has to satisfy a given set of constraints, which are both general purpose - e.g., it is not possible to allocate more than the available resources, a unit of resource cannot be assigned to more than one stakeholder or agent - and domain specific – e.g., a resource cannot be used by more than X stakeholders in a time interval.

The process of allocating resources can be either centralized or decentralized. In a centralized perspective, a central authority is in charge of the allocation process, while

in a decentralized solution, stakeholders exchange, trade and/or barter resources to reach a final allocation. Selecting one solution or the other resolves the trade-off between the stakeholders' objectives and the efficiency of the overall system. Indeed, centralized solutions usually pose emphasis on the performances of the system, while in decentralized solutions stakeholders' objectives play a prominent role.

In this paper, we present a software solution for resource management, which is based on three main modules: allocation, trading, and self-adaptation. Specifically, we propose a modular framework which allows the allocation, the trading, the self-adaptation, or any combination of the three components (see Figure 1). In fact, there might be systems which combine all the three mentioned elements. Resource allocation [10, 22] represents the ability of a system to (re)-assign/distribute the resources to their stakeholders. Several criteria may be used to allocate resources (e.g., the types of stakeholder, the resource requests' priority, the time of the requests, objectives of the requests). Resource trading [5, 19, 20] is the process of the exchange of resources among the stakeholders to achieve their own objectives. Self-adaptation [6, 15] represents the ability of a system to perform changes by itself in the system itself at runtime as a consequence of changes occurred in the execution context of the system or inside the system.



**Fig. 1.** Resource Management through Allocation, Trading and Self-Adaptation

This solution introduces a key innovative feature with respect to the state-of-the-art resource management systems because it proposes an integrated approach to resource management which may include all the modules described above and through which both global and individual objectives are addressed.

In the more general setting, we can imagine that our approach is both centralized and decentralized. The central authority performs a first allocation of resources and supervises the resource trading among the stakeholders, which is obviously a decentralized process. The result of the trading process is communicated to the central authority, which integrates all the partial results communicated by the stakeholders and it might either accept the new allocation, or try to adapt the system to the new allocation plan, when this new allocation plan is not appropriate from the centralized point of view/objectives. If the adaptation is possible, then the system adapts itself and accepts the new resource allocation plan. Otherwise, the system rejects the new allocation plan (indicating also which are the problems) and requires the stakeholders to resume the trading process. The resource management makes the right trade-off among the objectives of the stakeholders, which can be in conflict with each other (e.g., the service provider aims to maximize the resources utilization, whereas a user requires a certain level of system availability) or different (e.g., two users can require different quality attributes).

The scientific literature proposes a wide range of solutions for resource management in various application domains, which range from aeronautics [21] to telecommunications [5], from grid computing [10] to SOA [4, 17] and Web Services [3], from multimedia [18] to multi-agent systems. Typically, they focus on one or two functionalities offered by the framework proposed in this paper. For example, dynamic allocation and re-configuration (a facet of self-adaptation) of the resources based on the current controllers' workload in ATM are addressed in [21]. Self-adaptation in service-oriented application is discussed in [3, 4]. An example of resource trading in multi-agent systems is presented in [1].

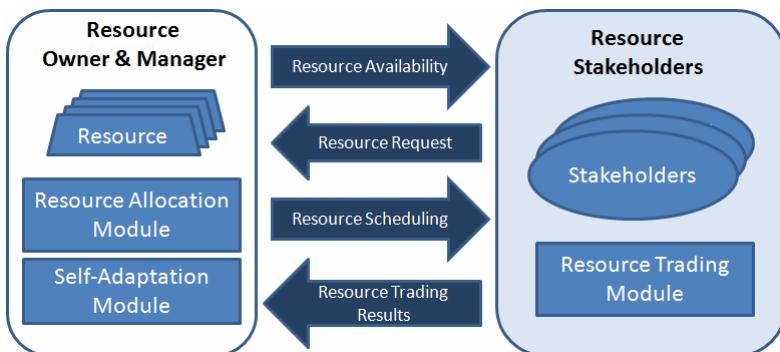
The rest of the paper is organized as follows. Section 2 describes our framework for resource management based on resource allocation, trading and adaptation. Section 3 presents two application examples of our framework on two actual case studies. Conclusions and further work are dealt with in Section 4.

## 2 Resource Management: Allocation, Trading and Adaptation

Given a set of limited resources to be shared among a plurality of self-interested stakeholders, we propose a framework for resource management, which allows the allocation, trading, and self-adaptation. The resources are used by the stakeholders to achieve their own objectives. Their allocation, trading, and self-adaptation have to satisfy a set of constraints, which can be either general, domain specific, or both.

The proposed framework for resource management is shown in Figure 2. It considers an owner of the resources, which is also the manager (Resource Owner & Manager - ROM) and several stakeholders (Resource Stakeholders - RS), which are the users of the resources. RS may ask ROM for resources allocation and modifications in the resource allocation. ROM represents the centralized authority, while the stakeholders represent the decentralized part of the framework.

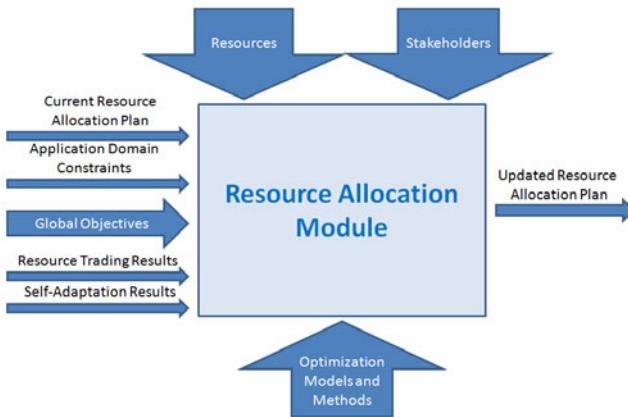
ROM allocates the available resources and accepts/rejects the requests of allocation changes received from the stakeholders. ROM holds a supervision role on the resource allocation process because it has knowledge on the resources (e.g., current number, availability) and on their constraints (e.g., time slot, safety/security). Moreover, it may be dictated by legal issues.



**Fig. 2.** The Main Elements of the Proposed Solution

## 2.1 The Resource Allocation Module

The Resource Allocation (RA) module assigns the available resources to the stakeholders based on global objectives (e.g., the efficient usage of the resources). To execute this task, RA has to know the pool of the resources to be allocated and the pool of stakeholders (see Figure 3). In addition to this information, RA receives in input the current resource allocation plan if any, and the application domain constraints. The RA module is equipped with a supervision capability on the use of resources thus guaranteeing a feasible and efficient allocation of resources. This functionality requires in input the results of the trading and the self-adaptation processes. To provide these functionalities, the RA uses assignment optimization models and methods [10, 11] suitably customized for the considered application domain.

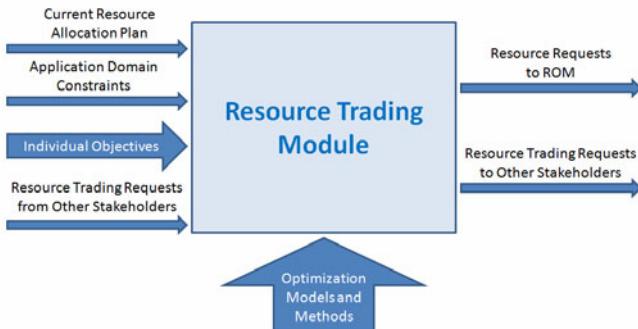


**Fig. 3.** The Resource Allocation Module

The described RA module is domain-independent and it can be used in different application domains.

## 2.2 The Resource Trading Module

The Resource Trading (RT) module, depicted in Figure 4, is used by the stakeholders to exchange resources with other stakeholders and to communicate with ROM. Each stakeholder has an RT module. The communication with ROM includes resource allocation/modification/usage/cancellation requests, which may or may not be a result of a trading process. The communication with the other stakeholders concern exchange requests, thus implying a trading process. Each RT receives in input the current resource allocation plan, if any, the application domain constrains, and the resource trading requests from other stakeholders. The stakeholder decides on an exchange of resources with other stakeholders based on its individual objectives and exploiting customized optimization models and methods. By customization of the optimization models we mean that the optimization models may consider specific constraints and objectives of the stakeholder, therefore different stakeholders may have different optimization models [14, 19].

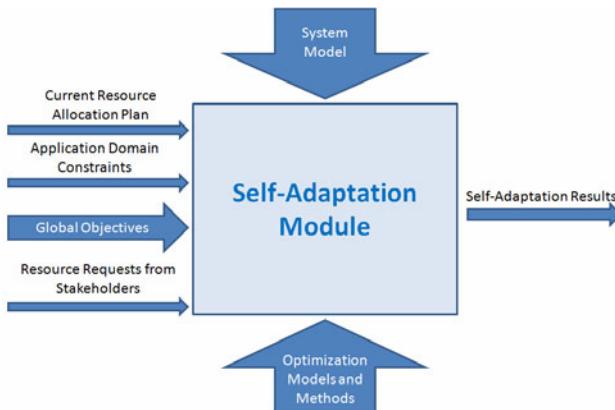


**Fig. 4.** The Resource Trading Module

The described RT module is domain-independent and it can be used in different application domains.

### 2.3 The Resource Self-adaptation Module

In the general vision of proposed framework, it may include a module for self-adaptation of the resources (see Figure 5). In this context self-adaptation is related to resources that may change over time in quantity and type according to some decision mechanisms. The self-adaption module, if implemented, belongs to the ROM.



**Fig. 5.** The Self-Adaptation Module

The self-adaptation process is triggered by the stakeholders' resource requests (e.g., requests for new allocations or the results of the trading process). The Self-Adaptation (SA) module computes a distance measure ( $d$ ) between the available resources ( $C$ ) of the system and the new allocation resource plan ( $P$ ) which would accommodate the stakeholders' requests. If this distance is lower than a given threshold  $d(C,P) < \delta$  then the system tries to adapt itself and modifies the available resources (whenever this is possible) in order to accommodate the stakeholders' requests. If the

distance  $d$  is greater than the given threshold, then the SA does not perform any adaptation. In any case, RA is notified of the SA results.

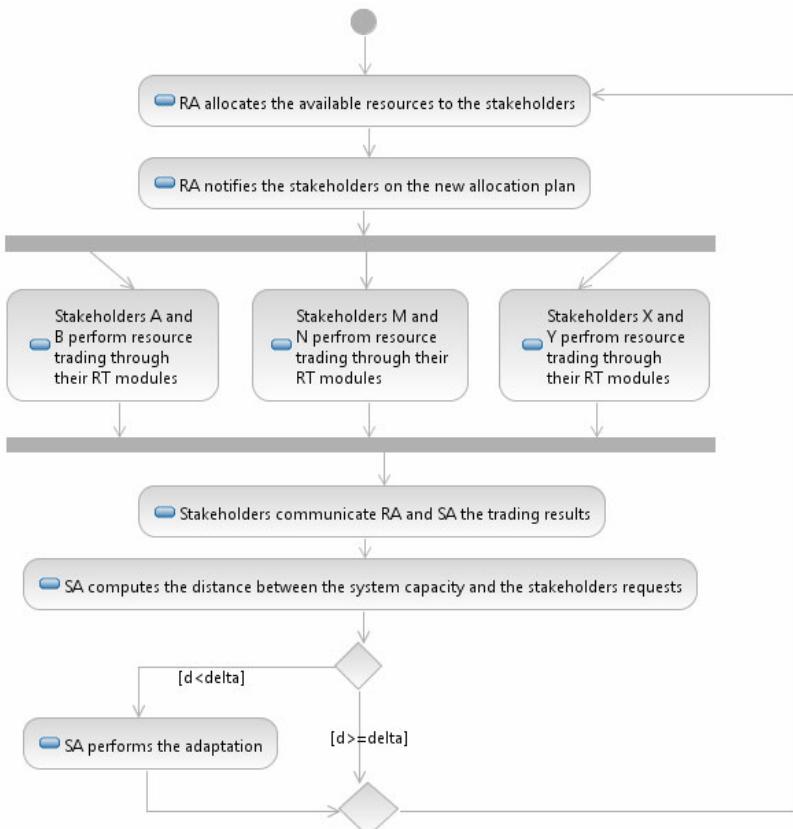
Also this module may exploit customized optimization models and methods [13].

The self-adaptation process is based on the MAPE (monitoring, analyzing, planning, and executing) feedback loop, which is exploited in the implementation of self-adaptive and self-management systems [6]. We address self-adaptation at the architectural level in an approach similar to the one proposed by Rainbow [9].

The described RT module is domain-independent and it can be used in different application domains.

## 2.4 The Interaction Workflow among the Modules for Resource Management

The workflow for the resource management framework is depicted in Figure 6. RA allocates the available resources to the stakeholders and notifies them the new resource allocation plan. The stakeholders may initiate a trading process with other stakeholders through their TR modules. This trading can be performed concurrently by the different stakeholders.



**Fig. 6.** The Interaction among the Modules of the Resource Management Framework

Once the trading process is over, the stakeholders communicate to the RA and the SA the results of the trading process consisting in variations of the previous resource allocation plan. SA computes the distance between the system capacity and the current requests of the stakeholders in order to decide whether adaptation would be effective or not. In the affirmative case, SA performs the adaptation of the resources.

If self-adaptation is performed, then the RA evaluates its results and updates the resource allocation plan based on these results. If no self-adaptation is performed, then the RA performs a re-allocation of resources trying to address as much as possible the stakeholders' requests (e.g., including the results of the trading process). The RA may require the stakeholders to resume the trading process by indicating also possible improvement hints in view of the global objective.

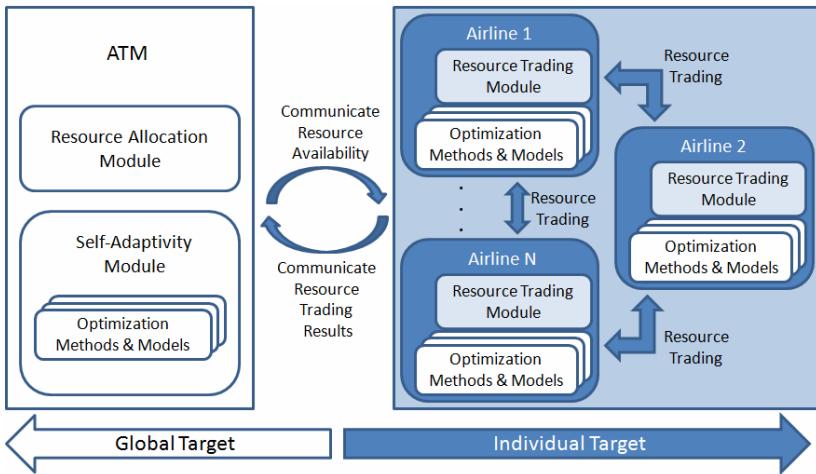
### 3 Application Examples

This section describes the application of our framework for resource management to two resource-constrained systems: the Air Traffic Management (ATM) and the Internetwork Service-Oriented Architecture (ISOA) systems.

#### 3.1 The ATM Example

An example of the application of our approach for resource management in resource-constrained systems is the Air Traffic Management (ATM) [8, 12, 20] system. In this application domain the resources are the capacity of the air traffic system elements, i.e., airports and airspace sectors. Capacity at airports is limited by the runway systems and the terminal airspace around them [14]. The capacity of en-route airspace sectors is limited by the maximum workload acceptable for air traffic controllers (measured as the average number of aircraft which are permitted to fly an en-route sector in a specific period of time) [12]. The main objective of the ATM system is safety, i.e., all flight operations have to be executed with no risk of collision. Due to the limited amount of capacity, imbalances between demand and capacity may occur at key times and points of the air transportation network. These local overloads create delays which propagate to other parts of the air network, amplifying congestion as increasing number of local capacity constraints come into play. Last year, more than 20% of US flights were delayed or cancelled (according to the US Bureau of Transportation Statistics). Similar statistics have been reported by European authorities. Airlines, which are the stakeholders of the ATM's resources, are very sensitive to a reduction of these delays and call for a wise use of the available resources. Specifically, they try to achieve their individual objective by gathering the required resources.

The proposed framework clearly fits to this application domain, and is consistent with the new decision paradigm of Collaborative Decision Making [20]. The ATM is ROM because it has the final decision on the resource allocation and has legal responsibility. The ATM receives as input information about the status of flights from the airlines, and tries to address airlines' requests. The main objective of airlines (besides the safety of the flights) is cost-effectiveness that is directly influenced by delays. However, the airline cost structure is sensitive information and is seldom disclosed outside.



**Fig. 7.** The ATM Application Example

The three modules, described in Section 2, are used in this case study as shown in Figure 7. The RA and SA belong to the ATM, while the TA belongs to the airlines.

The instantiation of the inputs and outputs in each of three modules in the context of ATM are shown in Table 1.

### 3.2 The ISOA Example

A further example of an application domain which needs resource allocation, trading, and self-adaptation is represented by Internetwork Service Oriented Architecture (ISOA), which is directly affected by the limitation of system resources [22]. The latter are considered for large-scale aggregation of distributed computing resources working together over the Internet, and should be able to dynamically adapt themselves (e.g., through reconfiguration mechanisms) in order to accommodate the requests for new allocations of servers' resources (e.g., CPU, memory and network bandwidth) to their services. Such new re-allocations could be triggered by several factors, e.g., dynamic changes of the runtime environment, current resources status, and/or variability of workload servers. A system should be able to predict its exact behavior, for example, as result of the impact of replacing one of its elementary services by analyzing what will be the impact to replace the service on the other services that will share its same resources (see, for example, [2], where it is remarked that more advanced and predictive models of adaptation are needed for systems that could fail to satisfy their requirements due to side-effects of change). If the system adaptation cannot be performed (e.g., for improving the efficiency of the resource allocation) we claim the cooperation between services (i.e. the resource trading mechanism). For example, a service provider could manage its high traffic load using additional resources provided by other providers and its unused resources could be offered to other providers (e.g., CPU-time and network bandwidth could be

re-negotiated among the providers). Besides, for improving the efficiency of the resource trading, the providers, could also re-negotiate the service-level agreements with their users for example.

The instantiation of the inputs and outputs in each of three modules are shown in Table 2.

**Table 1.** Inputs and Outputs in the Three Modules Applied to the ATM Case Study

<i>Input/Output</i>	<i>ATM Case Study Values</i>
<b>Resource Allocation</b>	
Resources	System capacity (Time slots)
Stakeholders	Airlines
Current Resource Allocation Plan	Official airline guides and revisions
Global Objective	Safety and throughput
Trading Results	Time slots exchange
Self-Adaptation Results	Modification of sectors capacity
Optimization Models/Methods	Customized integer programs, Assignment programs [7, 8]
Updated Resource Allocation Plan	Current scheduling of flights
<b>Resource Trading</b>	
Current Resource Allocation Plan	Current scheduling of flights
Application Domain Constraints	See for example [7, 8, 19]
Individual Objectives	Airline's costs
Resource Trading Requests from Other Stakeholders	Time slots exchange
Optimization Methods and Models	See for example [12]
Resource Requests to ROM	Time slots
Resource Trading Requests to Other Stakeholders	Time slots exchange
<b>Self-Adaptation</b>	
Current Resource Allocation Plan	Current scheduling of flights
Application Domain Constraints	See for example [7, 8, 19]
Global Objectives	Efficient use of available resources
Resource Requests from Stakeholders	Requested capacity
Optimization Methods and Models	See for example [13]
System Model	Reifications of the resources, stakeholders, constraints representations [16]
Self-Adaptation Results	Modified capacity of the air traffic system

**Table 2.** Inputs and Outputs in the Three Modules Applied to the ISOA Case Study

<i>Input/Output</i>	<i>ISOA Case Study Values</i>
<b>Resource Allocation</b>	
Resources	CPU, memory, network bandwidth
Stakeholders	Service providers
Current Resource Allocation Plan	Resources current allocation
Global Objective	Maximize the throughput of the multiple service workflows running on a server
Trading Results	Resources negotiation results and/or Service Level Agreements
Self-Adaptation Results	Modifications of the pool of resources and/or the time allocated for their usage
Optimization Models/Methods	Resource-Allocation-Throughput model [22]
Updated Resource Allocation Plan	Resources updated allocation
<b>Resource Trading</b>	
Current Resource Allocation Plan	Resources current allocation
Application Domain Constraints	Context-awareness, users' preferences, different classes of service consumers, efficiency, scalability
Individual Objectives	Satisfy users' quality of services requirements
Resource Trading Requests from Other Stakeholders	Resources negotiation and/or Service Level Agreements
Optimization Methods and Models	Service negotiation and SLA optimization models [10, 13, 18]
Resource Requests to ROM	Resource allocation, modification, and/or cancellation requests
Resource Trading Requests to Other Stakeholders	Resources negotiation requests and/or Service Level Agreements
<b>Self-Adaptation</b>	
Current Resource Allocation Plan	Resources current allocation
Application Domain Constraints	Context-awareness, users' preferences, different classes of service consumers, efficiency, scalability
Global Objectives	Maximize the throughput of the multiple service workflows running on a server
Resource Requests from Stakeholders	Resource allocation, modification, and/or cancellation requests
Optimization Methods and Models	See for example [13]
System Model	Reifications of the resources and service providers, constraints representations [16]
Self-Adaptation Results	Modification of the pool of the resources

Note that there are application examples in which only two of the three modules are necessary (e.g., multi-agents, e-learning, telecommunications, robotics, multimedia).

## 4 Conclusions and Future Work

This paper has presented a framework for dynamic allocation, trading, and self-adaptation of resources. The proposed framework is remarkable for integrating all the described aspects of the resource management. The framework can be applied in various application domains. This paper has shown its application in two different domains: ATM and Internet SOA. Being a modular framework, its functionalities can be used together, separately, or in combination. For each of these functionality, there have been indicated the inputs which should be specified when exploiting the framework, and the outputs to be expected. The interaction among the three modules has been shown through an activity diagram.

Future work concerns the enhancement of the proposed framework with business intelligence mechanisms in order to allow the storage of the solutions adopted in various situations, and the application of these solutions in similar situations in order to avoid computation overheads (when possible). A prototype implementing our approach is currently under development. We aim to apply this prototype on realistic examples to validate and improve its functionalities, as well as to enhance it for achieving specific properties of various application domains. We plan to compare our prototype to existing resource management systems through the same case studies in order to better outline its novelties. We also intend to introduce evaluation criteria and metrics for the performance measurements.

**Acknowledgments.** This work is partially supported by the *Enhancing the European Air Transportation System* EATS research project funded by the Italian Government within the PRIN program.

## References

1. Aldewereld, H., Buzing, P., Jonker, G.: Multi-Agent Plan Diagnosis and Negotiated Repair. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (2008)
2. Andersson, J., de Lemos, R., Malek, S., Weyns, D.: Modeling Dimensions of Self-Adaptive Software Systems. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 27–47. Springer, Heidelberg (2009)
3. Boone, B., Hoecke, S.V., van Segbroeck, G., Joncheere, N., Jonckers, V., Turck, F.D., Develder, C., Dhoedt, B.: Salsa: Qos-Aware Load Balancing for Autonomous Service Brokering. Journal of Systems and Software 83(3), 446–456 (2010)
4. Buccharone, A., Cappiello, C., Di Nitto, E., Kazhamiakin, R., Mazza, V., Pistore, M.: Design for Adaptation of Service-based Applications: Main Issues and Requirements. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 467–476. Springer, Heidelberg (2010)

5. Burgkhardt, D., Cosovic, I., Jondral, F.: Dynamic Spectrum Allocation by Hierarchical Resource Trading. In: VTC Spring, pp. 1796–1800. IEEE Press, Los Alamitos (2008)
6. Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J.: Software engineering for self-adaptive systems: A research roadmap. In: Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 1–26. Springer, Heidelberg (2009)
7. Dell'Olmo, P., Lulli, G.: A Dynamic Programming Approach for the Airport Capacity Allocation Problem. *IMA Journal of Management Mathematics* 14, 235–249 (2003)
8. Dell'Olmo, P., Lulli, G.: A New Hierarchical Architecture for Air Traffic Management: Optimisation of Airway's Capacity in a Free Flight Scenario. *European Journal of Operational Research* 144(1), 179–193 (2003)
9. Garlan, D., Cheng, S.-W., Huang, A.C., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-based Self-Adaptation with Reusable Infrastructure. *IEEE Computer* 37(10), 46–54 (2004)
10. Li, C., Li, L.: Joint Optimization of Resource Allocation and User QoS Satisfaction Control for Efficient Grid Resource Management and Scheduling. *IEEE Systems Journal* 3(1), 65–77 (2009)
11. Luo, Z., Li, J.S.: A Web Service Provisioning Optimization Model in Web Services Community. In: Proceedings of the IEEE Conference on e-Business Engineering, pp. 689–696 (2005)
12. Majumdar, A.: Understanding En-RouteSector Capacity in Europe. In: European Air Traffic Management: Principles, Practice and Research, pp. 65–95. AJ Cook, Ashgate Publishing Limited, Hampshire (2007)
13. Mirandola, R., Potena, P.: Self-Adaptation of Service-based Systems on Cost-Quality Attributes Trade-Offs. In: Proceedings of the 1st Workshop on Software Services, co-located with the 12 th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania (2010)
14. Hansman, R.J., Odoni, A.: Air Traffic Control. In: Belobaba, P., Odoni, A., Barnhart, C. (eds.) *The Global Airline Industry*, pp. 377–403. Wiley, Chichester (2009)
15. Raibulet, C.: Facets of Adaptivity. In: Morrison, R., Balasubramaniam, D., Falkner, K. (eds.) *ECSA 2008*. LNCS, vol. 5292, pp. 342–345. Springer, Heidelberg (2008)
16. Raibulet, C., Arcelli, F., Mussino, S., Riva, M., Tisato, F., Ubezio, L.: Components in an Adaptive and QoS-based Architecture. In: ICSE 2006 Workshop on Software Engineering for Adaptive and Self-Managing Systems, pp. 65–71. IEEE Press, Los Alamitos (2006)
17. Raibulet, C., Massarelli, M.: Managing Non-Functional Aspects in SOA through SLA. In: Proceedings of the First IEEE International Workshop on Engineering Non-Functional INformation for Emerging Systems, pp. 701–705 (2008)
18. Repantis, T., Drougas, Y., Kalogeraki, V.: Adaptive Component Composition and Load Balancing for Distributed Stream Processing Applications. *Peer-to-Peer Networking and Applications* 2(1), 60–74 (2009)
19. Vossen, T., Ball, M.O.: Slot Trading Opportunities in Collaborative Ground Delay Programs. *Transportation Science* 40(1), 29–43 (2006)
20. Wambsganss, M.C.: Collaborative Decision Making Through Dynamic Information Transfer. *Air Traffic Control Quarterly* 4, 107–123 (1996)
21. Webb, A., Sarkani, S., Mazzuchi, T.: Resource Allocation for Air Traffic Controllers using Dynamic Airspace Configuration. In: Proceedings of the World Congress on Engineering and Computer Science (2009)
22. Yau, S., An, H.: Adaptive Resource Allocation for Service-based Systems. *International Journal of Software and Informatics* 3(4), 483–499 (2009)