



## Nexus Services in Smart City Ecosystems

Paraskevi Tsoutsas<sup>1</sup> · Panos Fitsilis<sup>2</sup> · Leonidas Anthopoulos<sup>3</sup> · Omiros Ragos<sup>1</sup>

Published online: 20 January 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

### Abstract

According to market research, the number of smart cities is increasing rapidly. Information and communication technologies (ICT) provide the smart infrastructure that is the foundation for all the key themes related to a smart city, such as smart economy, smart governance, smart mobility, smart health, smart buildings, and smart water. As such, a smart city is constituted of various infrastructure components that form a complex system of systems, which is essential to collaborate effectively. Services play a central role in this vision of smart cities, as they are used as building blocks for effective collaboration, i.e., to achieve interoperability between heterogeneous parties and independence from the underlying infrastructure. In order to cope with the problem of complexity and the scalability in smart cities' systems, a solution is to provide autonomous, collaborating services that have situation awareness and are able to adapt dynamically to the changing needs of the environment forming a nexus of services. In this research, we propose a nexus model for smart cities' services collaboration by using the role modeling approach enhanced by the introduction of service teamwork roles. The teamwork roles definition is inspired both by human and agent team working models. We determine the dominant teamwork roles that prevail during service group cooperation, where the main goal of each role is to intervene and "act as a connector" during collaboration to keep the team of component services consistent with the goal of the group–team. The teamwork functionality is applied through the introduction of a new layer in the architecture of smart cities and is exploited to overcome some of the aforementioned problems. A case study that presents how teamwork roles could affect and benefit the service collaboration in a smart city environment is also provided.

**Keywords** Services · Smart cities · Service choreography · Role modeling · Teamworking

---

✉ Paraskevi Tsoutsas  
tsoutsas@teilar.gr

## Introduction

According to market research, the number of smart cities is increasing rapidly. This growth is expected to continue, since smart cities has grown from approximately two dozen to more than 100, with more than 600 cities to work on getting smart in the next years (Maddox, 2018).

While there is no universal definition for the smart city, an often used definition is that of International Telecommunication Union (ITU) report in 2014, which is the following: “A smart sustainable city is an innovative city that uses Information and Communication Technologies (ICT) and other means to improve quality of life, efficiency of urban operation and competitiveness of services, while ensuring that it meets the needs of present and future generations with respect to economic, social and environmental aspects.” (Kondepudi et al., 2014).

ICT provides the smart infrastructure that is the foundation for all of the key themes related to a smart city, such as smart economy, smart people, smart governance, smart mobility, smart health, smart buildings, and smart water. As such, a city is constituted of various infrastructure components that form a system of systems (SoS) (Anthopoulos 2017).

According to Mayk and Madni (2006), a SoS is a collection of systems that were originally designed as separate systems for specific and different purposes but they have been brought together within the SoS umbrella, in our case a smart city umbrella, for creating new capabilities required for the mission of a smart city. A smart city as a system is usually designed to accommodate various and diverse services, though not all services and their capabilities are defined precisely at least at the time of the initial deployment, and consequently, they are not included in the initial design. Further, since a smart city evolves dynamically, it could incorporate new enhanced and innovative services. This evolution includes continuous deployment of new services, reorganization of existing services, and automatic development of novel more competitive services. Commonly, a smart city is characterized using terms, such as interoperable, synergistic, distributed, adaptable, inter-domain, reconfigurable, and heterogeneous. Actually, it is an ecosystem that evolves over its lifetime.

Furthermore, the key characteristics of SoS (Maier 1998; Sage and Cuppan 2001), which in our case are focused to a smart city, should be a) operational independence of elements, b) managerial independence of elements, c) evolutionary development, and d) emergent behavior and geographic distribution. Although the above are quite important, however, another characteristic should be included, that of efficient collaboration, since cities’ electronic habitants are increasing rapidly with the massive introduction of internet of things (IoT) technologies.

Collaboration of smart infrastructure will provide the cognitive framework that smart cities subsystems need, in order to have the ability to cooperate transparently and autonomously for offering composite, complex, and more efficient services. We define this smart collaboration city infrastructure as the city nexus. However, developing smart city nexus is not straightforward, since, according to Vlacheas et al. (2013), while a wide set of predefined services and applications is available for employment in smart cities, there are technological barriers among objects when they are used across application domains.

Services are commonly considered for performing complex tasks, thus research has focused on the problem of the automatic service composition. An automatic and dynamic web service composition is a highly complex process, and the proposed

standards of related technology do not answer holistically the problems of web services discovery and composition yet (Sheng et al. 2014; Jula, Sundararajan and Othman 2014; Tosi and Morasca, 2015). Further, service composition and collaboration are limited to approaches that groups of services follow a plan for the composition taking into account only their functional behavior, and this cannot be characterized as collaboration. In many cases, services need to work together as a group to achieve common objectives, implying teamwork abilities, which are commonly used in the case of groups of humans, agents, or autonomous vehicles. Developing autonomously collaborative services, capable of exhibiting teamwork behavior, that would have situation awareness and adapt in the environment of a smart city, is a challenge for our research. A service capable of exhibiting teamwork behavior is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks and that is able to intervene and catch errors or prevent emergent behaviors that will put in danger the overall team goal, i.e., the overall execution of the composite service itself.

Teamwork has become an important research field, and its contribution to organizational performance has attracted attention of various research groups from several disciplines. In recent years, many scientists studied why humans succeed or fail in joint activities and a variety of models have been developed that follow social–psychological approaches for human team formation. Apart from human team working, another area where similar problems have been studied and such theories have been applied, relevant with this research, is the area of autonomous software agents and robots. As with humans, a group of autonomous software agents must accomplish given tasks by organizing themselves according to their individual characteristics and their teamwork behavior within the overall system.

Toward this direction, we exploit the role-modeling approach and the definition of behaviors to create roles that, except their functional behavior, will also have teamwork behavior. Smart city nexus could be modeled using this approach, which directs the team participants to cooperate in order to reach a goal. The focus is on architecting cooperative teams of services that form composite services, where the cooperative team is meant in a stronger sense than a composite service, which usually simply implies a set of bounded services. This teamwork hypostasis of cooperation in services makes the concept of collaboration stronger.

We use the term *Smart City Nexus Services* (SCNS) to differentiate the typical composite services that could isolatedly be produced in the web, by compositions that take place in a role domain having a teamwork attitude that is specific oriented to the goals of the smart city domain. According to Oxford dictionary, nexus is “*a complicated series of connections between different things.*” The concept of nexus for modeling location awareness software applications was used by Nicklas and Mitschang (2004) and as well by other researchers but in different contexts (Ananthakrishnan et al. 2013; Lange et al. 2009). In SCNS, the final product is a combination of services and applications’ internal data with externally sourced data and services bounded by teamwork services that act as a glue to produce nexus services and aim to serve smart city’s goals. They integrate resources of the domain software information systems, e.g., e-governance, with data of IoT services and other sophisticated tools in a sense that services form these systems could also participate to play a critical role during the composition process. A SCNS has the ability to come up with the requested goal or other similar to the requested one. If a goal is not achievable using city’s own resources,

this city has to employ other services in achieving this goal. To successfully do so, the designed composite service needs to be enabled with one or more of teamwork behaviors.

The problem we address in this paper is to exploit the existing architectures and augment it with a teamwork layer that can introduce the notion of teamwork collaboration within a set of services, agents, and other systems that “live” in a smart city forming the smart city nexus. This is quite important since smart cities are becoming more and more complex; they are composed of a number of interacting subsystems; the number of offered services is increasing rapidly making the management of the smart city cumbersome and costly. Therefore, there is a clear need to add autonomy, automation, and self-organization features to smart cities’ services in a fashion that resembles human teamworking. As such, the research questions addressed in this paper are:

- (i) which are the teamwork behaviors needed in a SoS such as a smart city,
- (ii) how we can incorporate a teamwork software layer within the reference architecture of a smart city, and
- (iii) how we can benefit from the introduction of such teamworking behavior in a smart city?

The rest of this paper is organized as follows. Section II describes the background and related work. In section III, we present the proposed role model and the proposed teamwork behaviors. In section IV, we present the teamwork layer for smart cities, and in section IV, we present an example on how this model can be applied within a smart city environment, while the last section presents conclusion, limitations, and ongoing work.

## Background

### Teamwork Theory

A smart city according to Hall (2000) is “A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, event major buildings, can better optimize its resources, plan its preventive maintenance activities and monitor security aspects while maximizing services to its citizens”. Obviously, smart cities’ environment is complex, that work is performed by teams composed of members that are either humans or IoT devices and autonomous software agents where each of them is specialized in specific tasks. For example, teams of humans, vehicles, and software services need to cooperate in various transportation scenarios (Schwartz et al. 2016), while humans, together with various smart city cooperative agents offered by various providers implementing different protocols, need to collaborate. As such, the success of a smart city use case depends more on collaborative rather than on individual efforts. In other words, a smart city is an ecosystem, where organization, businesses, human’s operators, users, IoT services, and cyber physical systems (CPS) should be effortlessly connected while exhibiting teaming behavioral attitudes. In order for these entities to be

effectively connected and collaborate as a team, models based on human team working literature were investigated and exploited.

Team working has been thoroughly studied by psychologists and human resource experts over the last decades. In early studies, some researchers have proposed models focused on specific characteristics that team members should have, such as personality, functional expertise, competencies, and goal orientations. (Rousseau, Aubé and Savoie, 2006; Salas, Cooke and Rosen 2008). As the team research matured, research moved firmly from dealing with single characteristics that members should have to a variety of behaviors that members should expose (Kozlowski et al. 2015; Berlin, Carlström and Sandberg 2012). Ultimately, in current research, member's tasks and behaviors are often clustered into distinct roles within the system that are aligned with the expertise of each team member (Mathieu et al. 2015). Increasingly, researchers propose that teammates, along with the operational tasks that they perform in a team, also have to play some other teamwork role, such that of coordinator, contributor, and idea generator. Indeed, as it is witnessed by empirical studies, these approaches are effective in a variety of contexts, tasks, and domains.

There were many attempts to classify roles and behaviors in the context of human team working. Among them, the most prominent models are (i) Belbin's work in team roles (Belbin 1993); (ii) Parker's set of team player styles (Parker 2008); and (iii) Margerison and McCann model (Margerison and McCann 1990). The last model defines eight different roles, namely (a) explorer–promoter, (b) assessor–developer, (c) thruster–organizer, (d) concluder–producer, (e) checker–inspector, (f) upholder–maintainer, (g) reporter–advisor, and (h) creator–innovator. Each role is linked with predefined behavior and tasks. For example, “Creator–innovator” role is linked with forward thinking, new ideas, and new ways of doing things. People playing this role in a team come up with new strategies and different approaches to tasks, creating and experimenting with new ideas in order to handle various situations and challenges.

The various agent-based frameworks of team behavior proposed in the literature were also investigated to identify and analyze the different agent teamwork factors among various types of teams in related areas. Generally, agent-based teamwork factors address different collaboration attributes, such as (i) how the team is organizing itself, e.g., by creating rules for collaboration and communication; (ii) how the team is forming its strategy for future direction, e.g., by planning and decision making; and (iii) how the team work together to achieve synergy, e.g., by following rules of trust and engagement. Fifteen primary factors are revealed from agent-based models in the literature, which merit particular attention across different team tasks and group sizes. Following the analyses of Tsoutsas et al. (2019), we present the three dimensions and how the teamwork factors are clustered to:

- Organizing factors, such as collaboration, communication, and coordination.
- Strategy-related factors, such as planning, learning, decision making, evaluation, and teamwork policies.
- Synergy related factors, such as ad hoc team setting, autonomy, delegation, joint intention, knowledge of teammates' capabilities, knowledge sharing, and trust.

The framework for the goals of this research paper is partially based on existing approaches in service modeling and teamwork behavior. However, the primary

objective for smart cities is not only to provide services coming from different smart systems but also to combine them through basic communication and collaboration. According to our point of view, the benefit comes through the ability to bind them in novel services that incorporate the abilities of the composed services or systems; this way, they could implement a more efficient composition, through the collective intelligence they gain from the domain during the execution, while exploiting the teamwork behavior of participants of the team which contribute when errors occur that lead to abnormal situations.

## Services Composition

Currently, in large complex systems, services are composed by other services. Therefore, service composition or service orchestration (composition without central control) is an important aspect of service provisioning. Our proposed teamwork layer should work seamlessly with current services offering an enhanced smart city teamworking ecosystem, and as such, it is based on service composition/orchestration techniques.

Service composition approaches range from practical languages aspiring to become industrial standards (e.g., Business Process Execution Language (BPEL) and ontology of services (OWL-S)) to more theoretical models and languages (e.g., automata, Petri nets, and process algebras). The problem of service composition is the problem of constructing a new composite service, by using available services that perform some desired functionality, when no single service satisfies it, which is quite important in a SoS environment such as a smart city.

According to Nacer and Aissani (2014), web service composition is a technique which assembles web services in order to achieve a particular goal, via primitives of control and exchange. Apparently, this problem belongs in an evolving area, since software is structured more and more often as a service and services are operating in an open, changing and distributed environment where control is loose. Therefore, the problem of composition and collaboration in many cases is the same, since similar requirements apply. Many reviews on web service composition and their classification survey, e.g., manual vs. automated and static vs. dynamic, among others have been published in Bano and Ikram (2010), Khadka and Sapkota (2010), Patel and Shah (2016), and Sheng et al. (2014), where researchers compare composition methods according to the level of automation or on use on standards for web service composition. For example, the review of Sheng et al. present an analysis of existing major techniques, research prototypes, and standards on web services composition, while the work of Patel and Shah indicates more open and unresolved research challenges in the area. The need to compose services from disparate and heterogeneous environments initiated a number of research efforts both in industry and in academia.

Regarding the industry approaches, some standards that determine the composition as executable workflows are Business Process Execution Language for Web Services (BPEL4WS) and Business Process Modeling Language (BPML). BPEL provides a language for the formal specification of business processes and business interaction protocols. BPEL4WS, like DAMLS service model is are focused on representing service compositions, where a process flow and bindings between services are known a priori. BPML is based on calculus and, like XLANG, supports semantics such as nested processes and complex transactions that are not addressed by BPEL4WS.

Academic research activities have also been resolved by various models, and a criterion for clustering that is used in general is to focus on issues related to the functional or non-functional properties of the composition. The first category can be divided more to approaches considering only the syntactic description and to approaches considering semantic description. These usually deal also with a specific model of interaction such as graph theory, artificial intelligence planning approaches, workflow, matching, and logic-based approaches Rao et al. (2004); Berardi et al. (2005); Baryannis (2014), Hatzi et al. (2015) propose composition methods among other studies in this category.

## Smart City Architectures

Smart cities are distributed computing environments composed of a large number of software services. This architecture enables the continuous evolution of smart city ecosystems. Therefore, a key quality of a smart city architecture is its ability to accommodate new services by automatically composing and executing them as novel software services. This service composition process is critical, since in many cases, there are interrelationships between city's core systems, given that these systems cannot work in isolation. On the contrary, it is quite common to operate in close collaboration, e.g., smart transport network relies on traffic management. Interconnecting these systems obviously improve their efficiency and intelligence. Numerous examples of such synergies exist, e.g., smart water–smart energy and smart energy–smart buildings.

Service composition in most cases is based on service orchestration, which is a basic concept of service-oriented architectures (SOA). Service orchestration is a centralized approach for composing services out of existing atomic software services. However, service orchestration is better suited to static environments, when there exists a coordinator, and plans are known in advance while minimal changes happen during the execution. Nevertheless, this is not the case in smart city ecosystems, where changes are introduced frequently and the number and the type of offered services are not upfront defined. For such cases, service choreography is a preferred solution, since composition of services is done on a peer-to-peer fashion, leading to autonomously operating services. This need was early identified, thus choreographies were included in business process model and notation (BPMN 2.0) specifications (Allweyer, 2016).

As already mentioned, in the smart city, case myriads of heterogeneous services operate independently. This trend will grow even more in the near future, since smart cities need to provide even richer behavior and functionality. According to ITU Telecommunication Standardization Sector (ITU-T) (ITU-T 2015), the overall smart city architecture should provide support for transportation services, e-government services, e-business services, safety and emergency services, smart health services, tourism services, education services, smart buildings, waste management services, smart energy services, and smart water services.

The only way to keep control of such complex systems is by developing services that are able to act and interact independently and on a demand basis. The development of smart cities' systems and applications demands software development environments that are able to support a number of functional and non-functional requirements. The functional requirements that need to be satisfied by a development environment, as they are described by in Santana et al. (2017), are:

- Data management, which includes collection, storage, analysis, and visualization of city data.
- Application run-time support for facilitating deployment and integration of smart cities' applications.
- Sensor network data management and control.
- Data analytics functionality for analyzing massive volumes of data produced by a smart city.
- Service management according to SOA or other service management standards.
- Tools for conceptually defining smart cities models, organizations, etc.

Furthermore, a number of non-functional requirements need to be introduced to enrich the existing list, related mainly but not exclusively to the quality characteristics of the provided services. More specifically, some of the existing non-functional requirements are interoperability, scalability, security, privacy, configurability, etc.

According to ITU-T (ITU-T 2015), as shown in Fig. 1, the smart and sustainable city (SSC) architecture should be layered and it consists of a) the sensing layer; b) the network layer; c) the application layer; and d) the operation, administration, maintenance and provisioning, and security (OAM & P & Security). Many such frameworks have been presented in the literature, focusing on different architectural aspects. For example, in the work of Gaur et al., (2015), a software framework is presented that includes a semantic layer, which enables exploitation of domain-specific data based on the concepts and relationships between these concepts.

Anthopoulos and Fitsilis (2014) explored various smart cities around the world and concluded that the architecture that is preferred by well-managed cases is the multi-tier architecture, which is applied in new, existing, and smart planting cases, while it addresses both soft and hard infrastructure, and it considers natural environment and the evolving IoT in terms of sensor installation.

Apparently, architectures that used to be operational in the near past have deficiencies and seem to be inadequate for the future. For example, they are tuned for static service provisioning but not for dynamic service composition, they are controlled in most cases centrally, and they do not allow ad hoc collaboration between services, while smart city subsystems are interconnected and integrated without exhibiting any signs of intelligent behavior (situation awareness, adaptive behavior), etc. In order to overcome these deficiencies, in the sequel, we present the introduction of a new layer, which will provide teamwork functionality and could be utilized to solve some of the abovementioned problems.

## Role Modeling of a Smart City Domain and Teamworking Roles

### The Role Model

A smart city provides an intelligent way to manage components such as energy, transport, schools, and buildings on their own, but what if these systems could interoperate in a domain designed to serve communication for specific purposes?

For the effective exploitation of assets and services in a smart city nexus, a continuous flow of information is needed in order for services, when executed, to be



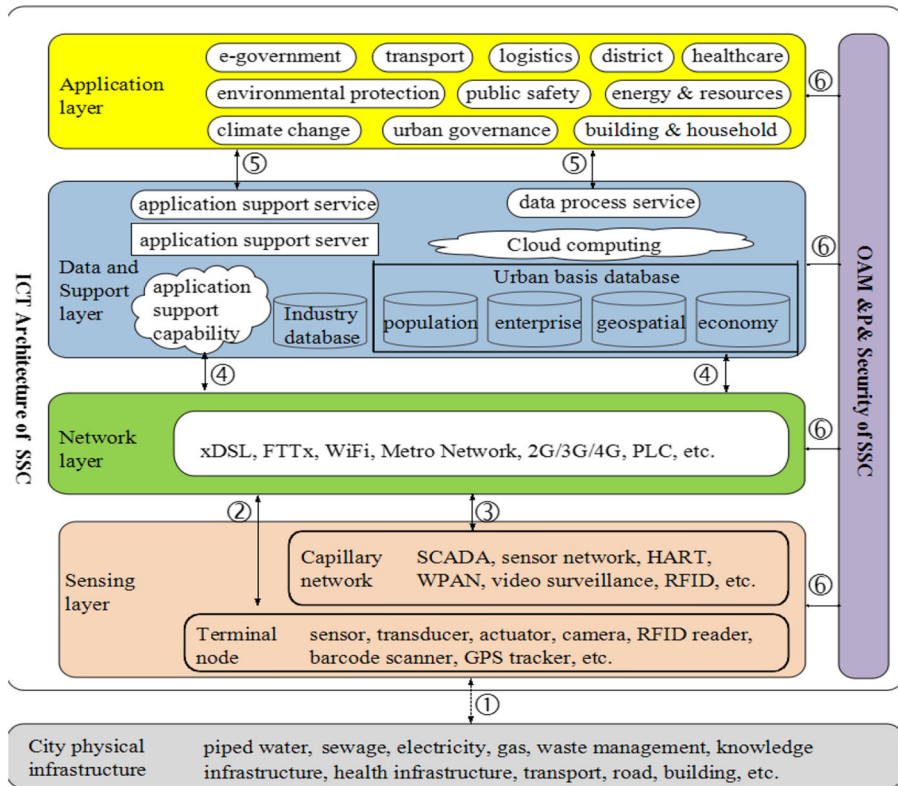


Fig. 1 Smart and Sustainable City Architecture ITU-T (ITU-T 2015)

adapted to the city changes that happen in realistic environments. Through the perpetual reception of data indicating the behavior of participants in the city domain, local decision-makers could be informed about events of their concern, e.g., that an asset or a service in a specific area is hardly used and could be better exploited, by possibly changing some constraints for using it or by moving it to another area. Similarly, by knowing the hours that the employees of a specific building are coming and working could aim in optimizing the energy needed in this building, while predicting with more accuracy the parking seats they need or the bus routes they could use will improve their daily commute to work. Such everyday processes that take place in a smart city are required to be adapted to everyday needs.

To address the challenges, we represent and solve the composition problem through role modeling (Tsoutsas et al. 2017). Modeling services to a *smart city role domain* could add value to the composite services derived to implement specific domain processes. We exploit the applicable concepts and technologies offered for automated composition and further enhance services by enabling teamwork behaviors to the above entities. Role modeling has the advantage to model both the operational and teamwork behavior of participants.

In order to present the role modeling in a formal way, we have constructed the UML class diagram. In Fig. 2, we present the role model specification to depict how objects

constituting the system are interrelated. The main concepts used by the role modeling approach are *roles*, *communicative actions*, and *behaviors*. A communicative action (ca) is a type of interaction between two roles and expresses, encoded, the reason for interaction, e.g., acceptOrder, fail, inform. A behavior expresses a potential collaboration between roles that mean to communicate using the communicative action. A role expresses a behavior to a peer role, while a participant, which could be a service, a human, or an agent, plays a role that express a behavior. The way that a role interacts with another role is considered as a behavior of the first role. A role expresses different aspects of collaboration to other roles through different behaviors. Therefore, a role is modeled as a set of the expected behaviors it exhibits, which states the interactions in which it is potentially involved. Although the RM diagram presents the possible interaction between roles through the execution of the appropriate behavior, we need, additionally, to state explicitly under which circumstances the role has the ability to execute a behavior and, when the behavior is executed, which are the changes that happen. These are expressed as *sentences* and are the precondition and effects, respectively, of the *transition* that happened in the domain, when a behavior is executed. Transition and sentences are expressed in a formal system, which enable reasoning over state transitions.

In order to model a smart city domain as a role domain, we first need to define the objectives of it. According to the role modeling approach, all domain goals identified are organized in an ontology. From domain goals derive, the communicative intends and abstracts behaviors are partitioned to roles of the domain. The communication of roles with other peer roles constitutes the behavior of each role. Once a role communication is defined, the preconditions and effects of each communication to be executed are stated. Having all the details to design the role model of the domain, we facilitate the role model with the participant services. Available services are enrolled to the domain, and for each service, the roles it plays are determined. For this reason, extra details or attributes and their meanings (semantics) must be identified by name, data type, constraints, and default values. Consequently, each service is mapped to create a relation between the service and the roles to be played. In the next step for the utilization of the role model domain, solution design could be created via finding plans that correspond to domain goals. Since the process of service enrolment to the semantic domain is completed, different solution designs that correspond to use cases could be created as a sequence of paths of behaviors that could be executed to accomplish given domain goals. All the solutions are stored as behavior paths in the plan library.

### Teamworking Roles

Even though the role model is an abstraction that describes the patterns of interactions among a set of entities, our intention is to introduce specialized teamwork roles that intervene when particular behaviors take place, for infusing team working behavior within smart city services.

Motivated by the organizational behavior and the teamwork theory (Margerson and McCann 1990; Belbin 1993), in order to bring teamwork abilities to smart city service composition, we argue that services similar to agents and robots, except from functional behavior, they also have to exhibit teamwork behavior when they cooperate. In case

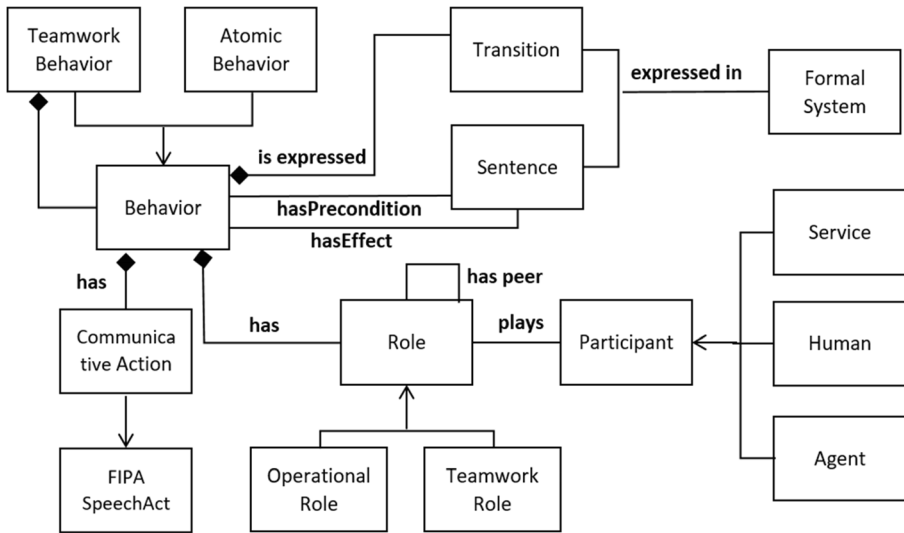


Fig. 2 The UML class diagram for the role model

that no member services exhibit such behavior, then other services should be included into the composition in order to provide this behavior. This behavior of the composite service will be beyond the individuals capabilities. Following it, it exploits its ability to better cooperate since reactive systems in the real-world do not always work as planned.

To achieve this and based on the role-model paradigm for services presented, we introduce a teamwork role model enhanced by service teamwork roles. We propose eight additional service roles that are teamwork to be involved during the composition process, which are *developer, organizer, promoter, inspector-monitor, producer, advisor, innovator, and maintainer*. The eight roles were derived by combining the teamwork theory focusing on the behavior of high-performing human teams and the fifteen teamwork factors emerged from agent team working, which are actually the behavioral descriptors for these roles. The enriched model can be proved beneficial to create more efficient and stable teams of services, i.e., composite services such that their components—members promote better understanding, and much more likely, they are able to cooperate effectively and achieve their objectives. The roles involved in the case study that follows are briefly described below while detailed description is provided in (Fitsilis et al. 2018):

**Developer Role**

During the composition of services, the developer role finds a plan and in cooperation with other teamwork roles of the domain, qualifies the development of teamwork-enabled composed services. The result is a composite service that is generated by establishing teamwork attitude during its execution. The team development involves everything that groups do to be formed and operate, i.e., how they come together, how they evolve and dissolve, and the different phases participants can go through.

## **Organizer Role**

Organizer role is a role needed for organizing and monitoring the execution of various activities. It is possible that different services are assigned both the roles of developer and organizer; however, these services should be strongly coupled.

Similarly, the inspector role is the role that monitors the execution of the plan within the domain of a smart city. It is responsible to keep track of the progress, to inform organizer for possible delays and to trigger the developer when a new plan re-scheduling is required.

The promoter role is a role that enables interacting with different domains, e.g., by promoting services offered outside the domain of specific smart cities. A service having this role should be aware of the capabilities of all services of the domain. This is done with the aim to expand the offered services of a smart city or to look about potential new services that are in demand.

## **Producer**

Producer is a generic role that is assigned to all offering added value services to the citizens within the domain of a smart city. Therefore, the producer role works on a systematic way to produce work outputs. Just like a factory in the real world, the purpose of services for playing these role is to produce. It is the doer of the system; this role characterizes, for example, all smart city services. It is invoked by the monitor role to participate in a case. In general, it gives feedback to the promoter who searches for opportunities to promote their usage.

## **Advisor**

The advisor role is needed for exploring new alternatives to develop the offered services within the domain of a smart city. Through history execution of services in the domain, it can reason and evaluate a set of services that will make the team formation. In a more advanced form, the advisor is based on semantic constructs such as domain ontologies in order to offer better reasoning using knowledge retrieval algorithms and provide recommendations of better quality.

## **Maintainer Role**

The main goal of the maintainer is to ensure that standards and processes of the domain are upheld and to maintain team functionality. It checks for new updates for the system and keeps its configuration.

## **Innovator Role**

The innovator role could be assigned to all experimental services of the domain or to services running at a test system, e.g., a landscape that has not been deployed yet.

Therefore, a smart city domain, except form the operational roles, also contains teamwork roles that contribute to its functionality. Upon a user request for a service, the

advisor role intervenes in order to interact with clients and understand their needs. It takes the request, collects all the requirements and checks if it matches according to its semantic similarity with existing goals of the domain repository. If it is so, it retrieves a plan and provides it to the developer role to proceed with the service allocation. Otherwise, it informs the developer role to search for a new plan, i.e., a sequence of behaviors with the given preconditions and effects. The developer role extracts from the plan the sequence of behaviors and maps the appropriate services for the service composition. It is the “team builder” of the domain; it makes an evaluation to find the appropriate set of services as a final product that could become a team to execute the composite service. In the sequence, it calls the monitor role to supervise the execution of the plan.

At this point, we should highlight that our model is an attempt to model the teamworking behavior within a domain, and its aim is to address important problems that real systems have to handle and resolve such as security, trust, computability, and availability of resources.

For example, if a service has access rights to a dataset, can this right be delegated to another service of the domain for improving performance of the system?

Similarly, if we consider the advisor role, it has as responsibility finding appropriate plans for a new or alternate service composition, if exist, that their goals have matching based measures which are better compatible to the request or to benefit from domain knowledge and advising team members in order to ensure more successful group communication. In order to advice, it needs to know the computation time needed for a specific request which in the general case is not known. Therefore, we should limit “advices” only to queries that we know in advance the computational time needed.

In the next section, we present a specific case study of a smart city domain.

## The Teamwork Layer

In Fitsilis et al. (2018), it was presented the basic components of the architecture, since specific functional requirements do not exist; they are generic and not analytical for a particular smart city implementation. What we discuss here is best described as a conceptual application architecture layer, capturing the most prominent requirements for exhibiting an intelligent teamwork-collaborative behavior to smart cities’ applications. However, this layer could be modified and transformed as necessary, to address all specific functional and quality requirements for specific smart city frameworks or projects.

The teamwork layer, ideally, could be used with the role-based modeling approach, for presenting the services/components that cooperate within the smart city ecosystem. According to this approach, for a smart city/domain, the role model of the domain is defined. The role model contains all roles and behaviors of the domain; each role is a set of behaviors. The set of roles that each service acquires implies paths for the collaboration of the service with other participants of the domain. Defining services as set of roles allows an abstraction and helps to capture other entities that might exist and cooperate in the domain to form heterogeneous teams, e.g., teams of services, agents, and robots. This is also necessary since, although some services may have the

same functional requirements, not all services can exhibit the same behavior concerning non-functional requirements. Role modeling of services and its application to model composition are discussed by Tsoutsas et al. (2017).

One of the main concerns in this model is for participants of the smart city to improve the collaboration of the composed systems when they cooperate. Services enrolled within the same domain will be aware of this fact and will be able to share knowledge artifacts of the domain existing in various forms and formats. Further, through the role of the *advisor*, services are informed about other services that have similar behavior; their capacity, which of them are trusted; their functioning within the system in relation to the smart city's semantic model (e.g., smart city ontology); and the geospatial information of each service (e.g., which smart building is in the proximity of a smart vehicle). The satisfaction of these requirements would lead to the implementation of important operations within a team such as *delegation, common intention, and knowledge sharing behavior*.

The implementation of the SCNS essentially follows the three-step implementation, which is:

- a) Service design phase,
- b) Service composition phase, and
- c) Service execution phase.

During the service design phase, a choreography could be designed using a business process modeling language, such as BPMN 2.0. The design is done after capturing the service requirements in collaboration with domain experts and users. Service composition could be done using BPMN 2.0 and after discovering the appropriate service instances needed. At this phase, the planer cooperates with the advisor role to provide a plan for the composition. This is done in order to extract and evaluate the set of services that are enrolled and are available within the domain. For describing the service plan, the behavior, the interactions, and message exchanges need to be specified. This is implemented in the context of the *developer* role. The result of service composition is the description of services to be executed, and in this context, the *inspector* role need to be activated. Meanwhile, service *innovator* is in collaboration with service *promoter*, and according to the availability of services in service registry, they either promote existing services to be used or search for new available services that could be found from other providers through the internet. The *inspector* role in collaboration with the organizer role is the necessary components for running and monitoring the final choreography. The teamwork role collaboration during each phase is presented in Fig. 3.

## A Case Study—the School Bus

We take as a use case the school bus transportation in the smart city, which is provided to students who live within the boundaries of the city area. The route that a school bus follows for the students' transfer is calculated according to the distribution of student's residence. The information to design the paths derived from the e-governance system and users in the beginning of the school year calculate them. In fact, this is a solved problem and this kind of capability for traffic systems is included in city's information systems. Nevertheless, this only covers the extraction of a straightforward plan, which may give the shorter route for picking the students and transferring them to school.

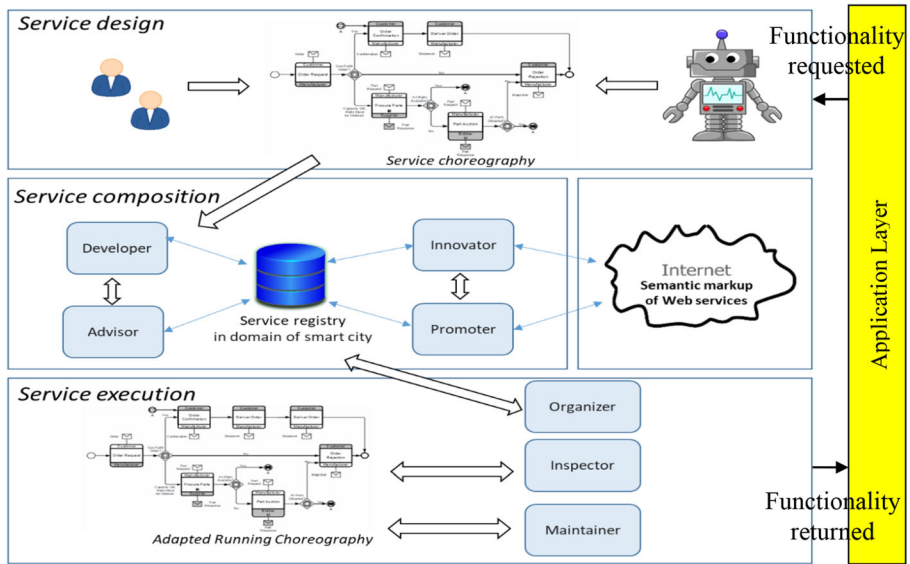


Fig. 3 Smart and sustainable city service nexus – teamworking layer

However, suppose that from last year’s evaluation questionnaires, parents state that specific areas school busses usually delay to pick up students, whenever the weather is rainy. This is a remarkable fact, but it could not help officers improve the service, unless the system could get the extra information related to weather forecast in time for changing the school bus path, or at least inform parents in time about the delay. The municipal information system is very unlikely to store weather results, but there are plenty of weather forecast services in the web that do. On its own, the school bus traffic system used by the municipal simply calculates the daily commute to school that each student should follow. However, in a *smart city role domain*, enclosing services of the above system with services that provide observations of traffic according to the data are produced from cameras provided (IoT services), and combining them with weather forecast data could predict the future in a way that expert participants (teamwork enabled) can act quickly to prevent possible delays or other dangers for bus routes.

Based on the above requirements, we create part of the role model that is relevant to our example and present it in Fig. 4. The involved participants are students, which play the *student* role, but in reality, they involve citizens that use the local transport. School busses and city busses play the *transferer* role, while information services like weather forecast and Meteo play the *forecast* role, which extend the advisor role, while car technicians play the *maintenance* role. The arrow in the role model depicts a collaboration path between two roles and indicates who the initiator is, while the numbering of the arrows corresponds to the lines in Table 1, providing extra details about the collaboration. The colored roles are teamwork roles that contribute to the domain functionality. The example describes a simple but not a trivial transfer case. Through this use case, we present how SCN that derives from teamwork role modeling can add value in a smart city domain. Our intention is to formalize the teamwork role model with a system that enables reasoning over the situations the system evolves.

A goal of the composite service we are looking for is to get all students at school. Before leaving the garage, the bus should check the weather service in case it finds something unusual. This is depicted in the role model with the arrow labeled (1) and is described in Table 1 at step 1. In order for the role *transferer* to communicate with role *forecast*, i.e., the behavior to be executed, the plan should be calculated. The result is for the *transferer* to get knowledge about the weather condition. Supposing that everything is okay with the weather, the bus collects the students according to the designed path as this was defined before it starts the itinerary. Meanwhile, on its way to the school, the bus displays an urgent maintenance alarm. Then, how shall the composite service that performs the student transfer will achieve its goal? In Table 1, we can see the services enrolled in the domain, the role each service place, the behavior it executes, and the precondition and effects of each behavior.

In this situation, the teamwork roles *developer*, *inspector-monitor*, *advisor*, and *maintainer* should intervene and contribute with solutions in order to complete successfully the student transfer. The *inspector* will catch the error and ask the *developer* for alternative solutions. The *developer* role would help in changing the initial plan. The *advisor* would check for other school busses in the surrounding area; check their capacity; and inform if there is one to change its route, providing that it can get there in a reasonable amount of time to come to pick up students, so they could change bus and get to their destination. The *maintainer* is informed about the error, and depending on its type, it gives a prediction about how long the bus will be out of order. The decision that the composite service should make is critical for the success of its execution, i.e., the students transfer.

In Table 1, we also see the different services that participate in the smart bus composite service, the roles they play, the behaviors they execute, and the goal they achieve after the execution. Initially, an information service of transfer role communicates with a forecast role to get the weather condition. Depending on the reply, the

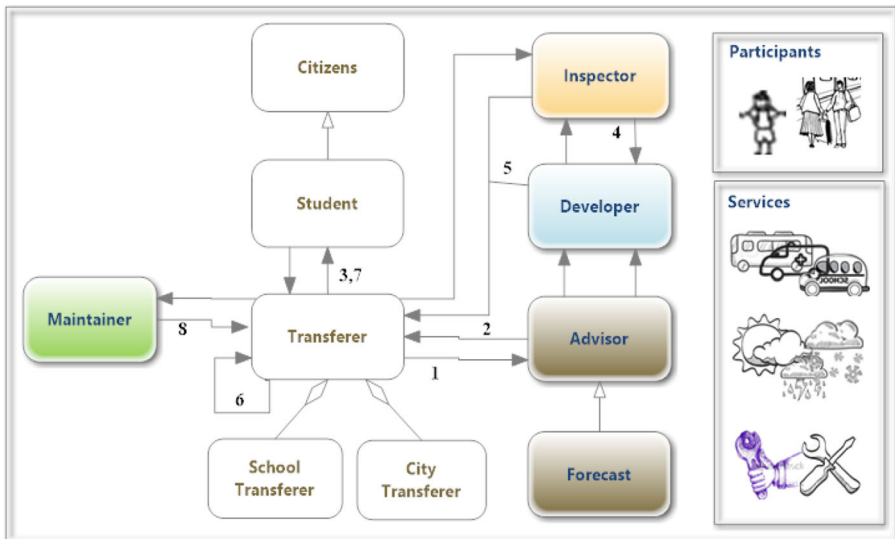
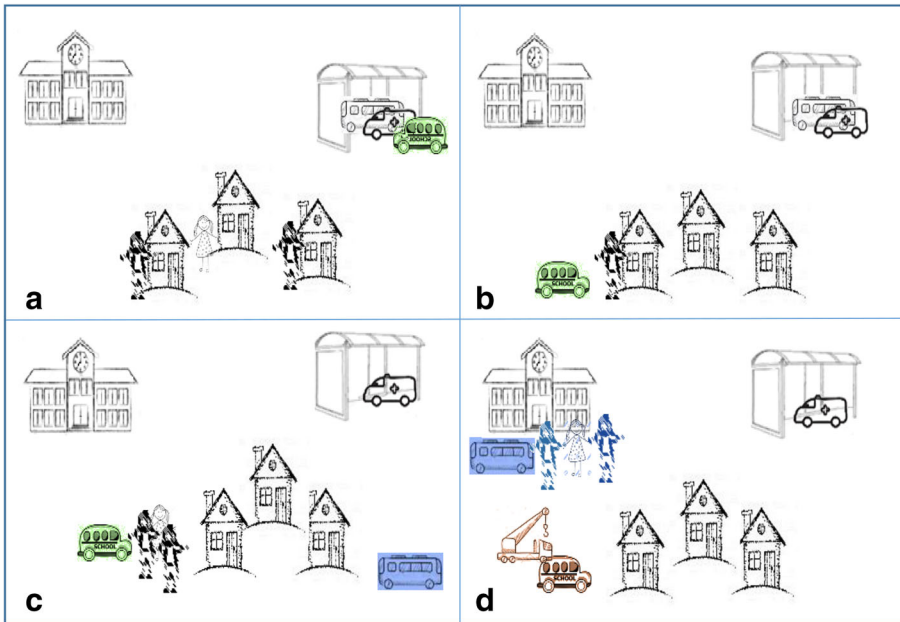


Fig. 4 Part of the role model of the smart city domain along with the participants



**Table 1** Services, roles, and behaviors of the smart bus example composite service

Step	When (preconditions)	A service	Acts as role	Through the action	In collaboration with role	That is played from participant	To achieve the goal (effects)
1	Path to follow is calculated	Information	<i>Transferer</i>	Get weather prediction	<i>Forecast</i>	<i>Weather for Meteo, HMNS</i>	Known weather condition
2	Extreme weather conditions	Plan designer	<i>Advisor</i>	Check alternative plans	<i>Transferer</i>	Smart bus	Get adapted route
3	(Adapted) path calculated	Smart bus	<i>Transferer</i>	Pick up student	<i>Student</i>	A student	A student is on the bus
4	Smart bus breaks down	Smart bus	<i>Inspector</i>	Find alternative	<i>Developer</i>	Path designer	Find other busses nearby
5	Smart bus breaks down, other bus came	Smart bus	<i>Developer</i>	Delegate to other bus	<i>Transferer</i>	Smart bus 1	Bus 1 arrives and students are on bus 1
6	All student are boarded	Smart bus	<i>Transferer</i>	Perform the route	<i>Transferer</i>	Smart bus	Arrive at school
7	Arrive at school	Smart bus	<i>Transferer</i>	Drop off a student	<i>Student</i>	A student	Student gets off at school
8	Smart bus break down	Technician	<i>Maintainer</i>	Maintenance	<i>Transferer</i>	Smart bus	Damage diagnosis, fix damage



**Fig. 5** The model takes a plan and recruits it with services that exhibit teamwork behavior to help with its goal achievement. The presented model in four district states. (a) The initial situation, all students are at home. (b) A school bus on its route to the school. (c) After the bus brakes down, another bus will take students to school. (d) The goal situation achieved, all students are at school, the bus is going to get maintained

*transferer* decides if it follows the schedule or it requests a new quote to the *advisor* role to change its itinerary in case extreme weather conditions exist in a place included in the route. Otherwise, if the reply includes nothing abnormal, then the transfer role continues executing the designed plan. In case that, during the execution of the composite service and after the pickup of some students the bus broke down, then the transfer role, which is played by the smart bus, calls the maintenance center. If the plan is executed successfully, the transferer sends a confirmation about it and the role interaction terminates. By using the teamwork roles of the model and leveraging the teamwork abilities of services, we enable a composite service execution that in other circumstances will fail to execute.

In Fig. 5, we present the initial, two intermediate situations that teamwork roles intervene, change of the plan in order for a solution to be found and achieve the goal situation of the case study.

## Conclusion and Future Work

Since the complexity of current smart city systems is increasing, there is a clear need to infuse autonomy, decision-making, collaboration, and teamworking behaviors in smart city services. The reasons for infusing such behaviors to existing systems are many: lowering the cost of offered services, improving resource usage, improving maintenance, increasing citizen satisfaction, increasing city resilience, etc. The vision is that a smart city should be perceived as one system and not a collection of subsystems.

Currently, existing service frameworks do not have knowledge about what other services are operating within the domain, not even services included in the same composition plan.

Moving toward this direction in this paper, we make two contributions

- we present a role model which is a key tool for modeling collaborative teamworking service behavior and
- we present a set of teamworking roles that can be used for creating a teamworking layer.

Subsequently, we present an analytical example on how this role model and some specific teamworking behaviors can be applied within a smart city in a case of student transportation. More analytically, we habilitate the operational roles of the domain with indicative teamwork behavior that services need as participants of an optimal team in order to underlie the required teamwork abilities and go beyond other approaches by providing teamwork grounded service abstractions.

One of the future directions is that the proposed model could be extended out of the domain of smart cities. This quite easily can be transferred and applied to other domains, such as smart manufacturing, or to any complex IoT environment.

Also, to apply the model in real scale, all smart cities services need to be redesigned and partially redeveloped. In order to better study this problem, in our future work, we plan to simulate several smart city services by using an appropriate simulator (Tsoutsas et al. 2018) to prove the usability and the benefits of our approach. Further, the proposed model could be embedded to services that are running to smart devices having significant computer resources, or when the resources needed for completing a task is not known upfront (e.g., algorithm completion time), making the implementation of some roles challenging.

## References

- Allweyer, T. (2016). *BPMN 2.0: introduction to the standard for business process modeling*. Pasig City: BoD–Books on Demand.
- Ananthakrishnan, R., Bryan, J., Chard, K., Foster, I., Howe, T., Lidman, M., & Tuecke, S. (2013). Globus nexus: an identity, profile, and group management platform for science gateways and other collaborative science applications. In *Cluster computing (CLUSTER), 2013 IEEE International Conference on* (pp. 1–3). New Jersey: IEEE.
- Anthopoulos, L. G. (2017). *Understanding smart cities: a tool for smart government or an industrial trick?* (Vol. 22). Cham: Springer International Publishing.
- Anthopoulos, L., & Fitsilis, P. (2014). Exploring architectural and organizational features in smart cities. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on* (pp. 190–195). New Jersey: IEEE.
- Bano, M., & Ikram, N. (2010, August). Issues and challenges of requirement engineering in service oriented software development. In *In 2010 Fifth International Conference on Software Engineering Advances* (pp. 64–69). New Jersey: IEEE.
- Barayannis, G. (2014). *A novel specification and composition language for services* (Doctoral dissertation, Ph. D. dissertation, University of Crete, Greece, <http://elocus.lib.uoc.gr/dlib/f/6/e/metadata-dlib-1415268652-781614-13324.tkl>).
- Belbin, R. M. (1993). Team roles at work: a strategy for human resource management. zitiert in: Teamarbeit und Teamentwicklung.

- Berlin, J. M., Carlström, E. D., & Sandberg, H. S. (2012). Models of teamwork: ideal or not? A critical study of theoretical team models. *Team Performance Management: An International Journal*, 18(5/6), 328–340.
- Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., & Mecella, M. (2005). Automatic composition of transition-based semantic web services with messaging. In *Proceedings of the 31st international conference on Very large data bases* (pp. 613–624). Los Angeles: VLDB Endowment.
- ITU-T. (2015). Setting the framework for an ICT architecture of a smart sustainable city, Focus Group Technical Specifications.
- Fitsilis, P., Tsoutsas, P., Anthopoulos, L., & Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In *In proceedings of the eighth international conference on advanced collaborative networks, Italy* (p. 2018).
- Gaur, A., Scotney, B., Parr, G., & McClean, S. (2015). Smart city architecture and its applications based on IoT. *Procedia computer science*, 52, 1089–1094.
- Hall, R. E. (2000). The vision of a smart city. In proceedings of the 2nd international life extension technology workshop 2000. Paris, France.
- Hatzi, O., Nikolaidou, M., Vrakas, D., Bassiliades, N., Anagnostopoulos, D., & Vlahavas, I. (2015). Semantically aware web service composition through AI planning. *International Journal on Artificial Intelligence Tools*, 24(01).
- Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: a systematic literature review. *Expert Systems with Applications*, 41(8), 3809–3824.
- Khadka, R., & Sapkota, B. (2010). An evaluation of dynamic web service composition approaches. In *ACT4SOC* (pp. 67-79).
- Kondepudi, S. N. et al. (2014). Smart sustainable cities analysis of definitions. **The ITU-T Focus Group for Smart Sustainable Cities**.
- Kozlowski, S. W., Grand, J. A., Baard, S. K., & Pearce, M. (2015). Teams, teamwork, and team effectiveness: Implications for human systems integration. In D. Boehm-Davis, F. T. Durso, & J. D. Lee (Eds.), *APA handbook of human systems integration* (pp. 555–571). Washington, DC: American Psychological Association.
- Lange, R., Cipriani, N., Geiger, L., Grossmann, M., Weinschrott, H., Brodt, A., & Rothermel, K. (2009). Making the world wide space happen: new challenges for the nexus context platform. In *2009 IEEE International Conference on Pervasive Computing and Communications* (pp. 1–4). New Jersey: IEEE.
- Maddox T. (2018). Smart city technology market set to reach \$775 billion by 2021, <https://www.techrepublic.com/article/smart-city-technology-market-set-to-reach-775-billion-by-2021/> .
- Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering: The Journal of the International Council on Systems Engineering*, 1(4), 267–284.
- Margerison, C. J., & McCann, D. (1990). *Team management: practical new approaches*. Mercury Books.
- Mathieu, J. E., Tannenbaum, S. I., Kukenberger, M. R., Donsbach, J. S., & Alliger, G. M. (2015). Team role experience and orientation: a measure and tests of construct validity. *Group & Organization Management*, 40(1), 6–34.
- Mayk, I., & Madni, A. M. (2006). The role of ontology in system-of-systems acquisition. In *Proceedings of the 2006 Command and control research and technology symposium*. CA: San Diego.
- Nacer, H., & Aissani, D. (2014). Semantic web services: Standards, applications, challenges and solutions. *Journal of Network and Computer Applications*, 44, 134–151.
- Nicklas, D., & Mitschang, B. (2004). On building location aware applications using an open platform based on the NEXUS augmented world model. *Software and Systems Modeling*, 3(4), 303–313.
- Parker, G. M. (2008). *Team players and teamwork: new strategies for the competitive enterprise* (2nd ed.). San Francisco, CA: Jossey-Bass.
- Patel, S., & Shah, T. (2016). A survey on issues and challenges of web service development, composition, discovery. *Journal of Science and Technology (ISSN: 0975-5446), VNSGU*, 5(1).
- Rao, J., Kungas, P., & Matskin, M. (2004). Logic-based web services composition: from service description to process model. In *Proceedings. IEEE International Conference on Web Services, 2004.* (pp. 446-453). IEEE.
- Rousseau, V., Aubé, C., & Savoie, A. (2006). Teamwork behaviors: a review and an integration of frameworks. *Small Group Research*, 37(5), 540–570.
- Sage, A. P., & Cuppan, C. D. (2001). On the systems engineering and management of systems of systems and federations of systems. *Information knowledge systems management*, 2(4), 325–345.
- Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: *discoveries and developments*. *Human Factors*, 50(3), 540–547.

- Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojevic, D. S. (2017). Software platforms for smart cities: concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (CSUR)*, 50(6), 78.
- Schwartz, T., Feld, M., Bürckert, C., Dimitrov, S., Folz, J., Hutter, D., & Mronga, D. (2016). Hybrid teams of humans, robots, and virtual agents in a production setting. In *Intelligent Environments (IE), 2016 12th International Conference on* (pp. 234–237). New Jersey: IEEE.
- Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: a decade's overview. *Information Sciences*, 280, 218–238.
- Tosi, D., & Morasca, S. (2015). Supporting the semi-automatic semantic annotation of web services: a systematic literature review. *Information and Software Technology*, 61, 16–32.
- Tsoutsas, P., Fitsilis, P., & Ragos, O. (2019). Teamwork behavior: a review to interconnect industry 4.0 entities. In *Technological Developments in industry 4.0 for Business Applications* (pp. 1–25). IGI Global.
- Tsoutsas, P., Fitsilis, P., & Ragos, O. (2018). Services simulation in industry 4.0: a comparison of simulators. In *3rd International Scientific Conference on Industry 4.0*. Varna.
- Tsoutsas, P., Fitsilis, P., & Ragos, O. (2017). Role modeling of IoT services in industry domains. In *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences* (pp. 290–295). ACM.
- Vlacheas, P., Giaffreda, R., Stavroulaki, V., Kelaidonis, D., Foteinos, V., Poullos, G., & Moessner, K. (2013). *Enabling smart cities through a cognitive management framework for the internet of things*. New Jersey: IEEE.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Paraskevi Tsoutsas<sup>1</sup> · Panos Fitsilis<sup>2</sup> · Leonidas Anthopoulos<sup>3</sup> · Omiros Ragos<sup>1</sup>

Panos Fitsilis  
pfitsilis@gmail.com

Leonidas Anthopoulos  
lanthopo@gmail.com

Omiros Ragos  
ragos@math.upatras.gr

<sup>1</sup> Department of Mathematics, University of Patras, Patra, Greece

<sup>2</sup> Department of Business Administration, University of Thessaly, Larissa, Greece

<sup>3</sup> University of Thessaly, Larissa, Greece