

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A Reference Architecture for Digital Ecosystems

Alexandru Averian

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.77395>

Abstract

Digital ecosystems are a new type of application based on a “universal digital environment” populated by digital entities that form communities that evolve and interact with information exchange and who trade digital objects that are produced through the system. Entities that participate and form the ecosystem can be applications running not only on simple devices: wearable, sensors, actuators, but also on complex services executed on smartphones, tablets, personal computers, company servers, etc. A reference architecture for digital ecosystems is a step toward standardization, as it defines a set of guidelines in designing and implementing a digital ecosystem. Often such architectures are very abstract, difficult to understand and implement. In this chapter, we introduce a vendor- and technology-neutral reference architecture for digital ecosystems and apply this architecture to an actual use case.

Keywords: digital ecosystems, reference architecture, adaptive, context-aware

1. Introduction

A series of architectures have been taken into consideration in the construction of digital ecosystems. Briscoe presents in [1] the first neural calculus applications that bring together, in a new approach, elements of theory with service-oriented architectures, multiagent systems and distributed computing components, the proposed digital ecosystems to deliver business support. In [2], the principles and semantics used in digital ecosystems are formulated, as shown in [3], Chang et al. continue to research digital ecosystems by broadening their scope in areas such as transport, education and health. Article [4] presents the implementation steps toward an agent-oriented architecture of the digital healthcare ecosystem. All examples assume the existence of an environment of communication and intelligent agents or context-conscious applications that respond to changes that occur in the environment. In

this chapter, we introduce a new, high-level, reference architecture for the development of digital ecosystems. We propose a set of steps toward a more specific reference architecture for digital ecosystems, and introduce Reference Architecture for Digital Ecosystems (RADE), a six-layer architecture comprising: environment, context management, interaction, adaptation to goals, species management, user integration, and finally, we apply proposed architecture to an actual use case. This chapter is structured as follows. Section 2 presents the related work and current results in the field of ecosystem representation. Section 3 presents a new reference architecture for digital ecosystems. Section 4 describes a possible implementation of proposed reference model. Last section presents the conclusions and hints for future work.

2. Related work

As introduced in [5], the ecosystem oriented architecture (EOA) model, named architectural style, defines a digital business ecosystem as being an open ensemble of distributed services. This differs from a service-based architecture (SOA) because it needs to address new issues such as: decentralization, managing a distributed knowledge base, self-organization, and self-rebuilding. EOA is not a larger SOA or other SOA [6], it addresses a new type of evolutionary, dynamic, knowledge-sharing, self-organizing, self-controlling, self-reliant architectural model as it occurs in natural ecosystems [7]. The EOA architecture applied to a digital business ecosystem presupposes the following components:

- Services described from the perspective of the problem (business), the computational description (interface) is not sufficient, and a business specification is needed.
- Service registry—intelligent discovery mechanisms based on the business specification are required because the services will not be known in the construction of the system.
- Model repository—it is separated from the service register and must allow for a model-driven development, so that models can be categorized by users (folksonomy) and improved.

Minimal support services that facilitate communication between services and ecosystem development help participants integrate and publish new services.

An SOA-based application is owned and managed by a large organization, operates within its network, and interconnection (B2B) with other organizations is usually required. The user community must be the owner of the ecosystem, a P2P network is more appropriate in this case being more “democratic.” On the other hand, in a digital ecosystem, there must be no hierarchical topology based on interest, it must not be a single point of administration, and the system must be self-configuring and adaptive.

3. Reference architecture for digital ecosystems

In this section, the RADE model is presented—a reference architecture for digital ecosystems. After a brief introduction, the section 3.1 presents the anatomy of the digital species

participating in ecosystem. The following sections describe each layer in the species structure. The section concludes with a series of considerations on security, identity, and trust.

In order to present the components of a digital ecosystem, we start from the structure of a natural ecosystem. A natural ecosystem is made up of biotic (living entities) and abiotic components, whose interaction creates an ecosystem that perpetuates itself. More specifically, it consists of one or more communities of organisms of different species that manifest themselves in their habitat. Of each species, there may be several populations that activate in their microhabitat [8]. A community is formed by groups of populations of various species that operate in the same habitat. A habitat is a distinct part of the environment. An individual organism or a population can migrate from one habitat to another in search of resources, thus being able to compete with other organisms. A microhabitat is a subdivision of a habitat with its own specific properties. A population that operates in a microhabitat will tend to occupy a niche. This is a functional relationship between a population and the environment it occupies. Niche emerges as a strong adaptation of a population to the occupying microhabitat [9]. One can find more details about the habitats in articles [1, 10], which presents the first applications in the neural calculus area that bring together, in a new approach, elements of theory with service-oriented architectures, multiagent systems and element-distributed computing, and the digital business ecosystems.

Ecosystems are described as complex adaptive systems (CAS) consisting of diverse components that interact locally and are subject to the process of natural evolution and selection. Digital ecosystems are composed of populations of agents evolving (through natural selection) in the distributed environment. These fall within the definition of complex adaptive systems, having a nonlinear evolution and always aiming at a dynamic balance. We present the following components of a digital ecosystem:

- the communication medium is a P2P communications network or a real-time data distribution system (such as DDS—Data Distribution Service), TCP/IP- or UDP-based networks;
- habitat is a key element in the functioning of an ecosystem, it is a network node running the ecosystem services in which support services and optimization services (EVE) are run [5, 6]. Each user has a habitat that connects and launches queries in the ecosystem, users can define and activate new services. Habitats communicate with each other and can form clusters of habitats based on data exchanges between them.
- the entities of a digital agent ecosystem, context-conscious applications, or local or remote services, sensors or intelligent objects (IOTs) that can be connected to the habitat to provide context information; these are the species that populate a digital ecosystem.
- the context defines the environment in which a particular entity is in the ecosystem at a given time, it contains primary data relevant to the application, a high-level context obtained by aggregating primary data, and context contexts.

Reference architecture for digital ecosystems represents a step forward for standardization, because this defines a set of guidelines in designing and implementing a digital ecosystem. Usually, these types of architectures are very abstract, hard to understand and implement.

RADE architecture integrates devices and species, and it takes a device or a thing and upgrades it with context-awareness, adaptability, and autonomicity and produces a digital

species populating a digital ecosystem. The anatomy of a species is formed from six main layers and a security component, which can also be found on every layer, as can be seen in the next section (**Figures 1–6**).

3.1. Anatomy of species in ecosystems

From an ecosystem perspective, every actor participating in a supply chain ecosystem is represented by its digital counterpart in digital ecosystem. An entity is part of a species if it is designed and programmed to behave in a certain specific way, to use a certain type of resources and to act according to a specific context. In this section, we introduce the reference model of species participating in a digital ecosystem. The introduced model proposes a set of guidelines in designing and implementing of digital counterpart of players taking part in a supply chain ecosystem. Anatomy of species is comprised of six main layers and a security component, which can also be found on every layer, as can be seen in the figure below.

A reference architecture operates with components with a high degree of similarity, so they can be assembled correctly and safely, resulting in complex yet scalable solutions, providing flexibility for various application scenarios. In this respect, the following guiding principles can be found in different areas of architecture:

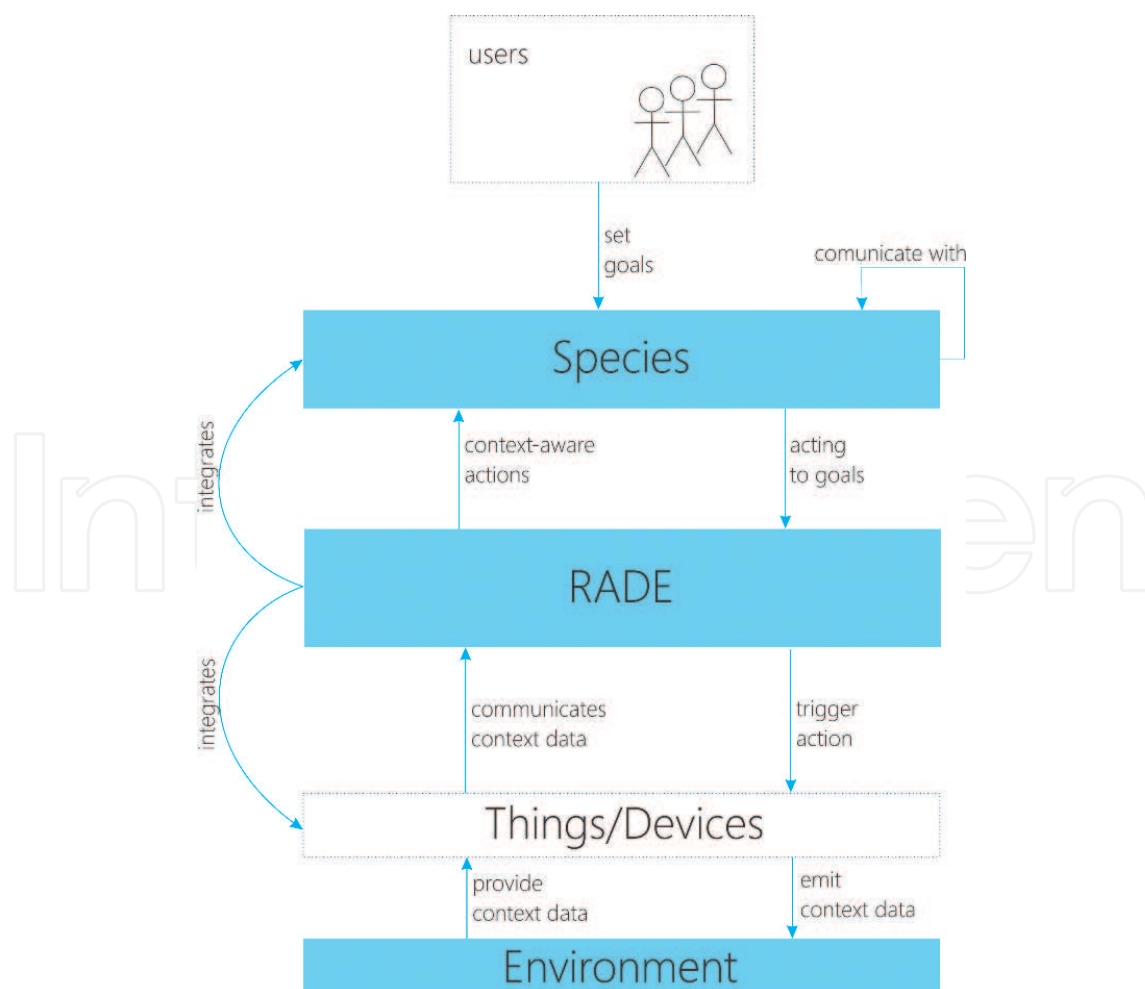


Figure 1. RADE overall architecture.

- Heterogeneity—ecosystems are open systems, a reference architecture must cover a wide variety of logical, physical and virtual entities, processing patterns and standards, and it must be able to use a wide variety of hardware and software platforms.
- Flexibility—this assumes that the system is easily changed, permits easy assembly of heterogeneous components, and the easy assembly of a varied set of components and services.
- Weak coupling—this involves the use of poorly coupled processing modules and a communication medium that allows the digital entities involved to be decoupled in time and space.
- Scalability—this requires the system to admit a large number of connected entities (theoretically unlimited); the system being open, it must admit the addition of new participants in a flexible way.
- Security—certain areas of application of digital ecosystems (e-health) will imply a strong connection between the physical world and the digital world, for the realization of secure systems, the model should include multilevel security measures including identification and authorization of digital entities and users, data protection, and authentication.

3.2. Environment

The environment level is the mode of communication between species, and it extracts the information from other entities and helps to communicate data/orders in the environment. The digital environment can be a peer-to-peer (P2P) system that has a number of advantages over a centralized (client/server) model that is not resilient, error tolerant, scalable, and vulnerable to attacks. These advantages result from the network definition mode, P2P is defined as a network in which the nodes are equivalent to each other in the sense that all nodes (in principle) can execute the same set of functions needed for network to work. The most important features of a P2P network are as follows:

- resource sharing through direct transfer with no centralized servers, however, sometimes centralized servers that can be used for setting up the network, node management, etc.;
- no centralized nodes, no central fall points, no central attack points;
- nodes actively participate in operations such as handling information, finding resources, and storing and managing data;
- the network has the ability to adapt to changes in connectivity, to changes in typology, and the ability to reconfigure itself after finding an error;
- the typology of a P2P network is tolerant to defects, having the ability of auto-organization in order to keep functioning;
- the network can be structured or not, and physical proximity between nodes is not important;
- IP, the links between nodes are TCP connections but can be represented as pointers to the IP address.

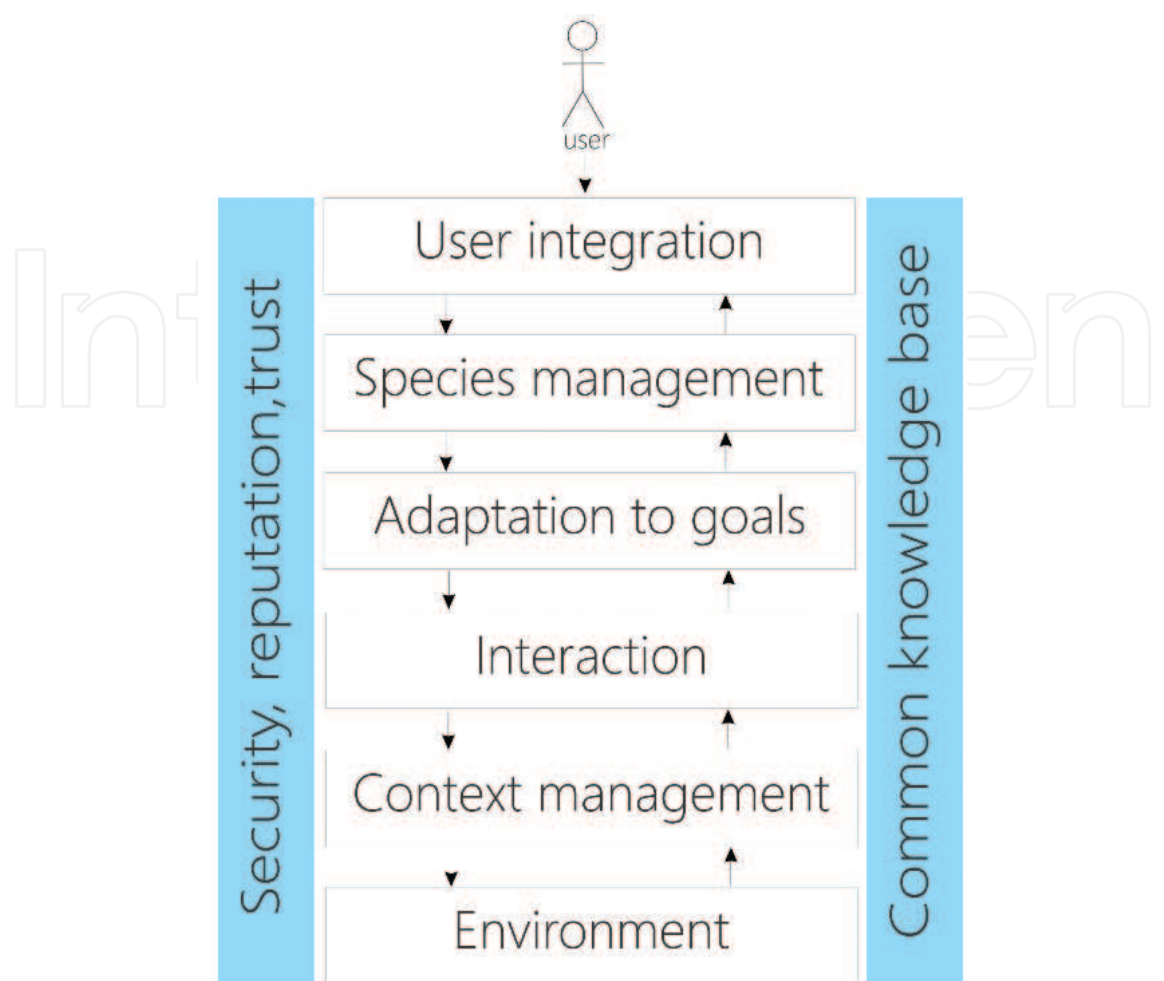


Figure 2. Anatomy of species in digital ecosystems.

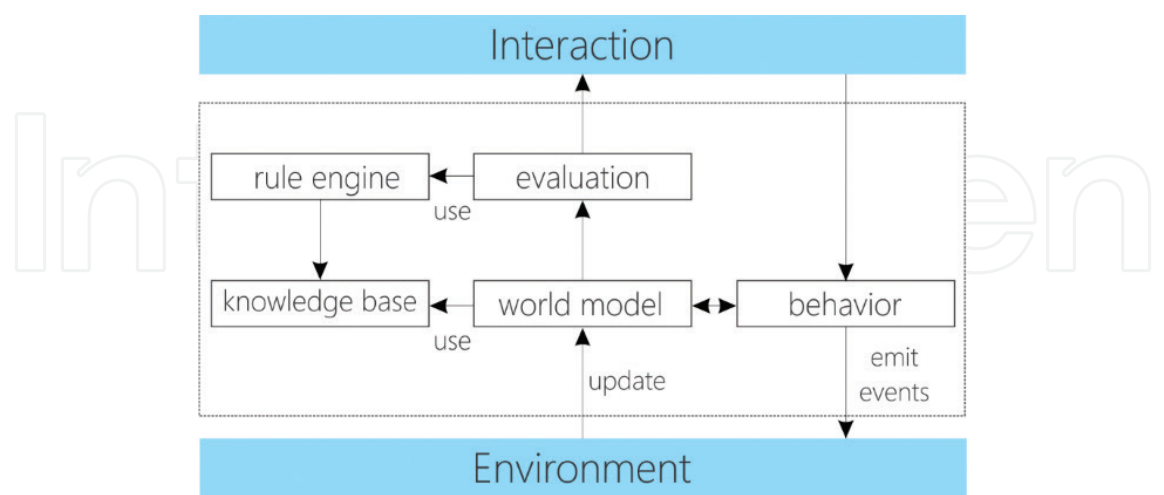


Figure 3. Context management.

Depending on the type of application, the level of access to the environment can be implemented through a messaging-oriented machine-to-machine type such as Data Distribution Service (DDS), Extensible Messaging and Presence Protocol (XMPP), Advanced Message

Queuing Protocol (AMQP), Message Queue Telemetry Transport (MQTT), or Constrained Application Protocol (CoAP). These systems are widely used to implement IOT applications. Message-Oriented Middleware (MOM) is a middleware product family that facilitates messaging across distributed systems. A MOM system uses one of the following communication paradigms: message passing, indirect queuing, *publish/subscribe* communication (data are published through a topic, and customers receive all messages posted by the topic they are subscribed to). Of the three communication models mentioned, *publish/subscribe* is best suited for building the level of access to the environment within the RADE architecture because it provides asynchronous, scalable multi-to-many communication. In this scheme, the messaging emitters communicate with the subscribers, without prior knowledge, through a distributed P2P infrastructure. The system allows a decoupling in terms of time, space, and synchronization. Disconnection over time allows the broadcaster and receiver to communicate without having to be online at the same time and to cooperate directly. Decoupling in space refers to the fact that the transmitter and receiver are unaware of each other, and their identity and location are not relevant. Synchronization decoupling refers to the fact that the receivers and transmitters do not have to synchronize, and the communication is accomplished by asynchronous notifications implemented with a callback function system. For sending messages to the environment or for exchanging messages between entities, an RPC communication scheme will be used.

3.3. Context and niche

Context data can only be used in the presence of a well-defined goal. Thus, some contextual information along with a defined purpose can generate actions to be taken. In other words, if we have a formal context consisting of a series of objects and a set of attributes, the purpose of the application is defined by a set of opportunities (affordances) that invite to action. Opportunities create a relationship between subject and object, the subject being the application/entity and the object can be any context information. These define the rules by which when a context object changes its state, the subject can act. The rules define the role a species has in its environment. The totality of interactions between a species and the environment forms a niche. The goal can be defined not only at the level of an application and at the level of a user queries, but also at

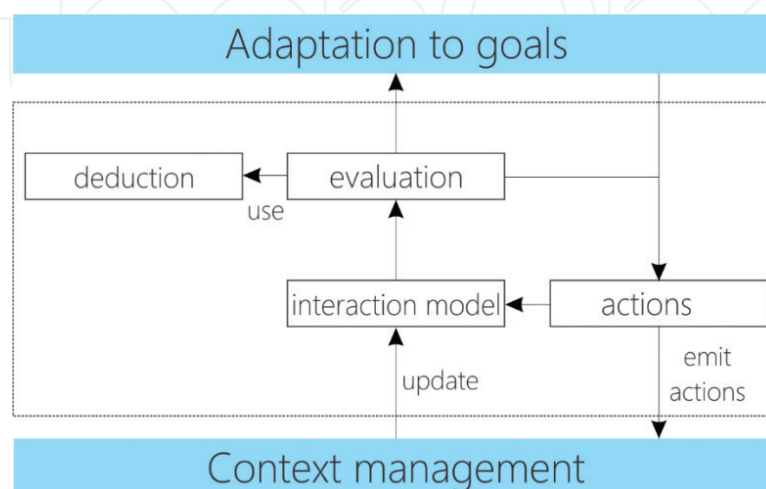


Figure 4. Interaction level.

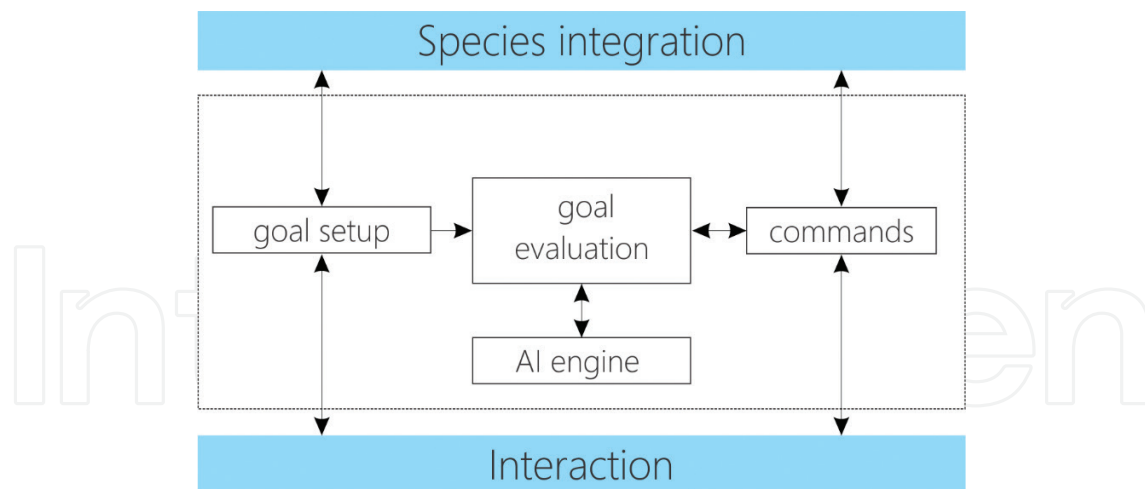


Figure 5. Adaptation to goals (optimization level).

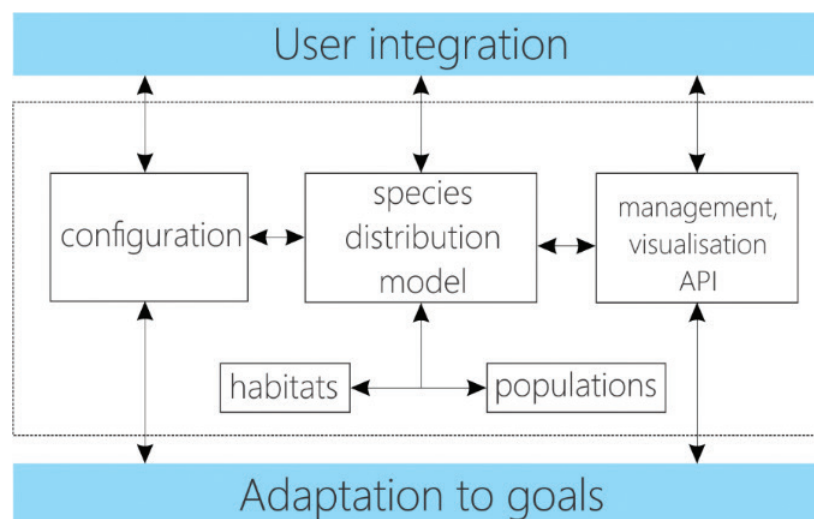


Figure 6. Species integration.

the level of applications that work in the background or at the level of intelligent objects or autonomous robots that work for the user but without direct intervention. Niche expands the concept of context, as defined by Dey in [11], to digital ecosystems; it encompasses any relevant information that is useful to characterize not only the situation of an entity in the ecosystem but also the mode of action of the entity over environment. In article [12], there are introduced as the stages of implementation of a multiagent health ecosystem, and finally, article [13] defines the context of e-health as avatars and virtual organisms as “a collection of information from e-health systems used by people and applications that characterize the situation of another entity (usually a person but may also be an application) in its environment used to interpret the state or behavior of the entity concerned.” A wide range of context approaches have been presented in the literature, especially in the field of ubiquitous computing. These are key-value, object-based, logic-based models based on ontologies, using graphical representation or markup. A classification of context patterns can be found in [14]. All examples involve the

existence of a communication medium and intelligent agents or context conscious applications that respond to changes that occur in the environment. The context programming model for digital ecosystems introduced by the author in [15] and presented extensively in [16] is an objectual model where the context is represented as a set of communicating streams; the streams can be connected to various filtering/processing operations, finally a series of reactive variables in the application are updated. Generally, the context management level contains logic for extracting and processing context data. At this level, the following operations are executed:

- extraction and primary processing of data,
- aggregation of primary data to obtain more complex contexts,
- assessing the context situation and issuing signals to the upper layers,
- reception at the upper level of orders, actions, behavior adaptation,
- issuing events, actions, and context data to the environment describing the state of the entity.

The context management level consists of five components as can be seen in the following figure.

The *knowledge base* component can be considered as a database where all the data describing the entity context are stored; they can include information about other objects or entities relevant to the application and their context data. In some cases, it will be implemented through a database management system (SGBD), and in other cases, it could just be a simple collection of information.

The *world model* component is a data model that reflects the real world situation in which the entity in question is running—the context situation. The component also retains the contexts' context of other entities and integrates the changes that occur in these situations. The data are stored in the knowledge base. The rule engine component defines and manages context rules and mechanisms. Generally, this component consists of a set of evaluation rules that also contain actions that can be taken to change the context situation. Concrete implementation can be varied, depending on the specificity of the application; in some cases, it can be a simple set of rules that evaluates data in the form of key-value pairs, and in other cases, it can be given by sophisticated systems for processing its ontology data system mining. The evaluation component evaluates the context situation reflected in the world model component and the changes that occur in it. Evaluation uses the assessment engine and produces status changes and actions that it communicates to the level of interaction. The behavioral component receives commands from the interaction level and emits events in the environment that signal the behavior of the entity. This component may add changes to the context model (world model).

3.4. Interaction

The interaction level receives information about changes that happen in the context, events, or messages received from other entities in the environment, orders (from trusted/authorized entities) that can lead to reconfiguration of the purpose. In the case of a reconfiguration (e.g.,

“abort mission”), or if there is a situation in which urgent measures must be taken (the occurrence of threat in the species), the level of interaction emits in the environment through the context component of the context a message by which it signals the situation. It also signals changes to higher levels of adaptation to goals and species integration and can ask (after receiving confirmations) to change or reconfigure the objective/purpose of the current entity. This level maintains communication and ongoing interaction with other participants in the environment, issues actions and events that describe the entity’s behavior and intentions in the environment. The interaction level consists of the following components as shown in the below

- model interaction is a model that defines the application’s action mode, how to relate the entity to the environment, reflecting the entity’s behavior in the real world. The interaction model describes how users understand the application. Defining a pattern of interaction is essential. Once defined and understood, users can understand and track the way an entity operates. This is a fundamental pattern that describes how certain elements relate to each other, may contain sub-modules for various subcomponents, and together they constitute the general pattern.
- evaluation permanently assesses the current situation and events occurring in the environment, applies the rules of the interaction model, generates messages for the higher level, and also issues commands to the action component.
- deduction—a deduction engine that can be used in decision-making on interaction and can deduce new interaction rules from observations on the evolution of the site.
- Actions—receives orders from the higher level or from the assessment component on the same level, applies entity-specific actions and issues lower layer behaviors to be published in the environment.

The interaction model depends on the application, it can also be a model with simple one-way rules, the cause-effect form, or it can be a very complex interrelation pattern. For example, for business digital ecosystems, *ActionWorks* Business Interaction Model can be used to coordinate the interaction between a customer group and a group of providers through a four-step feed-back loop: preparation, negotiation, delivery, and acceptance. For other applications, the Complexity of Interaction Sequences (CIS) model introduced in [17] can be used. This model uses interaction sequences that are defined as action steps that change the status of a system, and any problem that needs to be resolved is seen as a state to be reached as a result of executing a sequence of steps.

3.5. Adaptation to goals

Adaptability is the ability of a system to change its behavior according to new, unexpected situations [18]. The adaptive properties of an organism are closely related to the self-organizing property [19] and the emergence phenomenon. Applications from a digital ecosystem must solve concrete problems but also be computationally efficient. It will seek to establish a balance between the freedom of a system to self-organize and the constraints that apply to obtain useful solutions. Briscoe in [20] proposes a digital ecosystem model that incorporates evolutionary and self-organizing properties specific to natural ecosystems. The model

applies an EOA architecture to a distributed multiagent system, and evolving mechanisms are made on two levels within the evolutionary component (EvE). The first level is formed by a P2P network of agents (evolutionary population) that feeds a second optimization system that operates locally (at the habitat level) and exploits evolutionary algorithms to identify solutions that satisfy relevant local constraints. The local search process of the solutions is accelerated by the exchange of values (migration of individuals) between different habitats in which a calculation with similar constraints is executed. This level has the role of effectively solving complex problems so that the system is getting closer to the purpose for which both an individual system and the whole group are configured.

The goal evaluation component could implement the adaptive reference architecture defined in [21], which presents the structure of a MAPE-K Loop Adaptation Manager, comprising a series of activities to be followed in order to have complete feedback and adaptation. MAPE-K comes from monitoring, analysis, planning, execution, and knowledge. The four steps present in the loop correspond to the four activities that are also found in the medical field: observation, diagnosis, solution, and treatment. The Knowledge Base retains information about the adapted system and its context, and this information is used by all four stages of the feedback. Depending on the degree of adaptation of the component, an increasingly complex knowledge base is needed, along with the advanced deliberative mechanisms leading to the adaptation process. The system continually assesses its own state and context in which it finds and issues decisions (and internal or external commands) that adjust the state of the system toward the goal. At this level, an AI engine or a set of evolutionary algorithms such as genetic algorithms, bee colony optimization [22], and intelligence swarm will be used. In the case of digital ecosystems implemented as multiagent systems, membrane-computing models [23] can be used for specification and implementation.

3.6. Species integration

The concepts of species, individuals, integration and cohesion are widely debated in the literature of biology and ecology [24]. The term integration refers to the active interaction between the components of a system. Cohesion refers to cases where a component of a system behaves like the whole system, relative to a particular process. Thus, the presence and action of a part of a system does not affect the activity of another part of the system, although all parts are uniformly responsible for a certain type of stimulus and behave similarly to the same process. The level of species integration allows the integration and configuration of participating entities within a species. Also, at this level is the general purpose of a species, splitting and managing population populations to respond to queries or to solve a specific problem. A population is a part of a species that operates in a specific context. An entity becomes part of a population and a species if it is programmed to act in a certain way specific to the species, to use a certain type of resources and to act according to a specific context to the population to which it belongs.

3.7. User integration

The user and his applications are part of the ecosystem. The user integration level integrates the users and applications with which it interacts on the last layer, the level can be viewed as a service or as a graphical interface located on the highest level of architecture. The concrete

implementation of this level is dependent on the specificity of each application, actors and usage cases. At this level, setup commands and queries or commands will be launched by the ecosystem. In some cases, this layer takes care of authentication, authorization, accounting for the use of shared resources and payment services.

3.8. Security, identity, and trust

In building a digital ecosystem, security issues must be considered on each level. Depending on the specificity of the application, a series of attacks can be triggered at the level of connected devices, network, operating systems, application level, or user level. A digital ecosystem is an open system, besides the “classic” security issues, there may be problems of reputation and trust. In a distributed system, such as a digital ecosystem, there is a need for trust between users and organizations. Trust is a multidimensional concept that is hard to define and difficult to measure [25].

Article [26] analyzes trust from the technological, economic, behavioral, and organizational perspective. The technological dimension of trust expresses the subjective probability of an organization to believe that a particular infrastructure can facilitate transactions in line with its expectations. The technological dimension includes security services, mechanisms that ensure the confidentiality, authenticity, nonrepudiation and integrity of transactions, as well as mechanisms that ensure identity control and access to resources. A distributed identity management system must exist in the ecosystem so that it is possible to ensure the identity of a service provider as well as consumers to control access to resources. The economic dimension involves establishing relationships of interdependence between organizations (based on a cost-benefit analysis) and the use of IT infrastructure for trading, data transfer, and know-how. In [27], a model for the management and accounting of the use of services in digital ecosystems based on an SOA architecture is presented. The behavioral dimension of trust is derived from the characteristics of interpersonal behavior, which relate to competence, predictability, honesty, and good intentions. The organizational dimension of trust results from the use of good practices, quality standards, audit, risk management strategies, and process management standards.

4. Supply chain ecosystem

There are a variety of supply chain management models in literature as results from a recent review [28]. Markus and Loebbecke [29] use the term ecosystem as a unit of analysis in describing groupings of suppliers and distribution chains, which are understood as loose sets of organizations engaged in the creation and delivery of products and services, the same term is used by Iansit and Levien in [30] describing strategy as an ecology. In [31], the authors present the opportunity to develop a digital ecosystem for transportation and warehousing logistics. This involves building a supply chain [32] that would facilitate the integration and collaboration of small and medium-sized enterprises (SMEs) in particular, would encourage cooperation, would be an opportunity to create synergy, facilitate incubation, increase, and would bring prosperity to the business. The “Virtual Collaborative Consortium” digital ecosystem implemented in Australia is an example, which is a collaborative environment for all those involved

in the product distribution chain. In [33], an agent-based distributed supply chain model is proposed and a number of open issues are formulated. In [32], the delivery chain problem is formulated in terms of task dependency network, a mathematical model is proposed, and equilibrium and convergence issues are studied. In this section, we present an application of the digital ecosystem architecture on a section of the Amazon retailer chain as introduced in [34].

The automation of operations in a warehouse seems to be a difficult operation, but some companies have already made great strides in this direction. The orange robots, as can be seen in the next figure, are simple machines that move horizontally on a 2D grid in all directions, can enter under the shelves in the warehouse, lift them, and carry them to the desired destination. Kiva robots are generally used to transport the shelves of objects to be shipped to the selection and packaging table. After taking over the objects, they carry the shelves back to their place. In addition, they can be used for warehouse shelving operations, for more efficient use of storage space, for sorting and ordering shelves for delivery. Robots with a mobile arm operate on packing and putting packages on the conveyor. The drones’ species connects the packages on the platform and takes them to their destination. The following table summarizes a case of using the architecture for digital ecosystems on a section of the Amazon retailer supply chain. It includes actors, purpose, preconditions, a correct usage scenario, and postconditions (**Table 1**).

Actors	<div>1.Human operator</div> <div>2.Client</div> <div>3.Kiva robot species</div> <div>4.Species of mobile handler robots</div> <div>5.Drone species—Prime Air</div>
Goals	Delivery of products to recipients. Customer orders are quickly honored, delivery is done with the help of the drones in rural areas and peripheral urban areas.
Preconditions	There is a stock of products displayed on a website.
High-level success scenario	<div>1.The operator picks the general role for every species and for the robot population.</div> <div>2.The client makes an order in the system through the website.</div> <div>3.The system checks the stock and sends a movement order of the product to the packing line.</div> <div>4.Kiva Robots will bring the rack with the ordered products to the packing line.</div> <div>5.Manipulating robots pack the products and place the package on the delivery line.</div> <div>6.At the end of the line, another manipulating robot extracts the package from the tape and places it on a platform.</div> <div>7.A Prime-Air drone picks up the package, reads the code extracts the address of the destination and performs the delivery flight.</div> <div>8.The delivery is made, confirmed and the drone comes back to base.</div>
Postconditions	The client confirms the reception of the package online and can use the product.

Table 1. Use case summary.

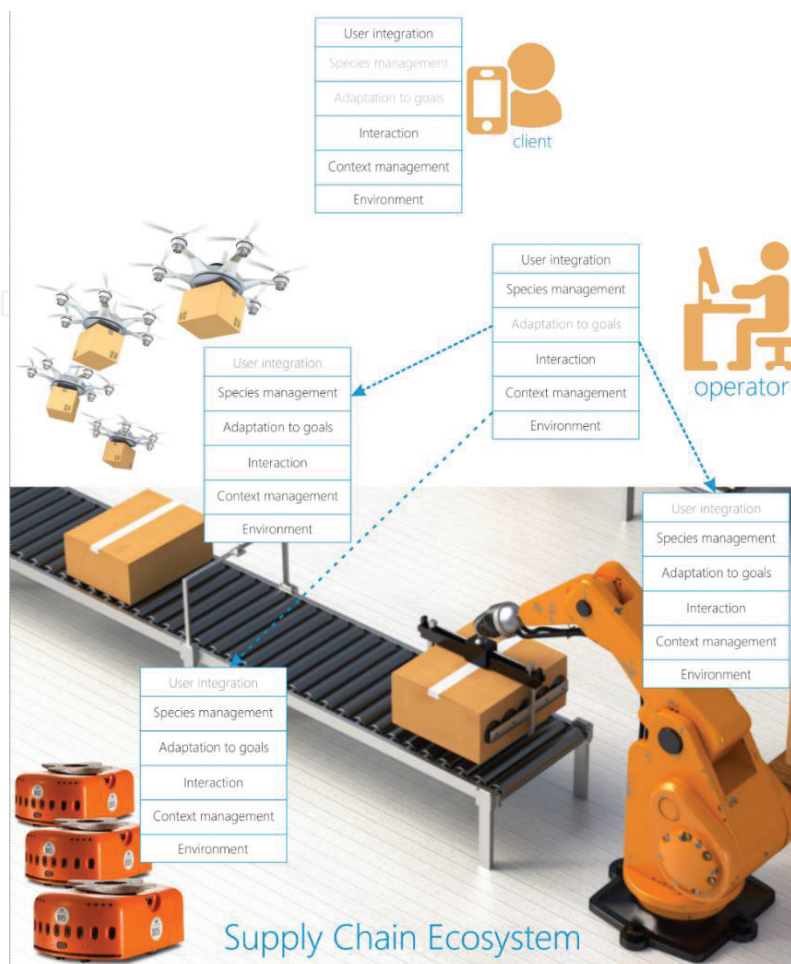


Figure 7. RADE architecture mapping on use case [34].

As can be seen from the **Figure 7**, the Kiva robot species, the arm robots, and the drones species do not contain the functions of the first level of user integration, the applications used by the operator species also implement the level of user integration but lack the level of adaptation to the goal. We can also see that all species integrates species management functions; at this level, each species of robots can be configured, setting the mode of work — the purpose. The operator performs this configuration through its own level of species management. Otherwise, it can be observed that all other layers of the RADE architecture are present within each species. Security features were omitted in this example for simplification. It can be considered that all digital objects are reliable, and access to the system is checked at the network level. In the following figure, we can see how to map the levels of the RADE architecture for each actor.

5. Conclusions

The research area of digital ecosystems is becoming more and more important. Despite a large number of relevant works in this area, the level of knowledge is still insufficient. Digital ecosystems offer many opportunities but also challenges for researchers and developers. In this chapter, we introduce a vendor and technology neutral reference architecture for digital

ecosystems and a possible application of this architecture to an actual use case. The introduced architecture proposes a set of guidelines in designing and implementing a digital ecosystem. The proposed model consists of the following six layers: environment, context management, interaction, adaptation to goals, species management, and user integration. In the final part of the chapter, we presented supply chain ecosystem, an application of the RADE model for a section of an ecosystem supply network, containing four species, namely the species of human operators, the species of drones, the Kiva robot species, and robots with a mobile arm. This work follows the study conducted in [15, 16] and continue the research that was presented in respective paper. Future research will seek to refine the model, will try to integrate with blockchain technology, and adopt some algorithms from AI domain.

Acknowledgements

The work presented in this chapter has been funded by the Sectoral Operational Programme Human Resources Development 2007–2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395.

Author details

Alexandru Averian

Address all correspondence to: aaverian@gmail.com

Politehnica University of Bucharest, Romania

References

- [1] Briscoe G, De Wilde P. Digital Ecosystems: Evolving Service-Orientated Architectures no. 507953. 2006 1st Bio-Inspired Model. Network, Information and Computing Systems. 2006
- [2] Boley H, Chang E, Digital ecosystems: Principles and semantics. In: Proceedings of 2007 Inaugural IEEE International Conference on Digital Ecosystems and Technologies; 2007. pp. 398-403. DOI: 10.1109/DEST.2007.372005.
- [3] Chang E, West M. Digital ecosystems and comparison to existing collaboration environment. WSEAS Transactions on Environment and Development. 2006;2(11):1396-1404
- [4] Vasilăţeanu A, Şerbănaţi LD. Towards an agent-oriented architecture of the digital healthcare ecosystem. UPB Scientific Bulletin, Series C: Electrical Engineering. 2012;74(2):87-102
- [5] Briscoe G, Sadedin S, Wilde P. Digital ecosystems: Ecosystem-oriented architectures. Natural Computing. 2011;10:1143-1194

- [6] Ferronato P. Architecture for Digital Ecosystems, beyond Service Oriented Architecture (IEEE-DEST 2007). In: 2007 Inaug. IEEE-IES Digit. Ecosyst. Technol. Conf., 2007. DOI: 10.1109/DEST.2007.372047
- [7] Levin SA. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems*. 1998;1(5):431-436. DOI: 10.1007/s100219900037
- [8] Begon M, Harper JL, Townsend CR, others. *Ecology. Individuals, Populations and Communities*. USA: Blackwell Scientific Publications; 1986
- [9] Lawrence E. *Henderson's Dictionary of Biology*. London, UK: Pearson Education; 2005. ISBN: 1408234300
- [10] Briscoe G, Sadedin S. Natural science paradigms. Nachira F, Dini P, Nicolai A, Le Louarn M, Rivera Lèon L, editors. *Digital Ecosystems*. Luxembourg: Office for Official Publications of the European Communities; 2007. pp. 48-55. ISBN: 92-79-01817-5
- [11] Dey AK. Understanding and using context. *Personal and Ubiquitous Computing*. 2001;1(5):4-7
- [12] Serbanati LD, Ricci FL, Mercurio G, Vasilateanu A. Steps towards a digital health ecosystem. *Journal of Biomedical Informatics*. 2011;44:621-636. DOI: 10.1016/j.jbi.2011.02.011
- [13] Șerbănați LD, Vasilățeanu A, Nită B. Strengthening context-awareness of virtual species in digital ecosystems. Bucharest: Conference: 19th International Conference on Control Systems and Computer Science (CSCS). 2013:503-510
- [14] Strang T, Linnhoff-Popien C. A context modeling survey. In: Workshop Proceedings. First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp; Nottingham, UK. 7 September 2004. Workshop on, <https://doi.org/10.1.1.2.2060>
- [15] Averian A. A programming model of context-aware applications in digital ecosystems. In: 17th International Multidisciplinary Scientific GeoConference SGEM 2017. Vol. 17; 2017. pp. 37-44. DOI: 10.5593/sgem2017/21
- [16] Averian A. Towards More Context-Awareness in Reactive Digital Ecosystems. In: Proceedings of Creativity in Intelligent Technologies and Data Science. Second Conference, CIT&DS 2017, Volgograd, Russia, September 12-14, 2017. 2017. pp. 640-654. DOI: 10.1007/978-3-319-65551-2_46
- [17] Appert C, Beaudouin-Lafon M, Mackay WE. Context matters: Evaluating interaction techniques with the CIS model. In: Fincher S, Markopoulos P, Moore D, Ruddle R, editors. *People and Computers XVIII—Design for Life: Proceedings of HCI 2004*. London: Springer London; 2005. pp. 279-295
- [18] Bradbury JS, Cordy JR, Dingel J, Wermelinger M. A survey of self-management in dynamic software architecture specifications. In WOSS '04: Proceedings of the 1st ACM. SIGSOFT workshop on Self-managed systems, no. November, 2004. pp. 28-33. DOI: 10.1145/1075405.1075411

- [19] Serugendo GDM, Gleizes MP, Karageorgos A. Self-Organising Software: From Natural to Artificial Adaptation. Springer-Verlag, Berlin, Heidelberg: Springer Science & Business Media; 2011. ISBN: 978-3-642-17347-9
- [20] Briscoe G, Sadedin S, Paperin G. Biology of Applied Digital Ecosystems. Digital EcoSystems and Technologies Conference; DEST'07. Inaugural IEEE-IES. IEEE, 2007
- [21] Horn P. Autonomic Computing: IBM's Perspective on the State of Information Technology. USA: IBM Press; 2001
- [22] Anescu G. A fast artificial bee colony algorithm variant for continuous global optimization problems. University Politehnica of Bucharest Scientific Bulletin Series C-Electrical Engineering and Computer Science; 2017;**79**(1):83-98
- [23] Vasile C, Dumitrache I. Multi-agent membrane systems. Scientific Bulletin Series C. 2016;**78**:3-12
- [24] Lee M, Wolsan M. Integration, individuality and species concepts. Biology and Philosophy. Nov. 2002;**17**(5):651-660. DOI: 10.1023/A:1022596904397
- [25] Hosmer LT. Trust: The connecting link between organizational theory and philosophical ethics. Academy of Management Review. 1995;**20**(2):379-403. DOI: <https://doi.org/105465>
- [26] Ratnasingam P. Trust in inter-organizational exchanges: A case study in business to business electronic commerce. Decision Support Systems. 2005;**39**(3):525-544. DOI: 10.1016/j.dss.2003.12.005
- [27] Malone P, Jennings B. Distributed accountability model for digital ecosystems. 2nd IEEE International Conference on Digital Ecosystems and Technologies IEEE-DEST 2008. 2008. pp. 452-460. DOI: 10.1109/DEST.2008.4635163
- [28] Seuring S. A review of modeling approaches for sustainable supply chain management. Decision Support Systems. 2012;**54**(4):1-8. DOI: 10.1016/j.dss.2012.05.053
- [29] Markus ML, Loebbecke C. Commoditized digital processes and business community platforms: New opportunities and challenges for digital business strategies. MIS Quarterly. 2013;**37**(2):649-654
- [30] Iansiti M, Levien R. Strategy as ecology. Harvard Business Review. 2004;**82**(3):68-81. DOI: 10.1108/eb025570
- [31] Chang E, West M. Digital ecosystems a next generation of the collaborative environment. Eight International Conference. 2006;**214**:3-23
- [32] Walsh WE, Wellman MP. Decentralized supply-chain formation: A market protocol and competitive equilibrium analysis. Journal of Artificial Intelligence Research. 2003;**19**:513-567. DOI: 10.1613/jair.1213
- [33] Walsh WE, Wellman MP. Modeling Supply Chain formation in Multiagent Systems. International Workshop on Agent-Mediated Electronic Commerce. Springer, Berlin: Heidelberg; Vol. 1788; 1999. pp. 94-101
- [34] Averian A. Supply chain modelling as digital ecosystem. In: Proceedings of International Scientific Conference ITEMA 2017; 2017. pp. 27-35

