Chapter 5

Leveraging Internet-of-Things to Support Circular Economy Paradigm in Manufacturing Industry

Giuseppe Pacelli, Enrico Ferrera, Rosaria Rossini,
Ilaria Bosi and Claudio Pastrone

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.80453

Abstract

Circular economy represents a fundamental alternative to the currently predominating linear economy model, while Industry 4.0 is a technological enabler to bring process innovation in the industrial domain. New economic models are needed in order to reduce material inputs and waste generation leveraging on ecodesign, recycling and reusing of products, new business models, and new technologies. Internet-of-Things and artificial intelligence can support the circular economy paradigm, through the development of a marketplace for connecting buyers and sellers of manufacturing services, raw materials and products toward building global supply chains. The core component of this marketplace is a novel, agent-based, brokering module that will apply both syntactic and semantic matching in terms of manufacturing capabilities, in order to find the best possible supplier to fulfill a request for a service, raw materials or products involved in the supply chain.

Keywords: circular economy, industrial symbiosis, Internet-of-Things, agent-based marketplace, artificial intelligence, waste management, trading

1. Introduction

Circular economy represents a fundamental alternative to the currently predominating linear economic model, based on the take-make-consume-dispose paradigm. The linear model has conducted to economic growth and welfare but ran its course. New economic models are needed in order to reduce material inputs and waste generation leveraging on ecodesign, recycling and reusing of products, new business models, and new technologies. Products and production systems need to be designed to be "circular": materials need to be efficiently processed and waste needs to be arranged and recycled.

Looking from a technological perspective, the circular economy generates needs in the field of manufacturing, processing, networking, identification, and recycling of materials and products. As Bányai discussed [1], the success of the operation of these functions is based on the optimal design and control. Most of the supply chain models include only the traditional procurement-production-distribution sub-processes used for converting raw materials to final products and deliver them to the wholesalers, retailers or directly to end users. In this case study is showed a complex green supply chain model in which sophisticated operation research heuristics has to be used to find the optimal solution in order to minimize the costs.

The main needs generated by circular economy are as follows:

- Advanced collection, sorting, and recycling technologies—apps, sensors, robots, etc.
- Efficient materials processing technologies—machine learning and artificial intelligence.
- Production technologies that support design for circularity—3D printing, disassembly, and repairability.
- Interactive platforms for enhanced connectivity—apps, websites, databases, and Internet-of-Things.

Industry 4.0 is fundamental to move from linear to circular economy model. Industry 4.0 refers to the introduction of new technologies, such as the Internet-of-Things, 3D printing, artificial intelligence, cloud computing, machine-to-machine communication, etc., in the traditional industrial domain in order to support the improvement of the efficiency and automation of the processes happening in the industry.

The introduction of circular economy generates new technological and nontechnological needs. The change in ownership and material management concepts, both at a consumer and at business level, generates a need for upscaling and acceleration of business concepts such as: products as a service, sharing platforms, peer-to-peer interactions, and industrial symbiosis [2, 3]. Industrial symbiosis approach focuses on the hidden value of waste resources within an industrial network which can be exploited through the cooperation [4].

Wasted resources are unused or reusable energy, water, materials, and residues of production processes. Industrial Symbiosis can help improve the overall efficiency of the industrial system: companies within an industrial symbiosis network establish beneficial business relationships among them based on these exchanges of resources. However, these relationships are not always easy to establish due to the difficulty in creating the network or lack of a communication platform for the network [5].

The vision of the circular economy is well suited to the concept expressed by the manifesto of the Physical Internet (PI) Initiative [6], a recent concept of breakthrough innovation aiming to improve by an order of magnitude; the economical, environmental, and societal efficiency and sustainability of the way physical objects are moved, deployed, realized, supplied, designed, and used. It attempts to achieve this scope by applying concepts from Internet data transfer to real-world shipping processes [7, 8].

Previous research has already proven that new inventory models enabled by and applied in PI could help reduce inventory levels, thanks to its high flexibility [9]. The vision of the physical Internet involves encapsulating goods in smart, ecofriendly, and modular containers ranging from the size of a maritime container to the size of a small box. That new inventory models enabled by and applied to PI could help reduce inventory levels, thanks to its high flexibility. These modular containers will be continuously monitored and routed, exploiting their digital interconnection through the Internet-of-Things.

Circular economy concepts, and more specifically industrial symbiosis, are vehiculating lot of research in the Internet-of-Things (IoT) domain. Indeed, waste characterization and identification can be considered as the most challenging steps in waste management, that is, the ones that kick off the subsequent valorization and exploitation. Once characterized and identified the waste, the IoT-domain literature proposes a huge number of infrastructures and platforms (e.g., see [10–12]), which helps to monitor features of interest, in real time or on historical base, communicating information only to subjects who are subscribed to specific topics of interest, just when they are available.

The aim of this chapter is to present how Internet-of-Things and artificial intelligence can support the circular economy paradigm, through the development of a marketplace for connecting buyers and sellers of manufacturing services, raw materials, and products toward building global supply chains. The core component of this marketplace is a novel, agent-based, brokering module that will apply both syntactic and semantic matching in terms of manufacturing capabilities, in order to find the best possible supplier to fulfill a request for a service, raw materials or products involved in the supply chain.

Section 2 provides the analysis of the background and state of the art of the multi-agent systems.

Section 3 of this chapter is dedicated to the description of a typical industrial symbiosis case study that the proposed solution intends to address.

Section 4 introduces the solution proposed by this work to support the circular economy, which integrates traditional IoT platforms with an agent-based approach to support a bidding system through the implementation of a marketplace.

Section 5 specifies the type of agents implemented in the marketplace, and finally Section 6 concludes the chapter addressing the future works to extend the platform.

2. Background and state of the art

Multi-agent systems (MAS) [13] have been widely studied in the literature, and some existing options have been evaluated before deciding to implement the marketplace. A multi-agent system is a computerized system composed of multiple interacting intelligent agents. Agent platforms are typically designed with a container paradigm, where all agents live in a well-defined agent container. Among the many platforms available on the Internet, a few subsets of candidates that might cover a good subset of composition features and requirements have been selected.

JIAC (Java-based Intelligent Agent Componentware) [14] is a Java-based agent architecture and framework that allows an easy development as well as the operation of large-scale, distributed applications, and services. The framework supports the design, implementation, and deployment of software agent systems. Then, the entire software development process is supported by JIAC. It also allows the possibility of reusing applications and services, also by modifying them during runtime. The strength points of JIAC are distribution, scalability, adaptability, and autonomy. In fact, JIAC V applications can be developed using extensive functionality that is provided in a library. This library has already-prepared services, components, and agents which can be integrated into an application. For example, individual agents are based on a component architecture which already provides the basic functionality for communication and process management. However, application-specific functionality can be provided by the developer and be interactively integrated.

SARL [15] is a general-purpose agent-oriented language. SARL focuses on providing the fundamental abstractions for dealing with interaction, concurrency, distribution, reactivity, decentralization, autonomy, and dynamic reconfiguration. These are high-level features considered so far as the major requirements for a good and practical implementation of modern complex software applications. SARL approach remains as generic as possible and highly extensible to easily integrate new concepts and features comparing to the variety of existing approaches and meta-models in the field of agent-oriented engineering and multi-agent systems. The language is platform and architecture independent.

JADE (Java Agent DEvelopment Framework) [16] is a software framework that eases the implementation of multi-agent systems through a middle-ware complies with the FIPA [17] specifications and a set of graphical tools that help the debugging and deployment process. Another helpful characteristic concerning JADE-based system is that it can also be distributed across machines with different OS. In a distributed scenario, the configuration of each system deployed can be controlled remotely thought a GUI. The configuration can be even changed at run time by moving agents from one machine to another, if required. Finally, JADE is completely implemented in Java language, and the minimal system requirement is the version 5 of JAVA.

SPADE (Smart Python multi-Agent Development Environment) [18] is a multiagent and organizations platform based on the XMPP/Jabber technology and written in the Python language. This environment offers features and facilities that simplifies the construction of MAS, examples are: the concepts of users (agents) and servers (platforms); an existing communication channel and an extensible communication protocol based on XML, just like FIPA-ACL [19]. Many other exist, but SPADE has been the first agent platform based on XMPP technology.

In **Table 1**, a comparative summary of the candidate agent platforms is reported, where respective pros and cons are addressed.

Platform	Pros	Cons	License
JIAC	Exploits Apache ActiveMQ as transport declarative agent definition through spring	Not FIPA compliant	Apache v2.0
SARL	Exploits ZeroMQ as transport	Not FIPA compliant	Apache v2.0
JADE	FIPA compliant	Supports only RMI transport. Few extensions for JMS/XMPP transports	LGPL
SPADE	FIPA compliant	Uses a custom XMPP server	LGPL

Table 1. Candidate agent platforms.

As it is possible to notice from the presented analysis, none of the platforms offers both a distributed approach and a general purpose implementation for the communication back end.

The platform proposed in this work, therefore, represents a unique example of a marketplace based on semi-automatic multi-agent collaboration.

3. Case study

Internet-of-Things can support the circular economy in many ways. A typical case study consists on the early and automated detection of scrap material levels and its management, also identifying a network of companies which take waste materials as raw material for their production process.

Whenever a desired level (e.g., 50, 75 or 90%) is detected by the scrap material storage area sensors, it will be annotated, reported, and recorded through IoT platform automatically. Getting accurate information about the scrap material storage area levels during the day will now be possible and the company producing the scrap material will stop occupying the staff time to manually detect scrap material levels at each storage area, with the system taking care of such procedure in each factory. Real time data will help optimize scrap material management procedures. Scrap material recycling companies will be either automatically contacted by the platform or they will receive continuous information on the current scrap material level. In response to such an interaction, they will be ready to put their offers to the system, once a new bidding process will be started by the company producing the scrap material. Such process can take place once the scrap material specification and the bidding specifications have been identified, with the company inserting the bid details in the platform. The system registers the new bid request and contacts the waste management companies for getting bids on scrap material. Companies contacted by the system typically belong to a selected subset of contractors. However, according to the company's policies, in some cases, the bid could be open to new contractors, for example, to find better offers fostering new collaborations. When the system contacts contractors for bidding, a deadline is provided on when the last bid will be accepted. Once the deadline is reached, the platform system will select the best (or n-best) offer according to the bidding process criteria (entered when the process started) and present

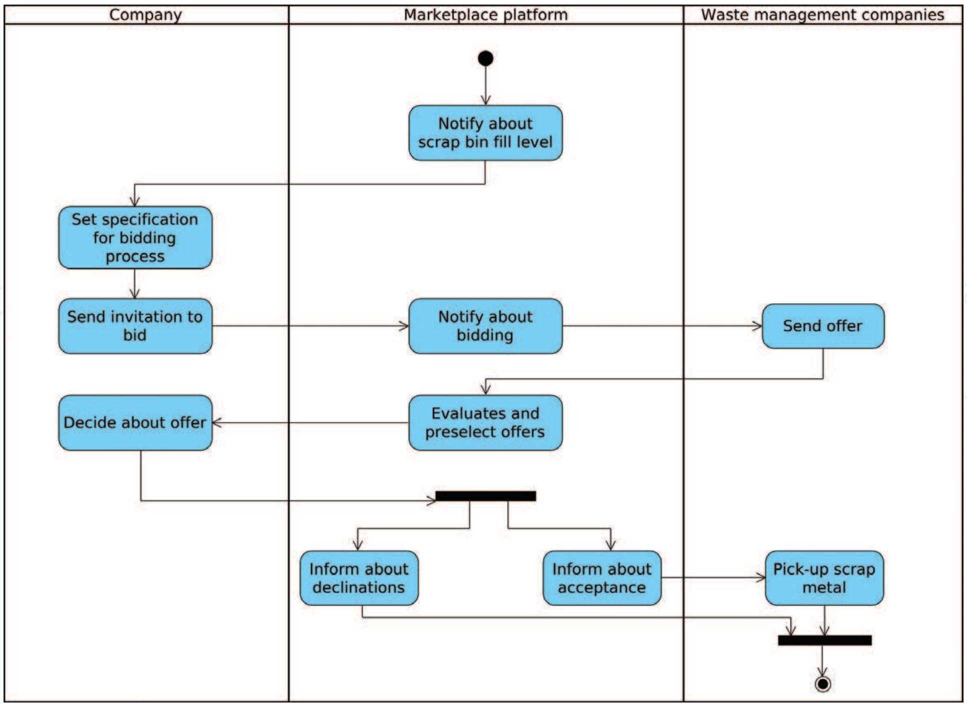
Company
Marketplace platform
Waste management companies
Notify about scrap bin fill level
Set specification for bidding process
Send invitation to bid
Notify about bidding
Send offer
Decide about offer
Evaluates and preselect offers
Inform about declinations
Inform about acceptance
Pick-up scrap metal

Figure 1. Scrap material collection and bidding process.

it to the relevant manager (this automation will reduce the time required for processing bids). Upon human approval, bidders will be automatically informed about the outcomes of their bids. The company will then arrange a pick-up date and time in agreement with the selected scrap material recycling company; such an arrangement will be automatized to the greatest extent (see **Figure 1**).

The criteria to award the contract will be price, company reputation, and the data agreement for the collection of the material. The literature presents several well-known algorithms that can be leveraged for automatic negotiations, such as contract net protocol [20], while many protocols are nowadays considered de facto standards for supporting secure peer-to-peer communication in IoT (such as, XMPP, MQTT, CoAP, and AMQP).

4. The bidding platform a.k.a. marketplace

This section introduces the solution proposed to support the circular economy in the use case presented in Section 3. The proposed platform integrates traditional IoT platforms with an agent-based approach to support a bidding system through the implementation of a marketplace. The marketplace is a fully distributed multi-agent system designed to support Industry 4.0 exchanges between involved stakeholders. It is particularly aimed to supporting automatic supply chain formation and negotiation of goods/data exchanges. The marketplace exploits a microservice architecture, based on Docker, and relies upon a scalable messaging infrastructure. Trust and security are granted in every negotiation step undertaken by automated agents on behalf of involved stakeholders.

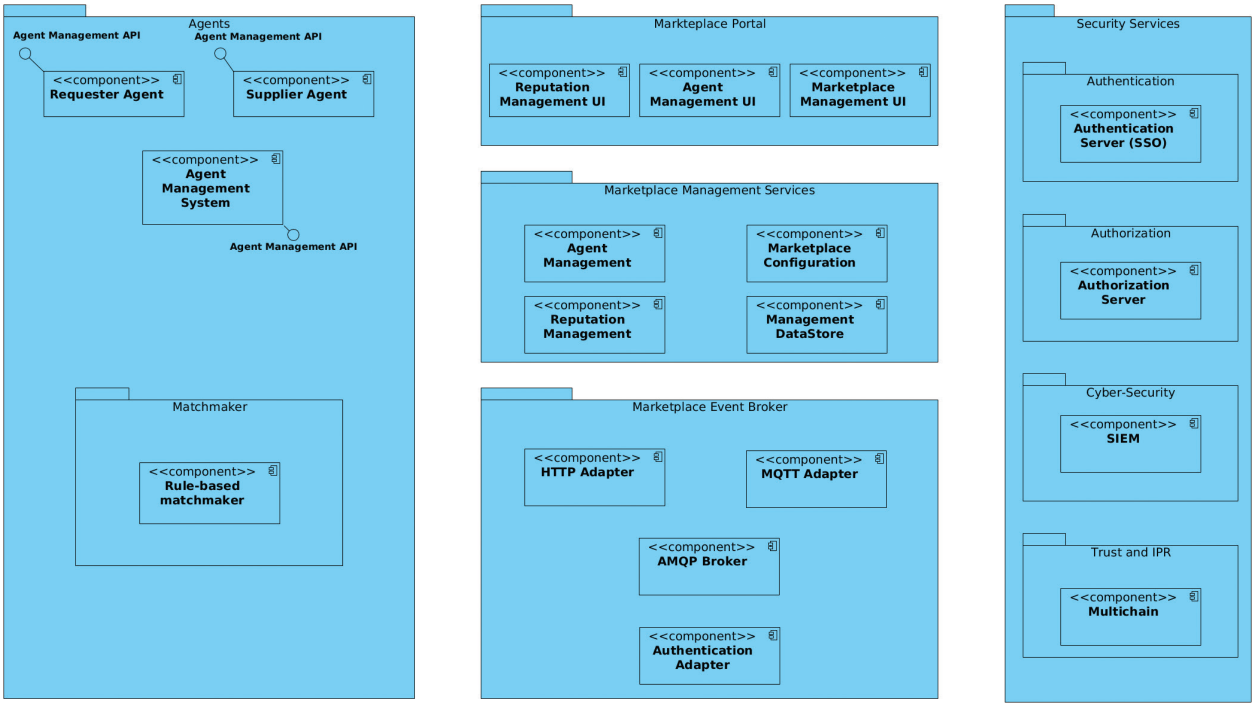
Agents
Agent Management API
Agent Management API
<<component>> Requester Agent
<<component>> Supplier Agent
<<component>> Agent Management System
Agent Management API
Matchmaker
<<component>> Rule-based matchmaker
Markteplace Portal
<<component>> Reputation Management UI
<<component>> Agent Management UI
<<component>> Marketplace Management UI
Marketplace Management Services
<<component>> Agent Management
<<component>> Marketplace Configuration
<<component>> Reputation Management
<<component>> Management DataStore
Marketplace Event Broker
<<component>> HTTP Adapter
<<component>> MQTT Adapter
<<component>> AMQP Broker
<<component>> Authentication Adapter
Security Services
Authentication
<<component>> Authentication Server (SSO)
Authorization
<<component>> Authorization Server
Cyber-Security
<<component>> SIEM
Trust and IPR
<<component>> Multichain

Figure 2. Marketplace components overview.

The marketplace includes the following elements: (a) a marketplace management portal; (b) a marketplace core based on Docker; and (c) a set of "default" agent implementations ready to be adopted by interested stakeholders.

The main building blocks of the marketplace are displayed in **Figure 2**.

4.1. Agent management system

According to FIPA specifications [21], an Agent Management System (AMS) is a mandatory component of every agent platform and only one AMS should exist in every platform. It offers the White Pages service to other agents on the platform by maintaining a directory of the agent identifiers currently active on the platform. Prior to any operation, every agent should register to the AMS to get a valid agent identifier, which will be unique within the agent platform.

4.2. White Pages service

A White Pages service is a mandatory component of any MAS system. It is required to locate and name agents on the system, making it possible for one agent to connect with one another. In the current implementation of the Agent Management System, the agent identifiers are stored in a MySQL database. MySQL has been chosen because it offers relevant features for the project such as on-demand scalability, high availability, and reliability. Other agent platforms, like SPADE, use MySQL as well for the database underlying the White Pages service.

Agents are stored according to the schema depicted in (**Table 2**):

Agent_id	Agent_owner	Agent_role
The unique identifier for the agent. It is the primary key for the table, having the constraint of being unique.	The name of the company owning the agent.	The role of the agent on the marketplace can be either 'Requester' or 'Supplier'.

Table 2. Table schema.

Since the directory service is offered by the AMS, the latter is the only agent allowed to directly interact with the database. When the AMS is executed, it needs the following configurations:

- IP address of the host running the MySQL instance.
- Valid username and password for performing CRUD operations on the database.
- Name of the database to operate on, since the same instance may run different databases.

5. Agents in the marketplace

There are two main categories of agents that can be defined a priori, depending on the kind of services provided:

- Marketplace agents (MA)
- Stakeholder agents (SA)

The first category groups all the agents providing services that are crucial for the marketplace to operate. The second category, instead, groups agents developed and deployed by the marketplace stakeholders to take part in chain formation rounds.

From an implementation point of view, they are very similar and share a large set of features, especially the communication protocols used for the interaction with stakeholders and other agents.

The communication protocols used by the MA for the interaction with the stakeholder on left side and with other agents on the marketplace on the right side are shown in **Figure 3**. Any agent can be controlled by the stakeholder via RESTful APIs which will be listed and explained in the following sections. Agents on the marketplace can exchange messages using Advanced Message Queuing Protocol (AMQP) as transport layer.

Stakeholder agents are instead deployed at the stakeholder's premises, and their purpose is to fulfill the stakeholder's interests. Stakeholder agents can be further divided into two different categories: requester and supplier. In the following, the reference implementations for the two different kinds of stakeholder agents will be described. However, both kinds of agents share a set of features allowing them to respond in different ways to the same kind of solicitations (such as input from GUI, messages from other agents) according to the state of their current behavior (when the solicitation is received).

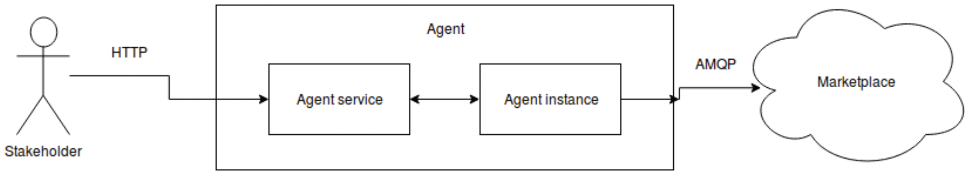

Figure 3. Simplified agents' communication logic.

Currently, agents support two different behaviors:

1. Authentication behavior: This is triggered upon the each agent activation. It consists of all the procedures that are necessary for the agent to correctly operate on the marketplace.
2. Negotiation behavior: This is the behavior each agent needs to support in order to be able to participate in marketplace negotiations. It is designed in a way that allows different negotiation protocols to be adopted.

The authentication behavior is the first one engaged by any agent since:

1. It takes care of getting the agent identifier from the AMS.
2. It authenticates the agent with the messaging broker, allowing secure message exchange over SSL.

Both these phases are mandatory for any agent to operate on the marketplace. After the authentication process, the negotiation behavior comes to play. It allows the agent to perform transactions and interactions with the other agents on the marketplace. Its behavior is constrained to the negotiation protocol that is being used.

Agents might or might not support a certain level of "intelligence" in their decisions regarding certain negotiation protocols. For example, an agent might have the capability of self-evaluation for the received offers, whereas another agent might not have such capability and, therefore, it will involve an external agent (providing such intelligence) during the decision process. An agent having such level of intelligence will be described as "smart," while it will be described as "dumb" in the absence of that.

5.1. Requester agent

The requester agent is the agent used by a factory to request the execution of an existing supply chain or to initiate a new supply chain. Due to the dynamics of exchanges pursued in a factory-oriented marketplace, there is no actual distinction between the two processes, that is, for any supply need a new chain is formed, and a new execution of the chain is triggered. The requester agent may act according to several negotiation protocols, which can possibly be supported by only a subset of the agents active on a specific marketplace instance. In the proposed implementation, it exists a baseline protocol, which must be supported by any agent, called CONTRACT-NET. In such protocol, a requester agent plays the role of "Initiator."

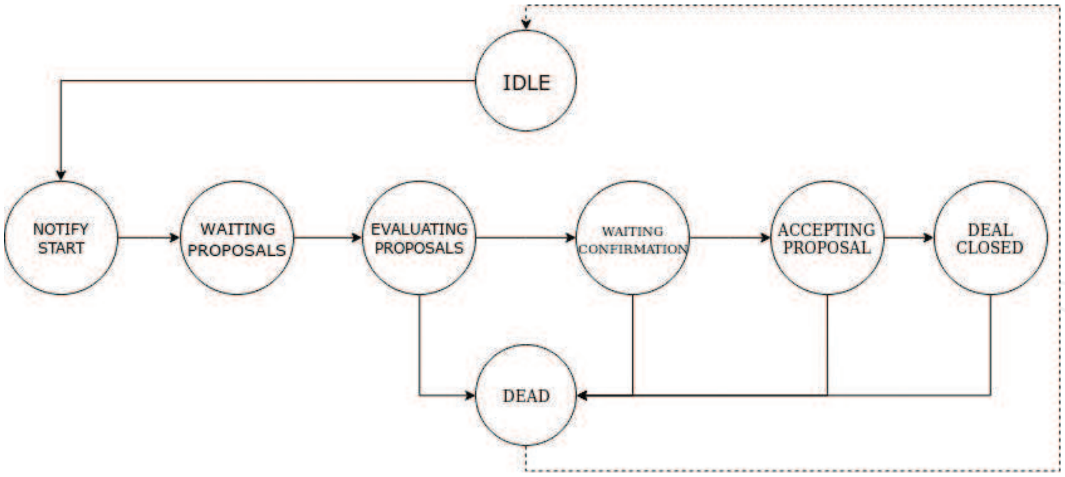

Figure 4. Requester states.

As described by Shoham [22], any agent can pass through a different set of states; the state of an agent consists of components such as beliefs, decisions, capabilities, and obligations. In the CONTRACT-NET, protocol requester agents go through different sets of states, performing different actions upon the receipt of a message (either from the UI or from other agents) according to their current state. In **Figure 4**, the states for requester agent are shown with arrows indicating the allowed transitions between one state and another.

5.1.1. Requester agent negotiation protocol

One of our factory-oriented marketplace strengths is the tight connection existing between intra-factory and inter-factory components. This connection allows the agent on the marketplace to operate in the optimum way to fulfill the stakeholder's needs.

Concerning the requester agent, such connection is the starting point of a new negotiation protocol: the IIMS notifies, through the learning agent (an agent available at the stakeholder's premises, in charge of intra-inter factory communication), that a certain good/service is needed. Upon receipt of such notification, the requester agent can react accordingly, using the most suitable protocol to fulfill the request. A possible interaction is here described from the requester agent point of view:

1. Marketplace system automatically notifies the stakeholder about the scrap metal bin fill level, through a notification from the learning agent (which receives the data from the sensors installed in the factory premises) to the requester agent.

2. Requester agent has the logic for starting a new bidding process through default parameters (set by the stakeholder).

3. Requester agent starts the bidding process, looking for the list of the supplier agents currently available on the marketplace that are capable in replying to such offer.

4. Offers come to requester agent, which can either evaluate them locally (smart agent behavior) or send them out for evaluation toward an external agent.

5. Once the deadline for receiving proposals has expired, requester agent notifies the UI about the best one(s) that have been received.
6. Stakeholder can now choose the preferred offer, select it, and notify the requester agent that can close the deal with the corresponding supplier agent.

The same input that is needed for a bidding process to start, that is, the fill level, can be used by the requester agent to update the supplier agents interested in receiving such information in order to be ready when a new bidding process will take place. The latest described scenario is here described from the requester agent's point of view:

1. The fill level information is received from the learning agent.
2. The requester agent forwards such information, in the marketplace, only to the supplier agent interested in receiving it.

5.2. Supplier agent

The supplier agent is the counterpart of the requester agent on marketplace. It is usually adopted by actual suppliers to respond to supply requests coming from other stakeholders in the marketplace. Factories transforming goods typically employ at least one requester agent to get prime goods and one supplier agent to sell intermediate products to other factories.

In the CONTRACT-NET protocol, supplier agents go through different sets of states, performing different actions upon receipt of a message (either from the UI or from other agents) according to their current state. In **Figure 5**, the states for supplier agent are shown with arrows indicating the allowed transitions between one state and another.

5.2.1. Supplier agent negotiation protocol

A typical supplier agent is registered on the marketplace remaining "silent" while waiting for a new call for proposal, issued by a requester agent. Once this message arrives, the supplier agent:

1. Evaluates the call for proposal message according to the policies set by the agent owner.
2. If the proposal is compliant with such policies, an offer is sent from the supplier agent, otherwise it goes back to the "silent" (IDLE) state.
3. Once the offer has been sent, the supplier agent waits for the requester's decision.
4. In the end, if the requester agent has accepted the proposal, a notification from the supplier agent is sent to the UI for notifying the stakeholder and waiting for his decision.
5. The final message for acceptance is sent back to the requester agent in case the stakeholders accept the deal, otherwise a reject message will be sent.

As already mentioned for the requester agents, the tight connection between intra-factory and inter-factory systems is a strength for the marketplace. For the supplier agent, this connection is exploited to dynamically adapt to the market needs.

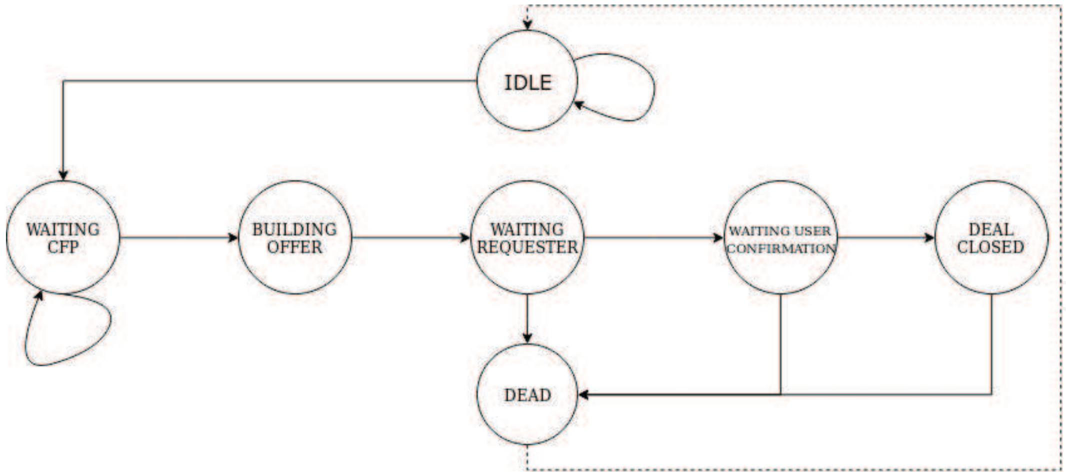

Figure 5. Supplier states.

For the supplier agent, this connection is here briefly described from the supplier agent point of view:

1. The supplier agent subscribes, via AMQP, to the queues where the fill level notification related to a good of a certain company will be received.
2. When such information arrives, the supplier agent can store it to exploit it for predictions of future fill levels.

The same behavior can be used for predicting the prices of a certain good/service. The supplier agent can, in fact, store information about the past bidding process and apply deep learning techniques to have a prediction about the possible future values of interest, that is, what will be the selling price for a certain good at a certain point of time in the future.

6. Conclusions and future works

The proposed marketplace platform shows an approach that enables the paradigm switch from the linear economy model to the circular one. The circular economy paradigm is based upon reuse, reparation, refurbishment, and recycling of existing materials and products and, as shown in the presented case study, these activities can be facilitated through the proposed marketplace approach. Particularly, a company can order the raw materials exploiting a certain supply chain available on the marketplace, process them, and sell the residual production waste through another dedicated supply chain available on the marketplace. On the other hand, a recycling company can buy (or simply collect) the offered waste, properly process it, and sell it again as raw material.

What is important to notice is that, even though existing collaboration models can be exploited on the marketplace, the proposed implementation has been designed to foster new collaboration models within and across the supply chain, making it easier for new actors to participate and collaborate with the existing ones. In order to increase the overall market competitivity while trying to maximize the participant profits, new protocol interactions based on different kinds of auctions are being studied. For instance, Vickrey auction [23] is being investigated, since it has been proved to be efficient, under certain circumstances, in maximizing the expected utility of each bidder.

One of the next important issues to be carefully addressed concerns the trust. Platforms like eBay imply the complete trust of strangers who take money before sending any wished goods. The reputation of sellers is built according to feedback provided by previous buyers and often is very hard for newcomers to sell items on the marketplace because they have to build up a good reputation. With this trust system is very easy to fake reviews in such a way to increase the reputation. Nowadays, Distributed Ledger Technologies (DLT), such as blockchain, are analyzed to be applied in many different application domains. Marketplaces can benefit from the power of DLT since it is built to create trust and transparency. In order to enhance the proposed marketplace approach, the authors are planning to leverage on DLT. DLT have received more and more attention in recent years as innovative methods to store and update data within and among organizations. Multiple copies of the ledger are held by different parties, with data added by consensus and without the need for a third party. This means that DLT are able to offer an immutable record, which means that data added to the ledger is in theory unchangeable, secure, and preserved for the life of the ledger, with the agreement of all participants. Additions to the ledger or changes to the governing structure are decided on a consensus basis by multiple participants. So, these systems provide a transparent and verifiable record of transactions. As a result, DLT can provide gains in trust among bidding participants.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation program through the COMPOSITION project (Grant No. 723145).

Author details

Giuseppe Pacelli[1], Enrico Ferrera[1]*, Rosaria Rossini[2], Ilaria Bosi[1] and Claudio Pastrone[1]

*Address all correspondence to: ferrera@ismb.it

1 Istituto Superiore Mario Boella—ISMB, Turin, Italy

2 Leading Innovation & Knowledge for Society—LINKS, Turin, Italy

References

[1] Bányai T. Optimisation of a multi-product green supply chain model with harmony search. In: DAAAM International Scientific Book. 2011. pp. 15-31

[2] Homrich AS, Galvão G, Abadia LG, Carvalho MM. The circular economy umbrella: Trends and gaps on integrating pathways. Journal of Cleaner Production. 2018;**175**: 525-543

[3] Ferrera E et al. Toward industry 4.0: Efficient and sustainable manufacturing leveraging MAESTRI total efficiency framework. In: Campana G, Howlett R, Setchi R, Cimatti B, editors. Sustainable Design and Manufacturing 2017. SDM 2017. Smart Innovation, Systems and Technologies. Vol. 68. Cham: Springer; 2017

[4] Holgado M, Morgan D, Evans S. Exploring the scope of industrial symbiosis: Implications for practitioners. In: Sustainable Design and Manufacturing. Cham: Springer; 2016. pp. 169-178

[5] Sakr D, Baas L, El-Haggar S, Huisingh D. Critical success and limiting factors for eco-industrial parks: Global trends and Egyptian context. Journal of Cleaner Production. 2011;**19**(11):1158-1169

[6] Montreuil B. Physical Internet Manifesto, Version 1.11. 1. CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation; 2012. Specific reference: Montreuil B. Toward a Physical Internet: Meeting the global logistics sustainability grand challenge. Logistics Research. 2011;**3**(2-3):71-87

[7] Yang Y, Pan S, Ballot E. Mitigating supply chain disruptions through interconnected logistics services in the physical internet. International Journal of Production Research. 2017;**55**(14):3970-3983

[8] Pan S, Ballot E, Huang GQ, Montreuil B. Physical Internet and Interconnected Logistics Services: Research and Applications. 2017;2603-2609

[9] Pan S, Nigrelli M, Ballot E, Sarraj R, Yang Y. Perspectives of inventory control models in the physical internet: A simulation study. Computers & Industrial Engineering. 2015;**84**:122-132

[10] Bonino D. ALMANAC: Internet of things for smart cities. In: 3rd International Conference on Future Internet of Things and Cloud; Rome; 2015. pp. 309-316

[11] Kamienski C, Jentsch M, Eisenhauer M, Kiljander J, Ferrera E, Rosengren P, Sadok D. Application development for the Internet of Things: A context-aware mixed criticality systems development platform. Computer Communications. 2017;**104**:1-16

[12] Brizzi P, Lotito A, Ferrera E, Conzon D, Tomasi R, Spirito M. Enhancing traceability and industrial process automation through the VIRTUS middleware. In: Proceedings of the Middleware 11 Industry Track Workshop; ACM; 2011. p. 2

[13] Wikipedia. Multi-Agent System. https://en.wikipedia.org/wiki/Multi-agent_system [Accessed: July 20, 2018]

[14] Lützenberger M, Küster T, Konnerth T, Thiele A, Masuch N, Hessler A, Keiser J, Burkhardt M, Kaiser S, Albayrak S. JIAC V—A MAS framework for industrial applications. In: 12th International Conference on Autonomous Agents and Multiagent Systems 2013; AAMAS; 2013

[15] SARL. General-Purpose Agent-Oriented Programming Language. http://www.sarl.io/ [Accessed: July 20, 2018]

[16] Bellifemine F, Poggi A, Rimassa G. JADE–A FIPA-compliant agent framework. In: Proceedings of PAAM. Vol. 99. No. 97-108; 1999

[17] FIPA. The Foundation for Intelligent Physical Agents. http://www.fipa.org/ [Accessed: July 20, 2018]

[18] Python Software Foundation. SPADE: Smart Python Multi-Agent Development Environment. https://pypi.python.org/pypi/SPADE [Accessed: July 20, 2018]

[19] FIPA. Agent Communication Language Specifications. http://www.fipa.org/repository/aclspecs.html [Accessed: July 20, 2018]

[20] Smith RG. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers. 1980;**C-29**(12):1104-1113

[21] FIPA. FIPA Agent Management Specification http://www.fipa.org/specs/fipa00023/SC00023K.pdf [Accessed: July 20, 2018]

[22] Shoham Y. Agent-oriented programming. Artificial Intelligence. 1993;**60**(1):51-92

[23] Wikipedia. Vickrey Auction. https://en.wikipedia.org/wiki/Vickrey_auction [Accessed: July 20, 2018]

IntechOpen