Specifying Game-Theoretic Behaviour for Agents in Industrial Supply Chain

Project Acronym	IoT4Industry
Document-Id	D.3
File name	
Version	FINAL document
Date	Start: 01 October 2014 End: 31 March 2015
Author(s)	Dr. Violeta Damjanovic-Behrendt
QA Process	Georg Güntner

Table of Content

IoT4Industry in a Nutshell

IoT4Industry is an exploratory project funded by the Austrian Research Promotion Agency, for the period October 2014 – September 2015. It stands for "Secure, Privacy-preserving Agents for the Industrial Internet". The core research areas of IoT4Industry relate to security, privacy and IPR/data protection requirements, translated into game-theoretic agent behavior, which is simulated in a demo factory floor environment, supporting a supply chain negotiation. A demo factory floor is implemented within our IoT-lab, which is located at Salzburg Research.

While it is still not clear which protocols and technologies will become mainstream for the Industrial Internet, it has become clear that security and privacy will feature strongly in any risk assessment concerning the adoption of practices using the Industrial Internet. Thus, in IoT4Industry, we focus on the use of policy-enacting, multi-agent systems that securely manage machines and manufacturing cells, and we build a feasibility demonstrator based on open source tools and firmware. In addition, IoT4Industry helps us to prepare for internationally recognized contributions to science and technology in the field of Industrial Internet, notably in Horizon 2020.

IoT4Industry Task 3 Description

(from the *IoT4Industry* project proposal)

D3: Specifying Game-Theoretic Behaviour for Agents in Industrial Supply Chains

Goals:

1. Survey paper on game theory and agents
2. Specification formalism for game-theoretic IoT agents

Description of the content

We need to survey in some detail, the current work on game theory (e.g. for supply chain management) and where applicable, the use of agent technology (or concepts) for formulating game theoretic problems in optimisation. On the basis of the survey work, we should be in a position to formally specify IoT Agents that can negotiate with each other, on supply chain optimisation through network effects.

Method

Desktop research on detailed results in game theory and use of agents for game theoretic experiments.

Milestones, results and deliverables

03/2015: Technical report published – formal specification of IoT agents with security concerns

08/2015: Survey paper submitted or close to submission

1. Problem Statement

The *IoT4Industry* project investigates Industrial Internet (*Industry 4.0*) applications and their requirements related to privacy, security and data protection. In D.2 "*IoT Technology Tracking and Industrial Requirements for Security, Privacy and IPR/Data Protection"*, we summarize the major privacy and security concerns that have been recognized so far as the highest rated problems in the adoption of Internet technologies for Industry 4.0. In this document, we further discuss how the agent's behavior can be specified via Game Theory (GT for short) and its mathematical models.

We base our approach on two major technology gaps such as: (1) there exist IoT frameworks, but they do not easily integrate with agent frameworks; and (2) there are tools for strategic analyses based on GT models, but there is no formalism for specifying the GT behaviour of an IoT agent. Hence, in our approach, we firstly survey the current work on GT and where applicable, the usage of agent technology for formulating GT problems in Industry 4.0 (e.g. in optimization, security, privacy, etc.). Secondly, on the basis of the state-of-the-art in supply chain optimisation through network, we formally specify behaviour of *IoT4Industry* agents. Based on our findings presented in D.2, we identify **coexistence of regular and malicious sensor nodes, and their interaction in WSN** as an application of high interest to be implemented in *IoT4Industry'* 3D printing-related supply chain environment. After reviewing currently present agent technologies for managing sensors and machines, and w.r.t. the *IoT4Industry* scenario that is described in D.6 "*Demo Implementation of Industrial IoT Factory Floor based on Arduino and Raspberry Pi*", we specify agents behaviour in *IoT4Industry* in a form of a malicious sensor node detection game.

A malicious sensor node detection game is a signaling game based on Bayesian game with imperfect information, which was initially presented in [GIBB92], and later in [LILJ12] [WACK09]. In our work, we use a signaling game model to solve the coexistence of regular and malicious sensor nodes that we further extend by the *OWASP (Open Web Application Security Project) Internet of Things (IoT) Top 10* project, and its taxonomy of security problems with IoT devices. Furthermore, we focus towards addressing those security problems that can result in data loss or corruption, and can lead to complete device takeover. For example, we focus on insecure Web interface, insecure network services, privacy concerns, insecure cloud interface, and insufficient security configurability.

Document organization. After presenting the state-of-the-art in the field of GT and agent technologies for managing sensors and machines, our work in Section 2 continues towards specifying GT-based behaviour of *IoT4Industry* agents in (additive) manufacturing supply chain. Furthermore, Section 3 presents the *IoT4Industry* signaling game model which incorporates both security and supply chain negotiation aspects of simulated agent behaviour. This model will be further integrated in the proposed agent system and its libraries, which will be discussed later in D.5 *"Security, Privacy, and IPR/Data Protection Requirements Translated into Game-Theoretic Agent Behaviour"*. Finally, Section 4 gives some conclusion remarks.

2. State-of-the-Art in Agent Technology and Game Theory for Industry 4.0

Agent technology is a promising approach for future manufacturing systems, which require a stable reconfigurability, robustness and good responsiveness. The potential of agents is in designing decentralized control systems over distributed autonomous and cooperative system entities. In parallel to agent technology, GT has ability to support the agents in the decision making, when various models of cooperation can be applied, or decisions have to be made either continuously (differential games), or over time (discrete games), etc. Therefore, in *IoT4Industry*, we combine agent technology and GT mathematical models to support the functionality of Industrie 4.0 applications in a more effective way.

In this report, we firstly discuss current agent-based technologies for managing sensor and machines, and look at the examples of agent-based industrial deployment, specifically in the field of supply chain applications. Secondly, we explore the current usage of GT approaches to support web cooperation, and collect examples of GT-based applications in industry and supply chain management. Based on both state-of-the-art technologies for agents and GT industrial deployment, and the *IoT4Industry* scenario, which is presented in D.6 "*Demo Implementation of Industrial IoT Factory Floor based on Arduino and Raspberry Pi*", in this document we make a step further towards specifying the *IoT4Industry*'s agents and their GT-based behaviour*.*

2.1 Agent Technology for Managing Sensors and Machines

According to the definition of FIPA (Foundation for the Intelligent Physical Agent), an agent is "*a kind of entity with autonomy, activity, mobility, reactivity, sociality, intelligence and other* *anthropomorphic features*” [FIPA02][WOOL02]. There are two sets of agent’s technology deployment such as [PEMA08]: (i) multi-agents and (ii) autonomous agents. While **autonomous agent paradigm** is appropriate in application domains with high requirements for systems with decision making autonomy, **multi-agent paradigm** is considered to perform well in the following application domains:

(i) domains with missing data or missing knowledge required for computation,

(ii) domains with a partial or temporary communication inaccessibility,

(iii) domains with restrictions on the information sharing (e.g. e-commerce applications, supply-chain management and e-business),

(iv) domains characterized by time-critical response and high robustness in distributed environment (e.g. time critical manufacturing or industrial systems control, with re-planning, or fast local reconfiguration), or

(v) scenarios that require solving complex problems, or controlling complex systems.

Fundamental challenges in Industry 4.0 applications related to agents are: agent-based decentralized management and coordination of physically distributed sensors and machines (smart objects). For example, decentralized sensors and machines need to autonomously update their responses in a dynamically changing environment, without human intervention. In parallel, they must coordinate their actions with nearby sensors. Such challenges have long been the topic of research in the area of multi-agent systems, resulting in an extensive set of formalisms, algorithms, and methodologies developed so far. One such example is known in the literature as **DCOPs (Distributed Constraint Optimization Problems)**, which brings several algorithms [ROJC09] as summarized below:

- **Algorithms that generate optimal solutions**, such as: Adopt (Asynchronous Distributed Constraint Optimization), DPOP (Dynamic Programming Optimality Principle), and OptAPO (Optimal Asynchronous Partial Overlay). These algorithms are not suitable for sensors that exhibit constrained computational and communication resources (e.g., either the computational cost or the number or size of messages exchanged increase exponentially with the problem size).
- **Approximate stochastic algorithms for solving DCOPs** are typically based on entirely local computation. They maximize a global utility function by having each agent update its state on the basis of the communicated (or observed) states of local neighbours that influence its individual utility. These approaches scale well and are thus well suited to

large-scale distributed applications, but they often converge to poor-quality solutions because agents typically communicate only their preferred state, failing to explicitly communicate utility information.

To address the above shortcomings, the authors in [ROJC09] present the results of the *Adaptive Energy-Aware Sensor Networks* project, which brings an approximate, decentralized solution that maximizes the social welfare of a group of agents (maximizing the sum of each agent's utilities) when any individual agent's utility depends on its own state and the state of a small number of interacting neighbours. This solution is based on the **max-sum algorithm**, a message-passing technique that is often used to decompose complex computations on single processors. It is evaluated on real sensor data within (i) the CNAS (Collaborative Network for Atmospheric Sensing) demonstration project and its sensor network for ground-level atmospheric monitoring, and (ii) the University of Southampton's network of weather sensors, which are located on England's southern coast. The evaluation results shown that the max-sum algorithm produces better solutions than approximate stochastic algorithms (e.g., Distributed Stochastic Algorithm), requires significantly less computational and communication resources than algorithms such as DPOP, and is robust to message loss [ROJC09].

Managing sensors and machines within an IoT setting brings many benefits, as discussed in [ZBSA12], although there are remaining challenges to be solved, such as security, trust and privacy [BAHO08][HU12][WUZM11]. For example, malicious subjects could pose serious threats to the normal operations of networks and damage the credibility of networks through fake services, conspiracy, non-cooperation and other malicious behaviour [WEBE10]. In order to decrease security concerns, increase reliability and credibility, and ensure information collecting, sharing and processing in dynamic IoT environments, the authors in [XUBC13] propose an autonomic agent trust model, that is based on a novel architecture called **TAEC (Trustworthy Agent Execution Chip)**. TAEC provides a reliable platform for the safe operation of agent. The core idea is to use the high-security, cost-effective software and hardware platform for the safe operation of agent. The proposed approach includes installation of TAEC on each sensor node, providing an autonomic trusted execution environment for the agent. The agent can run on the TAEC platform to complete its tasks, which brings double benefits: (i) protecting nodes and (ii) protecting the agents. Finally, the authors in [XUBC13] emphasize that using agent technology to build the credibility protection model for IoT systems requires that **agents and agent's platforms have to be built on all sensor nodes** [XUBC13].

2.1.1 Examples of Industrially Deployed Agent Technologies

The authors in [PEMA08] survey agent technologies with the potential to be deployed in industry domains. They identified the following bottlenecks in fast and massive adoption of the agent-based solutions in industry:

- limited awareness about the potentials of agent technology in industry,
- limited publicity of the successful industrial projects that employ agents, and
- misunderstanding about the technology capabilities, over-expectations of the early industrial adopters and subsequent frustration.

The same authors present several examples of multi-agent usage in industrial areas [PEMA08]:

- shipboard automation distributed control and diagnostic;
- agent-based production planning of engine manufacturing (e.g. mass-production of car engines in SkodaAuto);
- agents in Unmanned Aerial Vehicle (UAV) deconfliction (e.g. Air Force Research Laboratory, NY) [PSPU06];
- agents in RFID (Radio Frequency Identification) for data collection (from RFID readers) and filtering.

Development of agent applications is often supported by international defense industry, and makes an important use of the mental state modeling approaches (such as BDI (Belief-Desire-Intention) approach). In addition, multi-agent systems contribute to the field of distributed diagnostics and partial hypothesis fusion, which is of interest for various industrial companies, e.g. car diagnostics (Toyota, Bosch, and Denso), or supply chain integration in which the supplier and the customer are mutually dependent on the shared data and knowledge. However, at the same time the trading partners may be reluctant to share company sensitive information, since its disclosure may cause their competitive disadvantage. **In such a semi-trusted environment, some auctioning and contracting GT-based techniques may be applicable, which is one of the main research premises in *IoT4Industry*.**

2.1.2 IoT Agents for Industrial Supply Chain

Due to frequent changes in both the production and the market needs, supply chain remains a domain in which the information and data required for efficient planning are dependent on the shared data and knowledge. One of the reason for it could be that the business partners in dynamic supply-chain environment are rather not willing to provide a complete set of information to a centralized planning point [PEMA08], which is calling for an additional information strategy to be applied.

Besides numerous research projects in the field of agent-based logistics and supply chain planning, there is also a substantial commercial success reported in the multi-agent research community, as discussed in [PEMA08]. A typical example of an early adopter of the multi-agent technology in supply chain domain is Whitestein Technologies' **Living Systems/ Adaptive Transport Network (LS/ATN)** that was built for ABX, European logistics company. Whitestein Technologies uses an agent-based solution to (i) achieve performance scalability, (ii) reflect the geographical distribution of the nodes, (iii) provide local re-planning without the need to rebuild the whole plan, and (iv) increase robustness in a way that a single point of failure could be easily avoided. Whitestein Technologies carried out several performance tests to determine the overall cost saving potentials of the LS/ATN system. For example, they achieved 11.7% cost saving, based on performance analyses of 3500 transportation requests [DOCA05].

Another successful agent-based commercial deployment is called **I-scheduler**, which is a logistics scheduling system that provides a decision support to 46 Very Large Crude Carriers (VLCC) [HISW05]. Besides the scheduling of ship operations, I-schedulers is successfully deployed in road transportation applications such as those used by several UK road logistics operators. These applications have been tested with two sets of client data involving 50 and 200 trucks [HISW06]. The agent technologies are successfully applied not only in supply chain, but also in the domains such as production planning, e.g., **mass-oriented production** and **project-oriented production** [PASA05]. For example, **ExPlanTech** is one of the successfully deployed production planning multi-agent systems for monitoring and data collection. One of the important use cases of this system is decision making support that describes how changes in resource availability (e.g., hiring new people), in individual project due dates and project's priorities affects the global operation of the factory. ExPlanTech is integrated in Modelarna Liaz's local ERP system (a shop that manufactures casts, forms, and moulds for the leading European car makers) [PEVB05] [PRCV06].

The **AgentSteel System** for online planning of the steel production is a result of the cooperation between the Saarstahl AG and DFKI GmbH [JMLF05]. This system explores the InteRRaP multi-layered generic agent architecture [MULL96], which integrates reactive and deliberative behaviour, and is further integrated with the ITenvironment of Saarstahl AG.

Finally, several other research projects that investigates deployment of agent technology in virtual organization, supply chain management and inter-enterprise interoperability provisioning are discussed in [PEMA08]; e.g., Conoise (http://www.conoise.org/), Ecolead (see: http://www.ecolead.org), Athena (see: http://www.athena-ip.org/), AgentCities (see: http://www.agentcities.org/), COAX (see: http://www.aiai.ed.ac.uk/project/coax/). The authors in [SEFA07] investigate supply chain planning mechanisms supported by multi-agent systems, considering the state-of-the-art agent technologies in supply chain, such as Fox at al. [FOBT00], Swaminathan [SWAM98], Strader et.al. [STRA98], Montreuil et al. [MOFA00].

2.2 Game Theory and Web-Based Cooperation

John von Neumann and Oskar Morgenstern are considered as fathers of modern GT, which basic concepts have been presented in "*Theory of Games and Economic Behaviour*" [NEMO44]. In *IoT4Industry*, we are particularly focused on the applications of GT to Supply Chain Management (SCM). The authors in [CANE04] emphasize three GT-based approaches with the potential to be used in SCM, such as: (i) static games: non-cooperative, non-zero sum, and cooperative games, (ii) dynamic games: differential games, and (iii) games with asymmetric/incomplete information.

In the following, we summarize the main characteristics of each of the tree GT categories, as discussed in [CANE04].

- **Non-cooperative static games:** John Nash formally introduced the solution concept to non-cooperative static games in 1950. Here, the agents select strategies simultaneously and are thereafter committed to these strategies, e.g., simultaneous move, one-shot games. Non-cooperative GT seeks a rational prediction of how the game will be played in practice. From the perspective of SCM, it is not clear how a manager implements mixed strategy. Hence, the authors consider only non-cooperative games with pure strategies.
- **Cooperative static games:** To model situations and strategies to support sharing aspects and the synergy of cooperation, game theorists developed cooperative games. By solving a cooperative game, we calculate how much each agent should pay in a cost situation (to lower costs), or to receive in a profit situation (to increase profits). Cooperative GT allows modeling outcomes of complex business processes (e.g., negotiations), and answers questions, such as how well is the firm positioned against competition [BRST96]. In cooperative games, agents are free to form any coalitions that are beneficial to them.

There are several solution approaches to cooperative games, in which players share utility via side payments (called **transferable utility cooperative games**). For example, the Shapley value [SHAP53] has found numerous applications in economics and political sciences, considering the issue of fairness through the following conditions:

  - players not adding any value to the game should not benefit from it,
  - changing the names of the players should not affect their allocations, and
  - additivity of payoffs.

- **Discrete dynamic games** (games in which decisions are made over time): The authors in [STAC34] introduces the simplest possible dynamic game with sequential moves (in discrete time). Apart sequential moves, in dynamic games with simultaneous moves both agents take actions in multiple periods. There are two major categories of multiple period games: games with time dependencies and without time dependencies.
  - **Multiple period games without time dependencies:** These games are known as repeated games, because the exactly same game is played over and over again. Although repeated games have been extensively analysed in economics literature, they have not found many applications in the SCM literature.
  - **Multiple period games with time dependencies:** In these games agent's payoffs in each period of time depends on both previous and current actions. These games have found wide applications in SCM, where they are known as **stochastic games or Markov games**. Stochastic games were developed by Shapley [SHAP53a] and later extended by Sobel [SOBE71] [KISO74] [HESO84]. The theory of stochastic games is also presented in [FIVR96].
- **Differential dynamic games:** Differential games provide a natural extension for decisions that need to be made continuously. Although theory for stochastic differential games does exist, applications in SCM are quite limited [BAOL95].
- **Signaling games:** Cachon and Lariviere in [CALA01] consider a model of a signaling game with one supplier and one manufacturer. The manufacturer always benefits from a larger installed capacity, in case demand turns out to be high, but it is the supplier that bears the cost of that capacity. Hence, the manufacturer has an incentive to inflate her forecast to the supplier, and the supplier should view the manufacturer's forecast with skepticism.
- **Screening games:** In a screening game, the player who lacks information is the first to move. For example, in case that a game is based on the supplier-manufacturer game

from [CALA01], the supplier makes the contract offer. In the economics literature, this is also referred to as *mechanism design*, because the supplier is in charge of designing a mechanism to learn the manufacturer's information [POWH99].

- **Bayesian games:** In a Bayesian game, each player uses Bayes' rule to update his belief regarding the types of the other players. See [FUTI91] for more information on Bayesian games. An example of a Bayesian game is the capacity allocation game, that is studied in [CALA99], in which a single supplier has a finite amount of capacity. In this game, there are multiple retailers, and each of them knows his own demand, but not the demand of the other retailers. The supplier announces an allocation rule, the retailers submit their orders, then the supplier produces and allocates units. If the retailer's total order is less than capacity, then each retailer receives his entire order. If the retailer's total order exceeds capacity, the supplier's allocation rule is implemented to allocate the capacity. The issue is the extent to which the supplier's allocation rule influences the supplier's profit, the retailer's profit and the supply chain's profit.
- **Biform games:** Biform games, developed by [BRST03], can be thought of as "a non-cooperative game with cooperative games" that leads to specific payoffs. Biform games have been successfully adopted in several SCM papers. The authors in [ANBZ01] consider a game in which multiple retailers stock at their own locations as well as at several centralized warehouses. The authors in [GRSO01] analyse a similar problem, but allow retailers to hold back the residual inventory. The authors in [PLTA01a] [PLTA01b] analyse two similar games between two firms, with an option of pooling their capacity and investments in order to maximize the total value. In the first stage, firms choose investment into effort that affects the market size. In the second stage, firms bargain over the division of the market and profits.

2.2.1 Game Theory for Industrial Supply Chain

In SCM, Web-based cooperation between suppliers and customers goes far beyond the "I-cut, you-choose" method, or simple divisible procedures that consider proportionality and envy-free allocation of resources. In fact, one of the main features of supply chain is – cooperation, which is defined as the process of coordinating goals and actions of agents [THUN05]. This is how GT become an essential tool in the analysis of supply chains with multiple agents, especially when the agent's objectives are conflicting.

In terms of specific applications in SCM, the authors in [HADS00] consider the **newsvendor centralization game**, in which multiple retailers decide to centralize their inventory and split profits resulting from the benefits of risk pooling. [HADS00] further show that this game has a non-empty core under certain restrictions on the demand distribution.

As noticed by [CANE04], most of the existing GT approaches in SCM utilize only a few GT concepts, in particular concepts related to non-cooperative games. Some attention has been given to stochastic games but several other important areas need additional research, such as cooperative, repeated, differential, signaling, screening, and Bayesian games. Marketing and economics have been far more successful in applying differential games since deterministic models are standard in these areas.

Apart SCM, there are other industrial domains in which GT has been used so far, such as sensor networks with incentives for forwarding nodes [BUBO02] [WACK08] and for punishing misbehaving nodes [SRNC03]. The autonomous sensor network can make a use of non-cooperative GT techniques to optimise the system. For example, Nash Equilibrium can be used to get optimal solutions for energy conservation.

GT is also a powerful tool for Wireless Sensor Network (WSNs), energy harvesting technologies [NHRB07], Game Theoretic Energy Balance Routing (GTEBR) algorithm for uneven energy consumption [JICJ07], the energy efficient self-organization protocol [OXZO04], decentralized adoption in sensor networks for managing sensor activity with low coordination [CHTA04], etc. GT could be seen as a powerful tool for the design and control of multi-agent systems [RSSG10] [GMWI11], which requires two steps:

- Modeling the agent as self-interested decision maker in a GT environment (distribution of the optimization problem (game design)), and
- Specifying a distributed learning algorithm that enables the agents to reach a desirable operating point, e.g., the Nash Equilibrium of the designed game.

3. IoT4Industry Agents and their Behavior in Industrial Supply Chain

Our motivation to explore multi-agent systems and GT in an IoT industrial setup in the *IoT4Industry* project, can be summarized through several identified technological problems:

1) Frameworks for IoT: There are various IoT frameworks being developed and even marketed already, which differ in at least three dimensions:

- their technological readiness level;
- their use in different value propositions, e.g. Xively, Google NEST; and
- their openness in terms of source code licensing and API usage by third parties.

2) Industrial requirements for IoT: Knowledge-based "semantic" agents have not been widely explored in the past 15 years of Semantic Web research. The Eclipse-based JADE platform is one of the most comprehensive agent platform for research, but it is neither integrated with IoT, nor has it been exposed to industrial applications in manufacturing, nor it is modular as would be needed for ad-hoc networking scenarios and cross-company interoperation scenarios. Also, the FIPA agent protocol needs to be "stress-tested" against industrial and IoT requirements.

3) Formal description of the GT behaviour of agents: Society can expect that normally machines comply with technical regulations and safety guidelines. In the Io, and in business settings in general, the same will apply and therefore, agent-like software behaviour will have to demonstrate compliance with regulations. One promising approach could be to demonstrate certain GT properties of agents, e.g. truthfulness or compliance with specific GT setups.

While there is a body of over 40 years of mathematically well-founded research (known as **operations research**), very little of this knowledge has found its way into applied computer science, let alone web-based e-commerce. Hence, in this exploratory action, we experiment with creating environments where agents may differ in strategy, but are at least, auditable (this means we can analyse their actions and decision processes post-hoc), so as to understand the dynamics when e.g. logistics and manufacturing agents negotiate and renegotiate delivery plans between suppliers and customers, in order to optimise for a network benefit (e.g. in a multi-actor SCM scenario).

4) Omniscience in the Web is impossible: GT addresses a fundamental characteristic of our world and the World Wide Web: actors always have limited knowledge, have differing knowledge, can be truthful or less truthful, and may be informative or less informative. The WWW as a software-driven machine gives us to a large degree, the possibility to specify rules that must be obeyed by machines and the possibility to specify machine behaviours that differ widely from each other, but at the same time, comply with the overall rules of the game. One interesting research question is then: what kinds of agent behaviour should be "freely programmable" and what kinds of agent behaviour need to be externally constrained, verifiable, or at least auditable and to what extent can a software-based environment enforce such mandatory behaviour?

5) Compliance tests for security, privacy and overall trustful behaviour: Machines that make decisions on behalf of humans put the onus of responsibility on the organisations that make use of the machines. In this exploratory action, we would like to experiment with ways to audit and trace agent behaviour, and to test agents for mandatory behavior before they can be "released".

The design and the development of the *IoT4Industry* agent system is based on the description of architecture from D.6 "*Demo Implementation of Industrial IoT Factory Floor based on Arduino and Raspberry Pi*". Once we are able to simulate simple supply chain management and monitor manufacturing processes (additive manufacturing processes), we will then extend the functionality of the agents by adding GT behaviour patterns to the test bed. This requires some further research into how agent behaviour must be separated from environment-behaviour, especially to cover the Industry 4.0 requirements related to dynamic decision making and secure operation of supply chains. This will be the topic of D.5 "*Security, Privacy, and IPR/Data Protection Requirements Translated into Game-Theoretic Agent Behaviour*". In the following, we specify the *IoT4Industry* agent behaviour to implement GT models, by providing the best strategies for security and privacy in an IoT-based SCM industrial setup.

3.1 Specifying GT-based Agent Behaviour in Industrial Supply Chain

IoT4Industry explores applicability of signaling GT as a methodology in SCM. As discussed by Cachon and Lariviere in [CALA01], one solution for a high demand manufacturer is to give a sufficiently large lump-sum payment to the supplier: the high demand manufacturer's profit is higher than the low demand manufacturer's profit, so only a high demand manufacturer could offer that sum. This has been referred to as signaling by "burning money". A better signal is a contract offer that is costless to a high demand manufacturer but expensive to a low demand manufacturer. An example of such a signal is a minimum commitment, which is costly only if realized demand is lower than the commitment, because then the manufacturer is forced to purchase more units than desired. That cost is less likely for a high demand manufacturer, so in expectation a minimum commitment is costlier for a low demand manufacturer.

In *IoT4Industry,* the agents "perceive" their environment. One class of percepts (e.g., agents' observations) collects the mandatory rules of the game/ strategy in which the agents participate.

For example, the core of collaboration in SCM is based on a relationship between supply and demand. Effective planning of supply and demand is the critical task that needs to be designed and mathematically modeled. Hence, we firstly design the supply and demand collaboration graph, shown in Figure 1, which illustrates the way both demand and supply face the market's needs when purchasing or selling their units, while the price is a key factor for them to get profit.

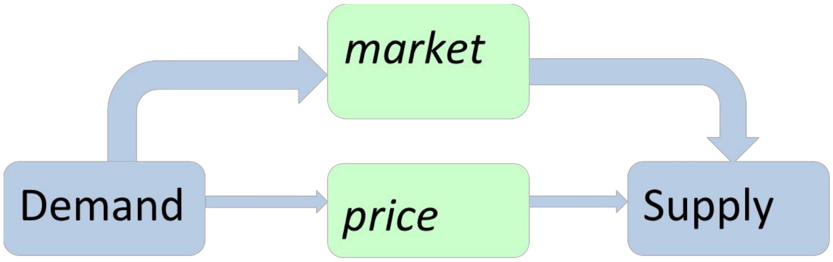

Figure 1: Demand-Supply collaboration graph in *IoT4Industry*

Secondly, we design a demand-supply collaboration game model by adopting collaborative planning model that is presented in [ZHYH05]. In the following, both the optimal demand and the optimal supply models, are given. Both models utilize the bargaining games, striving to maximize their profits.

3.1.1. The Optimal Demand Model (maximizing profit)

Note that the optimal demand model is based on [ZHYH05], in which *t* stands for time, and *d* denotate demand. The decision variables of the demand model are as follows:

$x_d(t)$: the output of demand to supply at time *t*;

$y_d(t)$: the output of demand to market at time *t*;

a_d: the bargaining price between demand and supply;

$a_d(t)$: the price of demand to market at time *t*;

b_d: the unit cost;

c_d: the unit tax value;

h : the unit stock charge;

W: The maximum limit of stock;

I(0): The initial value of demand stock;

m: The resources' quantities of demand.

The optimal demand model that maximizes demand profit is defined in (1):

$$\max \sum_{t=1}^{T} \left\{(a_d - b_d)\, x_d(t) + (a_d(t) - b_d - c_d)\, y_d(t) - h\, I(t)\right\} \qquad (1)$$

where $I_d(t) \leq W$, (t = 1, 2, ...T) represents stock limit of demand.

3.1.2. The Optimal Supply Model (max. profit)

Similar to the optimal demand model, in the optimal supply model (denoted by *s*), the decision variables can be defined as follows:

$x_s(t)$: the output of supply to demand;

$y_s(t)$: the output of supply to market;

a_s: the bargaining price between demand and supply;

$a_s(t)$: the price of supply to market at time *t*;

α: the converted ratio;

b_s: the unit cost;

c_s: the unit tax value;

h : the unit stock charge;

W: The maximum limit of stock;

I(0): The initial value of supply stock;

m: The resources' quantities of supply.

The optimal supply model that maximizes supply profit is given in (2):

$$\max \sum_{t=1}^{T} \{\alpha(a_s - b_s - c_s)(\, x_s(t) + y_s(t)) - (a_s x_s(t) - a_s(t) y_s(t) - h\, I(t)\} \qquad (2)$$

where $I_S(t) \leq W$, (t = 1, 2, ...T) represents stock limit of demand.

3.1.3. The Demand-Supply Model as a Signaling Game

We start from the basic definition of the signaling game model, which is discussed in [GIBB92] [LILJ12], and which we further translate according to *IoT4Industry* requirements:

A signaling game is a dynamic game of incomplete information involving two agents. In *IoT4Industry*, we refer to agent 1, who acts on the demand side as a Buyer (*Buy*), and to agent 2, on the supply side, as a Seller (*Sel*). We can also put both agents *Buy* and *Sel* into a different context: a manager agent monitors the game, communicates with other agents about status of each payouts, share common knowledge among plays (but not private knowledge of agent), determines winners or applies other game rules, e.g. time limit of a game, etc. The manager agent also provides the rules of the game to the agents.

In case of 3D printing farm, the timing of a signaling game can be defined as follows:

1. The buyer from the 3D printing farm needs to buy a filament in a specific colour, quality, and quantity (we call it $type_i$ from a set of filaments TYPE = {type_1, ... , type_r}), according to a probability distribution p(type_i), where p(type_i) > 0 for every *i* and p(type_l) + · · · + p(type_r) = 1.
2. The buyer from the 3D printing farm observes filaments $type_i$, then chooses one to buy (we call a signal) m_j from a set of possible signals M = { m_1, ... , m_j}.
3. The seller who provides filament, observes a filament m_j which is selected by the buyer, then chooses an action a_k from a set of feasible selling actions A = {a_1, ... , a_k}.
4. The payoffs of the buyer and the seller agents are given by U_{buy}(t*ype*$_i$, m_j, a_k) and U_{sel}(t*ype*$_i$, m_j, a_k), respectively. Different types of buyers have different payoff functions. Each player knows all information except that the seller doesn't know $type_i$.

A pure-strategy perfect Bayesian equilibrium of a signaling game is a pair of strategies (m_j∗(type_i), a_k∗(m_j)) and a probability distribution P(t_i|m_j), satisfying the following (3-5):

- Given P(type_i|m_j) and m_j, the seller's action a_k∗(mj) maximizes its expected payoff; i.e., a_k∗(mj) solves:

$$max_{ak} \sum_i P(type_i|m)\ U_r(type_i, m, a_k) \tag{3}$$

- Given a'(m), the buyer's signal m'∗(type_s) maximizes its payoff; i.e., m'∗(type_s) solves:

$$max_{mj}\ U_s(type_s, m_j, a'(m_j)) \tag{4}$$

- For each m, if there exists type_i such that m'(type_i) = m,P(type_i|m) follows Bayes' rule and the buyer's strategy:

$$P(\text{type}_i|\text{m}) = \begin{cases} \dfrac{P(type_i)}{\sum_{j,m'(t_j)=m} P(type_j)} & if\ m'(type_i) = m \\ 0 & otherwise \end{cases} \tag{5}$$

3.2. Specifying GT-based Agent Behaviour for Security, Privacy and Data Protection

In order to protect the transmission of code and data, we can rely on traditional network security technologies, which are mature and effective. For the problem of viruses attacking execution environments and host systems, there is a series of methods available, such as the sandbox, digital signature, authentication, authorization and resource allocation, proof code carrying, code inspection and audit. However, it is more challenging to protect task codes and data attacked by execution environments and host systems.

Therefore, in *IoT4Industry* we specify signaling GT-based agent behaviour covering security requirements, which are related to tasks and data attacks. Our game model involves two players: an attacker (called *Att*) and a defender (called *Def*), and two basic moves: (i) a signal sent by the attacker, and (ii) a response taken by the defender. The communication between the attacker and the defender occur every time when the defender receive a “signal” about the suspicious behaviour in the environment, or about the presence of the attacker. Hence, we firstly define perfect Bayesian equilibrium for signaling games and describe the various situations when the communication between the attacker and the defender exist, ranging from zero information, via incomplete, up to complete information settings. We define equilibrium for such communication situation between *Att* and *Def*, which is determined by the costs of signals between (i) *Att* and *Def*, and (ii) *Att* and the core system.

The defined equilibrium will be later translated into the agent behaviour. A player's strategy in any game could be presented in a form of a plan of actions. Firstly, we define all major possible attack points, as shown in Figure 2. For example, the attacker can try to access the IoT system by attacking either sensor nodes (or near sensor agents), or the IoT gateway, or the core IoT system. Note that the IoT Gateway is added as an important part of IoT architecture, with the role to:

- provide sensors with a single point of contact with external networks by using WiFi, GSM, or some other type of connectivity;
- perform the pre-processing of information (e.g. message filtering and aggregation), before they are sent to the data center;
- handle system logging every time it is required;
- act as a single point of access for monitoring the selected area of the environment;

- provide failure and disaster recovery in case of any action that may result in an interruption of gateway processing, at the first line of protecting the core IoT system;
- automatically provide critical security fixes; etc.

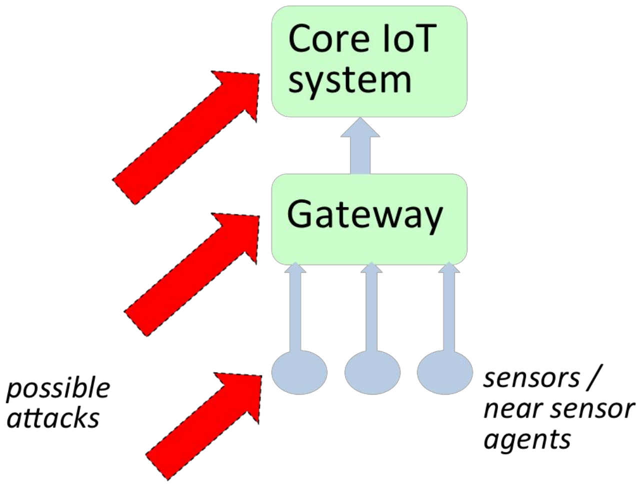

Figure 2: Possible attack points in *IoT4Industry*

A pure strategy for *Att* in a signaling game is a function m(*type*$_i$) specifying which message (signal) will be chosen for each type that nature might draw, while a pure strategy for *Def* is a function a(*m*$_j$) specifying which action will be chosen for each message that *Att* might send. A set of pure strategies can be illustrated by using the attack graph as shown in Figure 3.

Different attackers have different interests (which is here presented in a form of a payoff function). In order to simulate a set of possible strategies in *IoT4Industry*, we additionally take into consideration the *OWASP (Open Web Application Security Project) Internet of Things (IoT) Top 10* project. OWASP summarizes all known security problems with IoT devices, and methods to prevent them, through the following categories:

- I1: Insecure Web Interface
- I2: Insufficient Authentication/Authorization
- I3: Insecure Network Services
- I4: Lack of Transport Encryption
- I5: Privacy Concerns
- I6: Insecure Cloud Interface

- I7: Insecure Mobile Interface
- I8: Insufficient Security Configurability
- I9: Insecure Software/Firmware
- I10: Poor Physical Security

We are particularly interested to address security problems that can result in data loss or corruption, which therefore, can lead to complete device takeover. Hence, we focus on insecure Web interface (I1), insecure network services (I3), privacy concerns (I5), insecure cloud interface (I6), and insufficient security configurability (I8) when defining the player's pure strategies. As an example attack scenario, we refer on OWASP's case of the web interface reporting "Forgot Password" message, upon entering an invalid account which informs the attacker that the account does not exist. Once valid account is identified, password guessing can begin for an indefinite amount of time if no account lockout controls exist [OWASP].

In Figure 3, we translate the above strategy into the attack graph, by using the following notation: L denotes a location of the attack, which could be either sensor nodes, or IoT Gateway, or IoT core system (see red arrows in Figure 2), and (ii) C denotes consequences of the attack:

- L1: attacking sensors/ near sensor agents
- L2: attacking IoT gateway
- L3: exploit sensor gateway using changed username and/or password
- L4: exploit sensor gateway using cross-site forgery
- L5: attacking IoT core system
- L6: direct attack on IoT core system
- C1: access via sensors/ near sensor agents
- C2: access via gateway
- C3: access based on changing the default username and/or password
- C4: access based on cross-site scripting
- C5: SQL injection, or data loss and corruption.

Figure 3 illustrates the attack graph. In order to reach the target, the attacker would firstly try to access the IoT system via sensors or near sensor agents. After “conquering” sensors/near sensor agents, the attacker goes further by either attacking the IoT gateway (L2) or the IoT core system (L6), which will directly lead to data loss, or corruption of the system (C5). After attacking the IOT gateway, the attacker can either exploit sensor gateway using changed username and/or password (L3), or using cross-site forgery (L4). Both consequences, C3 and C4, would only lead to further attacking of the core system (L5) and data loss and corruption (C5).

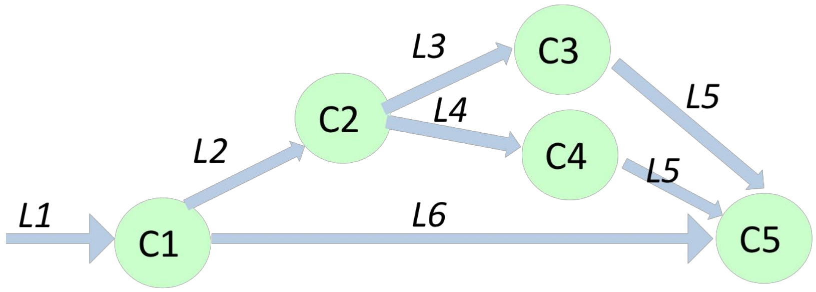

Figure 3: Attack graph

3.2.1. The IoT4Industry Security Model as a Signaling Game

After defining the attack graph, we need to define perfect Bayesian equilibrium for signaling games with two players: *Att* and *Def*. We also need Bayesian equilibria to satisfy the following two criteria [GIBB92]:

1. Sequential Rationality. Whenever an agent is called to play, she does something optimal;

2. Consistency of beliefs. What is optimal often depends on what an agent believes about the opponent(s), and we want to rule out an agent thinking something that is contradicted by the equilibrium strategies.

The definition below capture the above criteria in an easy notation [GIBB92]:

Definition 1. A (pure) perfect Bayesian equilibrium in a signaling game (of the form described above) is a strategy profile s^* and a system of beliefs μ such that

1. $s_1^*(\theta)$ solves $max_{a1 \in A1} u_1(a_1, s_2^*(a_1); \theta)$ for all $\theta \in \Theta$

2. $s_2^*(\theta)$ solves $max_{a2 \in A1} \sum_{\theta \in \Theta} \mu(\theta \mid a_1)\, u_2(a_1, a_2, \theta)$ for all $a_1 \in A_1$

3. $\mu(\theta \mid a_1)$ satisfies Bayes rule whenever applicable (e.g., whenever dividing by zero could be avoided).

The attacker's chosen strategy (attack action) corresponds to the chosen attack path (Figure 3). The attacker goal is to find the strategy that maximizes her expected payoff (see the equation (3) Section 3.1.3) by estimating whether each causality relation is possible or not. The attacker compares her expected payoff for each path, and chooses the best one. Similarly, the main goal of the defender is to find the strategy that maximizes her expected payoff, and for that purpose, we use the equation (4) from Section 3.1.3.

4. Conclusion

One of the main research premises of the *IoT4Industry* project is to overcome some of the greatest barriers that largely inhibit business from adopting the Industrial Internet, such as barriers related to security concerns. As found through many interviews with the industrial partners, the industry is not very positive when it comes to sharing company sensitive information. Therefore, in *IoT4Industry* our proposal is to use certain GT-based techniques such as signaling games, which incorporate both security and supply chain negotiation aspects of agent behaviour, without revealing sensitive information to the outside world.

In this report, we specify our GT-based signaling game model, which should be further integrated into the proposed agent system (agent libraries) and elaborated in D.5 *"Security, Privacy, and IPR/Data Protection Requirements Translated into Game-Theoretic Agent Behaviour"*.

References

[ANBZ01] Anupindi, R., Y. Bassok and E. Zemel. 2001. A general framework for the study of decentralized distribution systems. Manufacturing & Service Operations Management, Vol.3, 349-368.

[BAHO08] A. Bassi, G. Horn. Internet of Things in 2020: Roadmap for the Future, EPoSS, Brussels, EU. Online: http://bit.ly/1IZeYwb; 2008.

[BAOL95] Basar, T. and Olsder, G.J., 1995. Dynamic Noncooperative Game Theory. SIAM, Philadelphia

[BRST03] Brandenburger, A. and H.W. Stuart, Jr. 2003. Biform games. Working Paper, Columbia University.

[BRST96] Brandenburger, A. and H.W. Stuart, Jr. 1996. Value-based business strategy. Journal of Economics and Management Strategy, Vol.5, No.1, 5-24.

[BUBO02] S. Buchegger & J. Le Boudec, “Performance Analysis of the CONFIDANT Protocol,” Proc. 3th ACM Int’l Symp. Mobile AdHoc Networking & Computing, pp. 226-236, 2002.

[CALA01] G. Cachon, & M. Lariviere. 2001. Contracting to assure supply: How to share demand forecasts in a supply chain. Management Science. Vol.47, 629-646.

[CALA99] G. Cachon, & M. Lariviere. 1999. Capacity choice and allocation: Strategic behavior and supply chain performance. Management Science. Vol.45, 1091-1108

[CANE04] Cachon, G., and Netessine, S. 2004. Game theoretic applications in supply chain analysis. Supply Chain Analysis in the eBusiness Era, David Simchi-Levi and S. David Wu and Zuo-Jun (Max) Shen (Eds.). Kluwer.

[CHTA04] J. Chang & L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks,” IEEE/ACM Trans. Networking, vol. 12, no. 4, pp. 609-619, 2004.

[DOCA05] Dorer, K., and Calisti, M. (2005): An Adaptive Solution to Dynamic Transport Optimization. In: Proc. of AAMAS-05– Industry Track, ACM Press.

[FIPA02] FIPA. FIPA abstract architecture specification. Available from: http://www. fi-pa.org/specs/fipa00001/SC00001L.html; 2002.

[FIVR96] J. Filar, & K. Vrieze. 1996. Competitive Markov decision processes. Springer-Verlag.

[FOBT00] Fox, M.s., Barbuceanu, M. and Teigen, R. (2000). “Agent-oriented supply chain management”, In Int. Journal of Flexible Manufacturing Systems, 12(⅔), p. 165-188.

[FUTI91] D. Fudenberg, & J. Tirole. 1991. Game theory. MIT Press.

[GIBB92] GIbbon, R. 1992. Game Theory for Applied Economists. Princeton University Press, Princeton, New Jersey.

[GMWI11] R. Gopalakrishnan, J. R. Marden, and A. Wierman. An architectural view of game theoretic control. ACM SIGMETRICS Performance Evaluation Review, 38(3):31–36, 2011.

[GRSO01] Granot, D. & G. Sosic. 2001. A three-stage model for a decentralized distribution system of retailers. Operations Research.

[HADS00] Hartman, B. C., M. Dror and M. Shaked. 2000. Cores of inventory centralization games. Games and Economic Behavior, Vol.31, 26-49.

[HESO84] D.P. Heyman, & M.J. Sobel. 1984. Stochastic models in Operations Research, Vol.II: Stochastic Optimization. McGraw-Hill.

[HISW05] Himoff, J., Skobelev, P., and Wooldridge, M. (2005): MAGENTA Technology: Multi-Agent Systems for Industrial Logistics, In: Proc. of AAMAS-05 – Industry Track, ACM Press, pp. 60 – 66.

[HISW06] Himoff, J., Rzevski, G., and Skobelev, P. (2006): Multi-Agent Logistics i-Scheduler for Road Transportation. In: Proc. of AAMAS-06– Industry Track, ACM Press, pp 1514-1521

[HU12) X.D. Hu. The Security of the Internet of Things. Beijing: Science Press; 2012.

[JICJ07] Z. Jia, M. Chundi, & H. Jianbin, “Game Theoretic Energy Balance Routing in Wireless Sensor Networks”, Proc. Chinese Control Conf., pp. 420-424, 2007.

[JMLF05] Jacobi, S., Madrigal, C., Leon-Soto, E, and Fischer, K. (2005): AgentSteel: An Agent-based Online System for the Planning and Observation of Steel Production. In: Proc. of AAMAS-05– Industry Track, ACM Press, pp. 114-119.

[KISO74] A.P. Kirman, & M.J. Sobel. 1974. Dynamic oligopoly with inventories. Econo-metrica, Vol.42, 279-287.

[LILJ12] Lin, J., Liu, P., Jing, J., 2012. Using Signaling Games to Model the Multi-Step Attack-Defense Scenarios on Confidentiality. In Proceedings of the GameSec 2012, LNCS 7638, pp. 118--137, 2012.

[MOFA00] Montreuil et al., 2000. “A Strategic Framework for Networked Manufacturing”, Computer in Industry, Vol. 42, No. 2-3, pp. 299-317

[MULL96] Muller, J.P. (1996): The Design of Intelligent Agents: A Layered Approach. In: LNAI No. 1177, Springer Verlag, Heidelberg

[NEMO44] von Neumann, J. & Morgenstern, O. 1944. The Theory of Games and Economic Behaviour. Princeton: Princeton U. Press, 1944.

[NHRB07] D. Niyato, E. Hossain, M. Rashid, & V. Bhargava, “Wireless Sensor Networks with Energy Harvesting Technologies: A Game-Theoretic Approach to Optimal Energy Management,” IEEE Wireless Comm., vol. 14, no. 4, pp. 90-96, 2007.

[OXZO04] S. Olariu, Q. Xu, & A. Zomaya, “An Energy-Efficient Self-Organization Protocol for Wireless Sensor Networks” Proc. Intel. Sensors, Sensor Networks, and Information Process. Conf., pp. 55-60, 2004.

[PASA05] Paolucci, M., and Sacile, R. (2005): Agent-Based Manufacturing and Control Systems. CRC Press, Boca Raton.

[PEMA08] M. Pechoucek, V. Marik, 2008. Industrial deployment of multi-agent technologies: review and selected case studies. Autonomous Agents and Multi-Agent Systems, Vol. 17, Issue 3, pp 397-431, December 2008.

[PEVB05] Pechoucek, M., Vokrinek, J., and Becvar, P. (2005): ExPlanTech: Multiagent Support for Manufacturing Decision Making. In: IEEE Intelligent Systems, Vol.20, No.1, pp. 67-74

[PLTA01a] Plambeck, E. and T. Taylor. 2001a. Sell the plant? The impact of contract manufacturing on innovation, capacity and profitability. Working Paper, Stanford University.

[PLTA01b] Plambeck, E. & T. Taylor. 2001b. Renegotiation of supply contracts. Working Paper, Stanford University.

[POWH99] E. Porteus, & S. Whang. 1999. Supply chain contracting: non-recurring engineering charge, minimum order quantity, and boilerplate contracts. Working paper, Stanford University.

[PRCV06] Pechoucek, M., Rehák, M., Charvát, P., Vlcek, and T., Kolar, M. (2006): Multi-Agent Planning in Mass-Oriented Production, accepted for publication In IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews.

[PSPU06] Pechoucek, M., Sislak, D., Pavlicek, D., and Uller, M. (2006b): Autonomous Agents for Air-Traffic Deconfliction. In: Proc. AAMAS-06 – Industry Track, ACM Press, pp. 1498-1505.

[ROJC09] Rogers, A., Jennings, N.R., Corkill, D.D. (2009) In T.R. Payne (Ed.) Agents. Agent Technologies for Sensor Networks. IEEE Intelligent Systems. March/April 2009. Online available: http://eprints.soton.ac.uk/267194/1/x2age.pdf

[RSSG10] V. Reddy, S. Shakkottai, A. Sprintson, and N. Gautam. Multipath wireless network coding: a population game perspective. In INFOCOM, 2010 Proceedings IEEE, pages 1–9. IEEE, 2010.

[SEFA07] Santa-Eulalia, L.A., Frayret, J.M., D’Amours S., 2007. Agent-based Simulation for Distributed Supply Chain Planning: Conceptual Modeling, Analysis, and Illustration. CIRRELT-2007-11.

[SHAP53] Shapley, L. 1953. A value for n-person game. pp. 307-317 in “Contributions to the Theory of Games”,Volume II, H.W., Kuhnand A.W.Tucker, (Eds.) Prince-ton University Press.

[SHAP53a] L. Shapley, 1953a. Stochastic games. In Proceedings of the National Acade-my of Sciences, Vol.39, 1095-1100.

[SOBE71] M.J. Sobel, 1971. Noncooperative stochastic games. Annals of Mathematical Statistics, Vol.42, 1930-1935.

[SRNC03] V. Srinivasan, P. Nuggehalli, C. Chiasserini, & R. Rao,“Cooperation in Wireless Ad Hoc Networks,” Proc. IEEE INFOCOM, vol. 2, pp. 808-817, 2003.

[STAC34] Stackelberg, H. von. 1934. Markform und Gleichgewicht. Vienna: Julius Springer.

[STRA98] Strader et.al., (1998). “Simulation of Order Fulfillment Indivergent Assemble Supply CHains”, In journal of Artificial Societies and Social Simulation.

[SWAM98] Swaminathan, J.M., Smith, S.F. and Sadeh, N.M. (1998). “Modeling Supply Chain Dynamics: A multiagent approach”, Decision Sciences, Vol. 29, No. 3, pp. 607-632.

[THUN05] J.-H. Thun, 2005. “The Potential of Cooperative Game Theory for Supply Chain Management”, In Research Methodologies in Supply Chain Manage-ment, In H. Kotzab, S. Seuring, M. Müller, G. Reiner (Eds.), pp. 477-491, Springer, 2005.

[WACK08] W. Wang, M. Chatterjee, & K. Kwiat, “Enforcing Cooperation in Ad Hoc Networks with Unreliable Channel,” Proc. 5th IEEE Int’ Conf. Mobile Ad-Hoc and Sensor Systems (MASS), pp. 456-462, 2008.

[WACK09] W. Wang, M. Chatterjee, and K. Kwiat, “Coexistence with malicious nodes: A game theoretic approach,” Proc. International Conference on Game Theory for Networks (GameNets ’09), 2009, pp. 277-286.

[WEBE10] R.H. Weber, 2010. Internet of Things: New security and privacy challenges. Computer Law & Security Review; 2010, 26: 23-30.

[WOOL02] M. Wooldridge. An Introduction to Multi-Agent Systems. Chichester, England: John Wiley & Sons; 2002.

[WUZM11] Wu ZQ, Zhou YW, Ma JF. A Security Transmission Model for Internet of Things. Chinese Journal of Computers; 2011, 34(8): 1351-1364.

[XUBC13] X. Xu,, N. Bessis, J. Cao, 2013. An Autonomic Agent Trust Model for IoT Systems. In Proceedings of the 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks. (EUSPN-2013). Online available. http://bit.ly/1gWOdLw

[ZBSA12] A. Zelenkauskaite, N. Bessis, S. Sotiriadis, E. Asimakopoulou. Interconnectedness of Complex systems of Internet of Things through Social Network Analysis for Disaster Management. In Proceedings of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2012); 2012, p. 503-508.

[ZHYH05] B.-L. Zhu, H.-B. Yu, X.-Y. Huang, 2005. Game Theory-based Study on Collaboration Planning Model for Supply Chain. In Proceedings of the 7th international conference on Electronic commerce ICEC '05, pp. 365-369, Xian, China. August 15-17, 2005.