Using Blockchains for Agent-based Auctions

Leo van Moergestel, Mathijs van Bremen, Bianca Krieger, Martijn van Dijk and Erik Puik
HU Utrecht University of Applied Sciences, Utrecht, The Netherlands

Keywords: Lifecycle Agent, Blockchain, Part Exchange Market, Auction, Sustainability.

Abstract: To extend the lifetime of products, an agent is connected to the product. This agent has several roles. It depends on the phase of the lifecycle what these roles will be. One of the roles in the usage or recycling phase is to negotiate for buying spare parts in case a part of the product is broken. The same agent can also decide to offer spare parts to other agents to reuse working parts of a broken product. To accomplish this idea, a marketplace for agents has to be set up, where the auctions can take place. To support this concept, blockchain technology has been used. Blockchains are a new type of technology, known from bitcoins, but there are other cases where blockchains can be used. Blockchain is known for its decentralisation, transparency and for making trustful transactions. In this paper the working of different types of blockchains will be briefly explained and determined if they can be useful for online auctions by agents. A prototype of the marketplace using blockchains has been built.

1 INTRODUCTION

Spoiling resources is considered a bad habit in many cultures. It is not only a bad habit, but in the case of scarce resources, one should definitely consider reuse. Products made should have the capability to extend the life of the product as well as to offer the possibility to reuse parts and subparts. This could be done autonomously or with the help of the user/owner of that specific product. In (van Moergestel et al., 2013), a system is proposed where a so called product agent or lifecycle agent is used to monitor the product during usage and to support repair by consulting other lifecycle agents if they have spare parts available at an affordable price. To make this system work in a reliable way, blockchain technology can be used. Blockchains can be used for many applications. In this paper, the scope for using blockchains is focused on auctions by agents. With the use of blockchains, transactions can be made without trusting a third party, such as a bank or central administrator.

The rest of this paper is organised as follows: at first an explanation of the concept of the lifecylcle agent is given in Section 2 followed by a very short introduction to blockchain technology in Section 3. In Section 4, the market approach for exchanging parts will be explained. In Section 5 the usage of blockchains is introduced, followed by Section 6 where the prototype of the agent-based marketplace with the results of some special cases is discussed. Section 7 discusses related work. Section 8 contains a discussion and a view on future work. A conclusion and bibliography will end the paper.

2 THE LIFECYCLE AGENT

Every product made has its lifecycle as depicted in Figure 1. A product is designed, manufactured and after manufacturing brought to the end-user. The product will then enter its usage phase and finally be recycled at the end of its lifecycle.

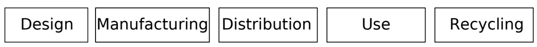

Figure 1: The lifecycle of a product.

The concept presented in (Moergestel et al., 2017) was that every product will start as a so called product agent that will guide the product during manufacturing. During manufacturing, information is collected by the product agent. Actually, most products or systems consist of a set of parts and subparts as depicted in Figure 2. Each subpart is first created by its own product agent and when the parts are combined, relevant data about the subparts is collected by the final product agent responsible for the final product. This relevant data might be a pointer to a product agent responsible for the subpart if such an agent exists. After manufacturing the product agent will become a lifecycle agent to support the product in other phases of its

Moergestel, L., Bremen, M., Krieger, B., Dijk, M. and Puik, E.
Using Blockchains for Agent-based Auctions.
DOI: 10.5220/0006587501920199
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 1*, pages 192-199
ISBN: 978-989-758-275-2

lifecycle, still having all the manufacturing data available. This could be done by embedding the agent in the product itself (Moergestel et al., 2017) or by putting the agent in the cloud. Even when embedded, it is desirable to put a copy of the agent in the cloud, just in case the original agent in the device becomes damaged. The agent in the cloud will be synchronised with the embedded agent. During all the phases of the lifecycle, the lifecycle agent will have its roles and responsibilities. One of the roles is monitoring the status and usage of parts and subparts. If one of the parts breaks or needs to be replaced the lifecycle agent will notice. When a part is broken or needs to be

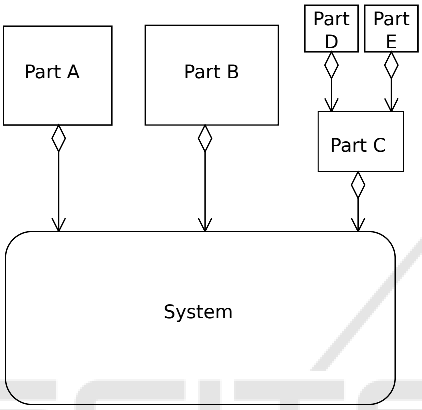

Figure 2: System/product consisting of subparts.

replaced, the lifecycle agent will enter a marketplace where other lifecycle agents exist that might offer the spare part needed. Important information about these spare parts is the usage so far resulting in the expected remaining lifetime and its economic value. This information should be reliable. When the product is at the end of its lifecyle the lifecycle agent can offer remaining still working parts on the same marketplace.

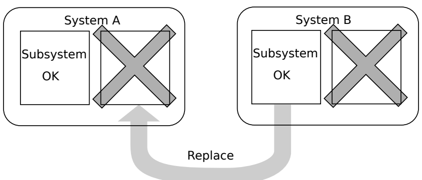

Figure 3: Two broken products exchanging a subpart resulting in repair of one product.

A simple example will show how useful this approach can be. Consider a product consisting of two similar parts. When one on these parts is broken, in most cases the whole product is considered to be broken. If we have two of these products a simple exchange of a working part from one system to the other system will result in one working system again. This is depicted in Figure 3.

3 A BRIEF EXPLANATION OF BLOCKCHAINS

A blockchain is a distributed database that maintains a continuously growing list of ordered records called blocks. The records can contain all sorts of data, for example transactions. When data in a block is recorded it cannot be modified. Before data can be recorded in a block the majority of participants has to approve the transaction (Nakamoto, 2008). Bitcoin is the first real-world application of blockchains. It requires no trust between participants. Each participant in the network has one or more wallets. A wallet consists of an asymmetric key pair. The public key also serves as an address to send coins to, comparable to an account number in the traditional banking system. When sending a payment to the network, the payment is signed with the private key of the sender. Then, the payment will end up in a pool of pending transactions.

3.1 Mining

Bitcoin defines a process for confirming transactions. This process is called mining. A computer that performs this process is called a miner.

Miners collect pending transactions from the network and put them into blocks. A block consists of:

- The hash of the previous block
- A number of transactions
- A cryptographic nonce

In Figure 4 a chain of two blocks is depicted.

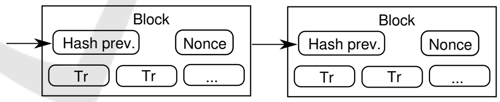

Figure 4: A blockchain of transactions (Tr).

For a block to be valid, the hash of the entire block must begin with a number of zeroes. This number will raise over time, increasing the required computational resources to mine blocks. Upon finding a block, the miner broadcasts it to other miners. The other miners accept the block when it only contains valid transactions.

3.2 Double Spending

Each transaction requires one or more so-called inputs. An input is an incoming transaction. Inputs must be spent entirely, but when the input is larger than the amount to spend, then they can be split. The remaining amount of the input is than sent back to the wallet of the sender. A bitcoin can be split in amounts of 0.00000001 BTC, called a Satoshi. Pending transactions that try to spend an input twice will be rejected by the miners in the network, thus preventing double spending.

4 MARKET FOR AGENTS

As already explained in Section 2 agents will exchange parts on a marketplace. This situation is depicted in Figure 5, where four agents exist. Two donors, offering parts and two acceptors asking for parts. One agent is asking for part *A* that is offered by one of the donors, while another agent is asking for part *E*. This part is not available and in such a case the agent has four possibilities:

1. it can wait for a donor to arrive.
2. it can decide to buy a brand new part. This might also be a possibility for the manufacturer to add a donor agent offering new parts at a price that is, as one might expect, higher than the price for the used parts.
3. it can decide to become a donor, offering its still working parts on the marketplace.
4. it can decide to become a donor, but meanwhile it will still check for the availability of the part it is searching for and if the part becomes available before it has sold any of its own working parts, it might decide to become an acceptor. More complicate scenarios are also possible if some of its working parts are already sold.

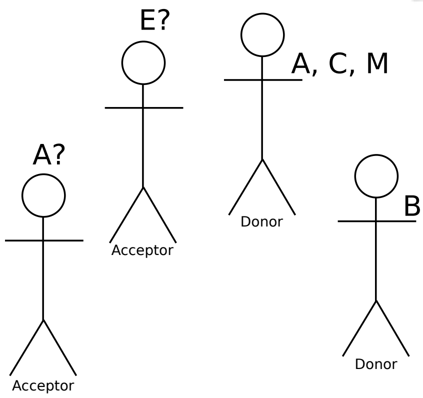

Figure 5: Market of agents.

Online auctions are susceptible to fraud. We discuss the most important possibilities here along with the solution offered by the concept presented in this paper.

- Because there is no easy way to prove the ownership of items, sellers could offer goods that they do not own. This form of fraud could be prevented by using a custom blockchain that supports coloured coins. A coloured coin is a software object that represents a product (or part of it) in the real world. When a user sells a product, the buyer gets the coloured coin. The seller can not offer the product a second time because he is not in possession of the coloured coin any more.
- On online marketplaces a lot of fake accounts are active or accounts representing (non-existing) stores. These fake-accounts can spoil the marketplace concept. By creating an auction that is only available for users who have purchased the product the number of fake accounts can be decreased significantly. Sellers (real or fake) not having a coloured coin mentioned before cannot participate in trading goods.
- The state of a product offered like usage or age, is not according to the real state. In our model the products will be equipped with a lifecycle agent, that will keep track of the usage of the parts. When the part is created, the coloured coin will contain a timestamp of the creating time. Cheating about the age is not possible. However the lifecycle agent should be trusted and there is still no guarantee that the usage reported is correct.

The context of this research is a multi-agent system for acquiring spare parts from broken products. In this system each product contains a lifecycle agent which has three tasks (Moergestel, 2014):

1. estimating the life expectancy of various parts of the product by monitoring the usage and age.
2. in case a part fails, deciding to sell the remaining parts, or acquire a replacement part.
3. selling the remaining parts, or acquiring new parts.

This paper focuses on the last task. We propose a system in which parts are represented by coins. These coins can carry metadata.

The prototype that was proposed in (Moergestel et al., 2017) is not based on blockchain technology and also has a key limitation. All agents register with an agent that acts as a kind of phonebook. If this agent becomes unavailable the entire system can no longer function. Blockchain technology solves this, by creating a decentralized knowledge base. One of the things that should also be implemented is a good way for the agents involved to communicate. In our prototype a blackboard approach has been used, where agents can post the parts they offer. An alternative to the blackboard could be the publish and subscribe pattern, where agents offering parts will publish it and agents that want to buy will be notified at the moment a part is available.

5 BLOCKCHAIN USAGE

In this paper we propose a system in which coins are used to complete transactions between agents. The coin represents a part of the device, and acts as a proof of ownership. For building blockchain-based applications, some factors have to be taken into consideration. Building a blockchain from scratch is a complex task. There are multiple platforms available for creating blockchain-based applications.

5.1 Considerations Regarding Multichain

Multichain is a GPL-licensed platform to build public and private blockchains. It allows a blockchain administrator to assign resources to wallets. There is no limit to the creation or availability of assets. Furthermore, administrators can manage permissions. Another key feature of MultiChain is the creation of datastreams on the blockchain. (Multichain, 2017)

5.2 Considerations Regarding Ethereum

Ethereum is a public blockchain aimed at facilitating smart contracts. It offers a turing-complete programming language for creating smart contracts. A smart contract contains conditions to be met before the contract is executed (ethtrade.org, 2016). One of the prime considerations is the cost of performing operations in the network. This price might rise unexpectedly. Ethereum has its own currency: ETH. This currency is used to pay for operations in the network. A lifecycle agent would need ETH to perform operations. The manufacturer would need to supply each agent with an amount of ETH. There is no guarantee that this amount will suffice when parts need to be replaced.

5.3 Choosing the Platform for the Prototype

For the first prototype the MultiChain platform has been chosen. The first consideration was the developer friendly nature of the MultiChain platform. Setting up a first blockchain for experimenting with the platform was done in a few minutes. Secondly, MultiChain can be configured to be a fully managed blockchain, a fully open blockchain, or something in between.

5.4 Public Blockchains

A public blockchain is a blockchain where every user can read and send transactions. There are no restrictions for users and the transactions are fully transparent. The public blockchains are considered to be fully decentralized. This means that every user has the same permissions and there is no hierarchy.

The advantage of a public blockchain is that no one has to maintain the network, it is transparent for all users and all users have got the same permissions.

The disadvantages of a public blockchain is that all the transactions are visible for all users. All transaction have to be stored somewhere. This means a lot of data storage will be used only to keep track of all previous transactions and a lot of storage will have to be available to store future transactions.

5.5 Private Blockchains

In contrast to public blockchains, a private blockchain can be managed. The access permissions are controlled, the right to modify or read a block is restricted per user. This means that a private blockchain will need to be maintained. The blockchain needs to be centralized to an organization.

The downside of using a private blockchain is that it has to be maintained and configured, creating a single point of failure. An administrator controlling the system has to grant permissions to nodes, the system will become dependable on the administrator. The administrator will represent in our case a company who wants to use the online auction for their products. When the company stops maintaining the blockchain, new users, depending on the configuration, will not be able to join the blockchain.

For creating new blocks on the blockchain miners are needed. In private blockchains there are less collaborators, and the question arises who will grant rewards for miners who have mined a new block. A solution can be to let a company handle the mining or the reward system. This is actually a strange situation because blockchains were invented as a distributed concept without a specific owner.

6 RESULTS

To demonstrate the concept of the marketplace, a prototype has been built. The prototype contains an admin-agent which creates the streams, divides the assets and checks if the lifecycle agent has got the required parts. When a lifecycle agent does not have any parts, the required part will be given by the admin-agent.

The lifecycle agent is based on a product containing three parts that will be monitored. If the products are working normally, the parts will decay over time and can break down. In case of a broken part the product will not be operational and the condition of the other parts will not further decrease until a replacement part is obtained.

When one of the parts gets broken, the lifecycle agent will search for a replacement of this part. If another agent offers the part we are looking for, the lifecycle agent will bid on it. In the prototype the first bidder is always the winner. When no other agents offer the part needed, the lifecycle agent will offer its other, still working, parts.

6.1 Functionality of the Admin Agent

The admin agent kickstarts the blockchain and assigns parts to lifecycle agents during the manufacturing phase. It actually assigns coloured coins, representing parts, to the lifecycle agent. How the agent manages to do this will be explained using the pseudocode below.

```
Create stream if not exists "metadata"
Create stream if not exists "bids"
Create stream if not exists "offers"
Create asset if not exists "part"
while(true) {
  addresses = get addresses - own address;
  for (address : addresses) {
    if (address has no transactions) {
       parts = get parts to add
       for (part : parts) {
          partId = generate random unique identifier
          Add partId, part to stream "metadata"
          Send partId to address
       }
    }
  }
}
```

As we can see above the admin has two states: initializing and running. During the initialization the admin makes sure the streams and the **part** asset are created. When it's in the running state (the while-loop) it will gather all the addresses of the lifecycle agents. If an address has never owned any parts the admin will create a number of the **part** assets. For each of these assets a new key-value pair will be added to the **metadata** stream. The key will be a unique identifier and the value will contain data regarding the part. When that is done the admin will send a **part** asset to the current address with the asset's identifier as data. The lifecycle agent then owns that part.

6.2 Functionality of the Lifecycle Agent

The lifecycle agent keeps track of the current state of each of its parts and handles communication with other agents in case a part breaks down. The inner workings will be explained using the following pseudocode:

```
partCount = -1
parts
while(true) {
  currentPartCount = get count of asset "part" we own
  if (currentPartCount != partCount) {
     parts = get parts we own
  }
  if (has broken parts) {
     if (has open offers) {
        if (has bids for open offers) {
           winner = first bidder
           close offer on stream "offers"
           send part to winner
        }
     }
     else {
        if (has matching offers for broken parts) {
           for (matchingOffer : matchingOffers){
            place bid for matchingOffer on steram "bids"
           }
        else {
        workingParts = get working parts
        for (workingPart : workingParts) {
           offer workingPart on stream "offers"
        }
     }
  }
  else {
     for (part : parts) {
        decay part
        update part on stream "metadata"
     }
  }
}
```

The lifecycle/product agent starts off by checking if it owns a different number of **part** assets than before. If it does, it will update its owned parts. After that it will check if any of its parts are broken. If none are broken, then the product is functional and only the metadata will be updated. In our concept, actually a simulation, the decay of each part is chosen randomly with a chance of instantly breaking down. The new decay value will then be published on the **metadata** stream. In case we do have a broken part, the agent needs to decide what to do next. First it will check if any of its open offers has any bids from other agents. If it does, the first bidder wins and the agent will send the **part** asset to the winner. The agent will also update the offer to let other agents know that the offer is no longer available. If the agent doesn't have any open offers up it will check if other agents are offering the parts we need. If they do, we place a bid for the specific part we need. If they don't, we simply publish offers for our working parts.

6.3 Using the System

Using the proof of concept talked about earlier we ran the program and observed what happened. The system consisted of an admin agent and five lifecycle/product agents, all on the same machine. Each product agent received three **part** assets. For the decay, which was updated on each loop, we chose a random value between 0 and 10 with a 5 percent chance of breaking down instantly. The decay value starts at 100 and 0 means that the part is broken. Now we will talk about two cases that have been studied during testing. The first case will be more detailed but about a smaller portion of the test. The second case will be more global and just looks at the test as a whole. Each lifecycle agent mentioned, will be identified by an upper-case character ranging from A to E.

6.3.1 Case 1

The first case started off with part 3 of agent A and part 1 of agent B breaking down around the same time. As there were no offers for part 3, agent A put up offers for its working parts. After that part 1 of agent C broke down as well. So at that point the same part 1 of both agent B and agent C was broken. This resulted in both agent B and agent C bidding on the same offer. As the bid done by agent B was the first one, agent B won the bidding. Agent A then sent the part to agent B and closed the offer. The last step of this case was that agent C no longer found any open offers for part 1. This resulted in agent C just putting up offers for its working parts.

As we can see in this case agent A and agent C ended up to be dysfunctional. Agent B, however, managed to get a replacement for its broken part. So instead of three dysfunctional products there were only two. This is exactly the design objective of the marketplace.

6.3.2 Case 2

For the second case we will just look at the donors and the winners. Donors are agents representing a product that is no longer functional. Winners are agents representing a product that, after being dysfunctional, is functional again.

In this small simulation, running over a longer time, several products became broken. One product became broken, could be repaired and broke again with again the possibility for repair. Yet another product could be repaired after one of its parts was broken.

The conclusion after running the simulation was that by trading the working parts to other agents we increased the lifespan of some products. Instead of having 5 broken products the system managed to fix two products, one of them even twice.

7 RELATED WORK

This section gives an overview of related work in the field of blockchain technology as well as the use of multiagent technology in related fields.

The number of publications about blockchain technology is huge, given the fact that is a rather new technology. In (Nakamoto, 2008) the concept is introduced with respect to bitcoin. Two interesting overview articles are given by (Zhao et al., 2016) and by (Yli-Huumo et al., 2016). In (Zhao et al., 2016) the focus is on business innovation while (Yli-Huumo et al., 2016) has the focus on the current research status of blockchain technology. A paper by Kroll (Kroll et al., 2013) has the focus on the security aspect of bitcoins. The concept of colored coins, as used in our approach is described by Rosenfeld (Rosenfeld, 2012).

Agent-based technology used for marketplaces is described by Bonabeau (Bonabeau, 2002). This is part of a study to simulate human systems. Among others, an infrastructure for an agent-based market has already been shown by Eriksson in (Eriksson et al., 1998). The concept of the lifecycle agents has first been introduced by (Moergestel et al., 2010) and in a more elaborated form in (Moergestel, 2014). The concept for monitoring by agents is also used by Burgess. In (Burgess et al., 2002) Cfengine is presented as a monitoring system for complex IT-infrastructures.

The concept of the lifecycle agent was the result of the research done on agent-based manufacturing. Two models described by (Bussmann et al., 2004) and (Moergestel, 2014) make use of an agent that supports the manufacturing process. In (Moergestel, 2014), this agent is called a product agent. After production this agent carries valuable information for the other parts of the lifecycle. This product agent transforms to the lifecycle agent, present in all phases to come.

By using this same agent that monitored the production as well as the distribution and usage phase agent again in the final phase of the lifecycle, component reuse and smart disassembly is supported. This is a very important aspect when it comes to recycling of rare or expensive building material. The status of the quality of used sub-parts is available from and presented by the lifecycle agent. The reliability of this information is based on blockchain technology.

In (Ashton, 2009) the concept of the 'Internet of Things' is explained by the first user of the term 'Internet of Things'. The main idea of this concept is that the content of Internet is not only built and used by humans and therefore largely depending on humans, but the content will also be built by things connected to the Internet that are programmed to do so. In our model, the driving force of IoT is agent technology, based on the concept of the lifecycle agent.

8 DISCUSSION AND FUTURE WORK

This section will summarise the strengths and weaknesses of the approach presented. It will also discuss future work and enhancements of the system.

8.1 Strengths of the Blockchain Approach

Generally speaking, the use of blockchain technology can give the following advantages:

- Blockchain technology does not rely upon a central authority, thus it does not have a single point of failure. Blockchain also is trust-less, meaning that two parties are able to make an exchange without the oversight or intermediation of a third party, strongly reducing or even eliminating counter-party risk. Furthermore, blockchains will reject invalid transactions. Because there is no central authority, transaction fees can be reduced.
- Additions to public blockchains are publicly viewable by all parties creating transparency, and all transactions are immutable, meaning they cannot be altered or deleted.
- Blockchain transactions can reduce transaction times to minutes and are processed 24/7.

Our prototype is based on a private chain, where all these advantages are not available.

8.2 Weaknesses of the Blockchain Approach

- Anonymity may be an issue with the current approach. Due to the anonymous nature of blockchain messages it is impossible to determine the identity of the other agent. Since there needs to be a bond of thrust between the agents (the product needs to be shipped when it is bought), this is hard to achieve when the agents themselves are anonymous.
- Mining a new block is a computationally complex task. This task must be performed by the agents, which are embedded systems. The processing power of the embedded systems may be insufficient to solve the computational puzzle in a feasible time. This may either cause the transactions to suffer from significant delays, or the embedded systems require more expensive hardware and in some cases possibly a higher power drain from the battery.

8.3 Discussion and Shortcomings of our Approach

In the current prototype there is no bidding system. The agent who bids first on a part will always win. There is also no current strategy of calculating a price the offered part is worth. This means that the lifecycle agent which offers the part has no reference for considering an acceptable bid.

When the broken part of a lifecycle agent is not offered by other agents, the lifecycle agent will offer its other, still working, parts. The lifecycle agent stops searching for the broken part. When another agent offers the part the lifecycle agent requires to repair himself the lifecycle agent will not detect this.

When all or most parts of a certain type break, a shortage occurs. This could drive the entire system to a halt. This could be addressed by introducing a warehouse-agent that represents the manufacturer. The warehouse-agent can offer parts for sale that are still in stock or in production.

An agent has no guarantee that a part that is bought will arrive. By introducing smart contracts, another application of blockchain technology, this problem could be solved. The contract then holds the money in escrow until the package carrier confirms delivery.

8.4 Future Work

The concept of the lifecycle agents open possibilities that still have to be implemented. A message could be sent to all agents in case a batch of parts needs to be recalled. Such a message would be broadcast to all agents. If an agent has a part that needs to be recalled it can take measures to get that part replaced. For the automotive industry a simple message on the car dashboard should suffice. In other situations the agent could contact the owner of a device to react. An important aspect in all situations is the possibility to replace parts easily. Nowadays, a lot of appliances exist where replacement is extremely difficult or can only be done by highly trained personal. Easy replacement of parts should be an important aspect of the design of an appliance.

In the future refurbish-agents could be introduced. Refurbish-agents would purchase broken parts, and offer refurbished parts on behalf of repair companies or the lifecycle agent marketplace. This could address shortage after the production of spare parts has stopped.

The collected knowledge of all lifecycle agents for a given product will result in valuable data on the reliability of parts. This information could be used to determine the average lifetime of a part resulting in a reasonable resale value of a used part.

The prototype marketplace should have a real auction possibility.

9 CONCLUSIONS

One of the primary issues regarding sustainability of our current society is the way broken appliances are discarded while most parts could be re-used. By creating a marketplace based on autonomous agents that have all the information about the parts and subparts of a certain product, the acquisition of spare parts will become less complicated. Thus, electronic waste is reduced, reducing environmental pollution.

Blockchains could be useful in multi-agent systems. In this use-case agents exchange information, and perform negotiations by using a blockchain. The main advantage of utilizing a blockchain instead of traditional FIPA messages over HTTP is that transaction history is recorded. Agents can then act based on this information. Secondly, the demand for parts is recorded using a blockchain, resulting in a knowledge base containing reliable data about the parts of a product.

REFERENCES

Ashton, K. (2009). That 'the internet of things' thing. *RFID Journal*, (22 july).

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287.

Burgess, M., Hagerud, H., Straumnes, S., and Reitan, T. (2002). Measuring system normality. *ACM Transactions on Computer Systems (TOCS) Volume 20 Issue 2*, pages 125–160.

Bussmann, S., Jennings, N., and Wooldridge, M. (2004). *Multiagent Systems for Manufacturing Control*. Springer-Verlag, Berlin Heidelberg.

Eriksson, J., Finne, N., and Janson, S. (1998). Sics marketspace: an agent-based market infrastructure. In *First International Workshop on Agent-Mediated Electronic Trading, AMET-98, Selected Papers*. Springer.

ethtrade.org (2016). Understanding ethereum. *https://ethtrade.org/Understanding_ethereum.pdf, 2017-06-22*.

Kroll, J., Davey, I. C., and Felten, E. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013.

Moergestel, L., Berg, M. v. d., Knol, M., Paauw, R. v. d., van Voorst, K., Puik, E., Telgen, D., and Meyer, J.-J. (2017). *Internet of Smart Things - A Study on Embedding Agents and Information in a Device*.

Moergestel, L. J. M. v. (2014). *Agent Technology in Agile Multiparallel Manufacturing and Product Support*. Utrecht University.

Moergestel, L. v., Meyer, J.-J., Puik, E., and Telgen, D. (2010). The role of agents in the lifecycle of a product. *CMD 2010 proceedings*, pages 28–32.

Multichain (2017). MultiChain Open source private blockchain platform. *http://www.multichain.com/, 2017-06-22*.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Rosenfeld, M. (2012). Overview of colored coins. *White paper, bitcoil. co. il*.

van Moergestel, L., Puik, E., Telgen, D., Folmer, H., Grünbauer, M., Proost, R., Veringa, H., and Meyer, J.-J. C. (2013). Monitoring agents in complex products-enhancing a discovery robot with an agent for monitoring, maintenance and disaster prevention. In *ICAART (2)*, pages 5–13.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., and Smolander, K. (2016). Where is current research on blockchain technology? a systematic review. *PloS one*, 11(10):e0163477.

Zhao, J., Fan, S., and Yan, J. (2016). Overview of business innovations and research opportunities in blockchain and introduction to the special issue. *Financial Innovation*, 2(1):28.