

Optimizing Service Level Agreements in Peer-to-Peer Supply Chain Model for Complex Projects Management

Florina Livia Covaci^(✉)

Business Informations Systems Department, “Babes-Bolyai” University,
Cluj-Napoca, Romania
florina.covaci@ubbcluj.ro

Abstract. The focus of this paper is to find appropriate approaches to facilitate end-to-end SLA (Service Level Agreements) in complex projects environments using Peer-to-Peer Supply Chain Model, to establish and enforce service levels between each pair of component consumer/provider, so that the overall project requirements can be achieved at the best utility value (SLA). The Supply Chain Formation problem is described in terms of a directed acyclic graph where the nodes are represented by the component suppliers/consumers. Intelligent agents send messages in the name of component suppliers/consumers on three constraints (scope, time, cost) giving raise to SLAs. The SLAs are expressed as utility functions and it is concluded that in complex projects scenario where the graph is always a tree the proposed model will converge to the optimal solution and the best utility value will be propagated autonomously across all component providers within the project environment.

Keywords: Peer-to-Peer Supply Chain Model · Project management Service Level Agreements

1 Introduction

In complex projects scenarios the project outcome may be the result of several deployed components/services, each of which autonomously manages diverse constraints which determine the quality criteria of the project outcome. In such a scenario where the component availability is dynamic, guaranteeing specific quality criteria of the final outcome is a real challenge. The quality of the final service/product delivered to the customer is strongly affected by those components/services employed to compose it. If just one of the composing services violates the quality criteria, the global quality delivered to the customer might get definitively compromised, to the detriment of the final customer. There is need for an automated mechanism that enables the negotiation of the project management Iron triangle [1] constraints for complex cooperative projects. Our present

work tackles the Supply Chain Formation (SCF) problem in the context of automated negotiation for complex projects environments, where a component can be both a service provider and a service consumer in different negotiation contexts at the same time. I am using the notion of Service Level Agreement (SLA) in order to define and ensure the quality criteria between the customers and the suppliers in the supply chain. A SLA represents a written agreement between the customer and supplier with emphasis on a mutually beneficial agreement and defines the key quality indicators of the service provided [2]. SLAs are fundamental for any kind of supply chain and assume a central position in popular IT service management standards such as ITIL (www.itil.co.uk). The Supply Chain Formation (SCF) problem has been widely studied by the multi-agent systems community. Numerous contributions can be found in the literature where participants are represented by computational agents [6–13]. These computational agents act in behalf of the participants during the SCF process. However, all the previous work regarding Supply Chain Formation (SCF) problem is dealing only with cost during SCF process. Our work proposes a multi-issue approach that introduces utility functions in order to model the interdependence between these issues. At each moment during the supply chain formation process, the agent wants maximizing its expected utility function. Each participating agent is sending messages under three constraints: scope, cost and time and is interested in obtaining those contract values that maximize their utility functions. Negotiation finishes with a contract (the SLA) that is composed of the actual quality criteria of the component provided by the supplier. In order to optimize the overall SLA in a complex project environment so that the overall project requirements can be achieved at the best utility value, we propose an automated supply chain formation model based on the RB-Loopy Belief Propagation message passing mechanism [3]. The present paper is structured as follows: Sect. 2 provides the background for SCF problem, Sect. 3 describes our proposed model for optimizing SLA in a P2P supply chain for a complex project scenario, Sect. 4 evaluates the proposed model and finally Sect. 5 provides conclusions and future work.

2 Background

The Supply Chain Formation (SCF) problem has been widely studied by the multi-agent systems community. Numerous contributions can be found in the literature where participants are represented by computational agents [6–13]. These computational agents act in behalf of the participants during the SCF process [14]. By employing computational agents it is possible to form SCs in a fraction of the time required by the manual approach [15].

The majority of the literature on SCF involve modeling the supply chain as a network of auctions, with first and second-price sealed bid auctions, double auctions and combinatorial auctions among the most frequently-used methods. SCF through auctions is a popular approach because auctions are frequently used in real-world tendering and sales situations, they are often able to form

good solutions to the SCF problem, and some auctions are able to guarantee various desirable game-theoretic properties.

Walsh and Wellman [9] proposed a market protocol with bidding restrictions referred to as simultaneous ascending (M+1)st price with simple bidding (SAMP-SB), which uses a series of simultaneous ascending double auctions. SAMP-SB was shown to be capable of producing highly-valued solutions which maximize the difference between the costs of participating producers and the values obtained by participating consumers over several network structures, although it frequently struggled on networks where competitive equilibrium did not exist. The authors also proposed a similar protocol, SAMP-SB-D, with the provision for decommitment in order to remedy the inefficiencies caused by solutions in which one or more producers acquire an incomplete set of complementary input goods and are unable to produce their output good, leading to negative utility. This use of a post allocation decommitment stage was recognized as an imperfect approach, however, due to the possible problems created by rendering the results of auctions as nonbinding.

Although auctions and negotiations are by far the most commonly-employed techniques in agent-based approaches to the SCF problem, there are some approaches in the past years that make use of graphical models and inference algorithms to tackle SCF and related problems. Next we will briefly review them.

2.1 Max-Sum

Max-sum [16] is a message passing algorithm that can find approximate solutions to optimization problems and it translates the problem into a factor graph. Max-sum has shown good empirical performance in a wide range of multi-agent systems coordination scenarios [17–21]. Max-sum provides an approximate solution for the problem of maximizing a function that decomposes additively in three steps. First, it maps the problem into a structure called local term graph. Then it iteratively changes messages between vertices of that graph. Finally, it determines the states of the variables.

Having $X = \{x_1, \dots, x_n\}$ be a sequence of variables, with each variable x_i taking states in a finite set D_i known as its domain. The joint domain D_X is the cartesian product of the domain of each variable. x_i refers to a possible state of x_i , that is $x_i \in D_i$. Moreover, X it is used to refer to a possible state for each variable in X , that is $X \in D_X$. Given a sequence of variables $X_f \subseteq X$, a local term f is a function

$$f : D_{X_f} \rightarrow R \quad (1)$$

X_f is the scope of f , and X_f is a possible state for each variable in X_f . Finally, a term whose scope is a single variable is said to be a simple term, and a term whose scope is two or more variables is said to be a composite term. A function

$$g : D_X \rightarrow R \quad (2)$$

is said to decompose additively if it can be broken as a sum of local terms. That is, whenever there is a set of local terms F (referred to as the additive decomposition F) such that $g(X) = \sum_{f \in F} f(X_f)$ the problem of maximizing a function that

decomposes additively can be expressed as follows: $g(X) = \sum_{f \in F} f(X_f)$ subject to: $x_i \in D_i, \forall i \in \{1, \dots, n\}$.

In order to maximize a function g that decomposes additively max-sum is mapping an additive decomposition F of g into a graph which is called the local term graph. The local term graph LTG is simply a specialization of the local domain graph defined in [22]. Each vertex of the LTG is, as its name suggests, a local term in an additive decomposition F of the objective function g . An edge between two terms in the LTG means that these two terms share one variable and that they are willing to exchange information about the variable they share. For each vertex $v \in \text{LTG}$, f_v represents its associated term. Vertices associated with simple terms are referred as simple vertices vertices associated to composite terms are referred as composite vertices. In general, the local term graph used by max-sum is built from F as follows. First, for each variable x_i , a simple vertex is created, associating with it a term (f_{x_i}) that is the addition of every simple term in F whose scope is x_i . Second, for each composite term f_j , a composite vertex is created. The simple vertex is labeled x_i associated to variable x_i and f_j the composite vertex associated to composite term f_j . Finally, each composite vertex is connected to the simple vertex of each of the variables in its scope. Notice that, since composite vertices are only connected to simple ones and vice versa, any pair of connected vertices in this graph share a single variable, the one corresponding to the simple vertex. The factor graph is a bipartite graph with two disjoint sets corresponding to the set of simple vertices and the set of composite vertices. After an additive decomposition of the function to maximize has been mapped into a local term graph, max-sum proceeds to the second step by iteratively exchanging messages over that local term graph. Each vertex of the local term graph is in charge of receiving messages from its neighbors, composing new messages and sending them to its neighbors. Therefore, there will be messages exchanged from simple vertices x_i to composite vertices f_j and vice versa. The message exchanged between a pair of vertices is a vector of real numbers, one for each possible state of the variable shared by both vertices. The exchange of messages continues until a convergence criterion is met. Initially, each vertex v will initialize the message from each of its neighbors w to zeros. After that, for each neighbor w , it will assess message μ_v^w send the message to the corresponding neighbor, and receive the message μ_w^v from vertex w . The procedure above, is repeated until convergence or a maximum number of iterations is reached. Max-sum is said to have converged after none of the messages change from one iteration to another [23]. A slightly less stringent criterion for convergence is to stop max-sum after the preferred states for the variables do not change from one iteration to another [17]. This second criterion is useful for instances in which the preferred state of the variables converges but the messages marginally change at each iteration. After the message exchange ends, the step three determines the states of the variables.

2.2 Loopy Belief Propagation (LBP)

LBP is the first peer to peer approach that has been used to solve the SCF problem in a decentralized manner [4, 12, 13, 24]. In [4], an LBP-based approach

was applied to the SCF problem, noting that the passing of messages in LBP is comparable to the placing of bids in standard auction-based approaches. The decentralized and distributed nature of LBP also allows for the avoidance of the scalability issues present in centralized approaches such as combinatorial auctions. LBP is a decentralized and distributed approximate inference scheme involving the application of Pearls belief propagation algorithm [5] to graphical models containing cycles. It uses iterative stages of message passing as a means for estimating the marginal probabilities of nodes being in given states: at each iteration, each node in the graph sends a message to each of its neighbors giving an estimation of the senders beliefs about the likelihoods of the recipient being in each of its possible states. Nodes then update their beliefs about their own states based upon the content of these messages, and the cycle of message passing and belief update continues until the beliefs of each node become stable. The work in [12] shows that the SCF problem can be cast as an optimization problem that can be efficiently approximated using max-sum algorithm [16] presented in the section above. Thus, the authors offer the means of converting a SCF problem into a local term graph, on which max-sum can operate. In LBP, the SCF problem is represented by a model in which each of the participants decisions is encoded in single variable. The states of each variable encode the individual decisions that the participant needs to make regarding her exchange relationships plus an inactive state. Moreover, the activation cost for a participant p is encoded by means of a simple term f_p , also called activation term. Each of these activation terms has the participants variable as its scope and takes value zero for the inactive state and the activation cost for any of the active states. In order to ensure that decisions are consistent among participants, in LBP, there is a compatibility term for each pair of variables representing potential partners. A compatibility term $f_{p_1 p_2}$ encodes the compatibility between the decisions of the two participants p_1 and p_2 . Two participants are in incompatible states whenever one of them is willing to trade with the other, but the other one does not. If two states are compatible, the value of the compatibility term is zero, otherwise is negative infinity. Hence LBP maps the SCF problem into a set of participant variables $X = \{x_1, \dots, x_N\}$, a set of activation terms $F_A = \{f_1, \dots, f_N\}$, one per variable, and a set F of compatibility terms. Then, solving the SCF problem amounts to finding a state assignment for the participant variables in X that maximizes the following reward function:

$$R_{LBP}(X) = \sum_{x_i \in X} f_i(x_i) + \sum_{f_{kl} \in F} f_{kl}(x_k, x_l) \quad (3)$$

in the equation above can be decomposed additively. Therefore, it can be mapped into a local term graph over which max-sum can operate in order to find a solution to the SCF problem.

2.3 Reduced Binarized Loopy Belief Propagation Algorithm (RB-LBP)

As LBP suffers from scalability issues in [3] the authors introduce the Reduced Binarized Loopy Belief Propagation algorithm (RB-LBP). RB-LBP is based on the max-sum algorithm and simplifies the calculation of max-sum messages through careful analysis of its local terms. The variables are binary which simplifies the supply chain formation process and each buy and sell decision is decoupled, encoded in a different variable, from the rest of buy and sell decisions. By decoupling these decisions the algorithm is able to reduce the number of combinations to take into account. Thus, the authors show in [3] that RB-LBP reduces the computation required by LBP, to assess SCs from exponential to quadratic, and the memory and communication requirements from exponential to linear.

The participants decisions and their costs for each participant p taking part in the SC is encoded in two kind of variables. On the one hand, an activation variable x_p that encodes whether participant p is active ($x_p = 1$) or inactive ($x_p = 0$), namely part of the SC conguration or not and the activation cost. Moreover, in order to introduce participants activation cost, the authors make use of activation terms. An activation term takes as parameter an activation variable and takes as a value the activation cost of the participant when the activation variable takes value one and zero otherwise. Formally, the equation for an activation term f_p for participant p can be expressed as:

$$f_p(X_p) = \begin{cases} C_p, & \text{if } x = 1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Furthermore each possible buyer p' of each of her input goods g , the authors in [3] create an option variable $s_{pgp'}$ that encodes whether p is selling good g to participant p' ($s_{pgp'} = 1$) or not ($s_{pgp'} = 0$). Similarly, for each possible seller p' of each ps input goods g , the authors create an option variable $b_{p'gp}$ that encodes whether p is buying good g from participant p' ($b_{p'gp} = 1$) or not ($b_{p'gp} = 0$). In order to guarantee that only one of the providers of a given good is selected, the authors make use of selection terms. Given a participant p offering good g , a selection term links the activation variable from the participant (namely x_p) with the different choices for that good (namely b_{*gp}), and enforces that one and only one option variable takes on value one if the activation variable is active and that all option variables take on value zero otherwise. Formally, the equation for a selection term f_S joining the activation variable x_p and option variables $o_1, \dots, o_n$ is expressed in [3] as:

$$f_S(x_p, o_1, \dots, o_n) = \begin{cases} 0, & \text{if } \sum_{i=1}^n o_i = x_p \\ -\infty, & \text{otherwise} \end{cases} \quad (5)$$

Furthermore in order to guarantee coherent decisions between participants, the equality terms are used. An equality term links buy and sell variables regarding

the same transaction and enforces that both variables take the same value. Formally, the equation for an equality term f_E joining variables b and s has been expressed in [3] as:

$$f_E(b, s) = \begin{cases} 0, & \text{if } b = s \\ -\infty, & \text{otherwise} \end{cases} \quad (6)$$

Contrarily to LBP, there is no need to keep a table for the local terms. It is possible to simply calculate the output of selection and equality terms using equations given above. The additive decomposition that RB-LBP uses to tackle the SCF problem following a P2P architecture can be expressed as:

$$\mathcal{R}_{RB-LBP}(X) = \sum_{x_p \in X_p} f_p(x_p) + \sum_{f_S \in F_S} f_S(X_{f_S}) + \sum_{f_E \in F_E} f_E(X_{f_E}) \quad (7)$$

where X_p is the set of participant variables, F_S is the set of selection terms, and F_E is the set of equality terms. Having this additive decomposition, it is possible to map the SCF into a binary local term graph by: (i) creating a simple vertex for each variable (that summarizes all the simple terms with that variable as their scope); (ii) creating a composite vertex for each composite term; and (iii) connecting with an edge each simple and composite vertex that share a variable.

3 Optimizing SLA in Complex Projects Management

In complex projects environments, a component can be both a service provider and a service consumer in different negotiation contexts at the same time (Fig. 1). Recursive constructs of project components as both a component supplier and a component consumer complicates the negotiation scenarios. In particular, a project component has to confirm that its own suppliers can support the SLA it negotiates with its consumer, before it commits to the SLA. Furthermore, it is assumed that a consumer and its providers have conflicting interests on SLA parameters; otherwise, both parties can simply reach an agreement by choosing their common optimum in their negotiation space. In short, automated negotiation within a complex project scenario using decentralized supply chain communication model involves a top-down mechanism to link project requirements to underlying components to conjointly guarantee end-to-end SLA of a project.

In the SCF process that this paper describes, the components of the system may be provided either by internal teams or external suppliers. The project managers for each component of the complex project environment are represented by computational agents. These participant agents act on behalf of the project managers during the project lifecycle and they have to drive the component development process of the complex system with multiple system components within dynamic constraints of scope, time and cost and meanwhile meeting the agreed SLA requirements. Each agent (SC-Supplier of a Component) that acts on behalf of the project manager of the component, negotiates only with a selected set of potential components providers, which match some predefined requirements of

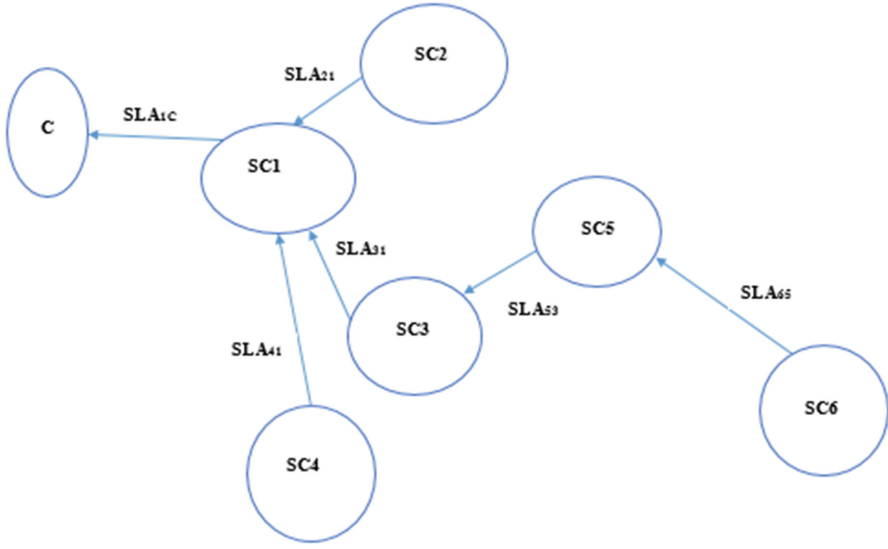


Fig. 1. Complex project example

the consumer. Hence, a project component can join a negotiation process as a potential component provider, if it fulfils the given requirements of a consumer.

All the previous research work that use computational agents to solve the SCF problem, make use of only of cost for pairwise agents and the optimization problem is treated as a profit maximization function. In comparison to all the other SCF research papers, the present approach is to translate the SCF problem as a negotiation game with non-transferable utilities. Our work is based on LBP approach as the passing of messages in LBP is comparable to the placing of bids. During the negotiation process between a service consumer and its provider, agents send messages regarding three issues: scope, time, cost have an exact way to estimate the utility they get, by making use of utility functions. By doing this, they can assess the benefits they would gain from a given SLA, and compare them with their own expectations in order to make decisions.

Below I provide a formal description of the supply chain formation problem in terms of a directed, acyclic graph (SC, E) where $SC = \{sc1, sc2, \dots, scn\}$ denote set of component suppliers represented by agents and a set of edges E connecting agents that might buy or sell from another. Let $X = \{x1, x2, \dots, xn\}$ denote the set of all system components. Each component has associated three constraints: scope, cost and time which participating agents are interested in negotiating. Negotiation finishes with a contract (the SLA) that is composed of the actual quality criteria of the component provided by the supplier. Notation $v_{x,i}^{sc,u}$ represents the expectation of user u on constraint i of component x supplied by component supplier sc , which in fact, are the quality values reflected in SLA. We denote by $U(v) = U_x^{sc,u}(v)$ the utility that the user u gets by

obtaining the actual value $v = (v_{x,scope}^{sc,u}, v_{x, cost}^{sc,u}, v_{x,time}^{sc,u})$ of the component that he gets. Similarity $W(v) = W_x^{sc,u}(v)$ represents that utility that the user sc gets by delivering the actual value v . When a provider (seller) negotiates with a consumer (buyer), both parties are interested in obtaining those contract values $v = (v_{x,scope}^{sc,u}, v_{x, cost}^{sc,u}, v_{x,time}^{sc,u})$ that maximize their utility functions $U(v)$ and $W(v)$ respectively. This means that at each moment during the negotiation, the agent sends a messages to its neighbours regarding the states of his variables, that is maximizing its expected utility function. The utility functions $U(v)$ will be calculated by means of weighted sum as follows:

$$U(v) = \sum_{k=1}^n w_k^{sc} * U_k(v_k), \text{ with } \sum_{k=1}^n w_k^{sc} = 1 \quad (8)$$

where $U_k(v_k)$ represents the utility that the consumer obtains by receiving the value v_k for the issue i_k and $0 \leq w_k^{sc} \leq 1$ represent the weights measuring the importance of a given issue k for a certain agent in the chain.

In order to optimize the overall SLA in a complex project environment so that the overall project requirements can be achieved at the best utility value, we propose an automated supply chain formation model based on the Loopy Belief Propagation message passing mechanism [13]. I will express the activation function in the activation term equation of the RB-LBP [3], as a SLA, which in our case is the vector $v = (v_{x,scope}^{sc,u}, v_{x, cost}^{sc,u}, v_{x,time}^{sc,u})$, so the equation for an activation term f_{SCi} for the project component SCi , $i = 1, \dots, n$, can be expressed as:

$$f_{SCi}(x_{SCi}) = \begin{cases} U_x^{sc,u}(v), & \text{if } x_{sc} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Hence the SLA optimization problem can be expressed as maximizing the following reward function:

$$\mathfrak{R}_{SLA-RB-LBP}(X) = \sum_{x_{SCi} \in X_{SCi}} f_{SCi}(x_{SCi}) + \sum_{f_S \in F_S} f_S(X_{f_S}) + \sum_{f_E \in F_E} f_E(X_{f_E}) \quad (10)$$

4 Evaluation and Simulation

As the proposed approach for message passing mechanism in the supply chain formation process is comparable to sending bids in auction-based approaches, in this section, I will analyze the game-theoretic properties of the proposed SLA-RB-LBP approach and further I will provide the results of the simulation for the proposed implemented model in PeerSim [28].

Hence regarding the game-theoretic properties of the SLA-RB-LBP approach, individual rationality is guaranteed in the proposed approach. Previous work in [9, 13] use a post allocation decommitment stage to eliminate unfeasible allocations. Using the selections terms in the proposed approach ensures

coherent decisions between participants and guarantees individual rationality. My approach is not incentive compatible because the dominant strategy for supply chain formation participants is not to truthfully reveal their private valuations, hence they do not know each others utility functions, agents being situated under an uncertain environment. This is an issue for real-life scenario. Our approach involves no payments either to or from the mechanism, and is therefore strongly budget balanced. In a complex project environment the binary local term graph is always a tree. In [25] the authors demonstrate that when the local term graph is a tree, max-sum is guaranteed to converge to the optimal configuration [25] and, it is known to provide neighborhood maximum configurations [26, 27]. We can conclude that the proposed approach ensures the allocative efficiency property and will always converge to optimal configuration so that the overall SLA can be achieved at the best utility value.

In order to validate the model proposed in Sect. 3, I have selected PeerSim for implementing the Supply Chain SLA protocol. The reasons for making this choice are the PeerSim performance regarding scalability and because it is based on components that allows prototype a new protocol, combining different plug-gable building blocks that are in fact Java objects. PeerSim is a single-threaded peer-to-peer simulator that is developed in a modular and scalable way [28]. The steps of the simulation process in PeerSim are [28]: 1. set the number of nodes in the configuration file 2. select one or more existing protocols (or implement a new one) to experiment with and initialize them 3. select one or more Control objects to monitor the desired properties (e.g. the values of the nodes) 4. run the simulation invoking the Simulator class with the configuration file, that contains the above information.

The simulation starts by reading the configuration file, given as an input parameter that defines the protocols to experiment. Then, both nodes and protocols are created and initialized. After the initialization phase, by default, every instance of the protocols running on each node is executed once per simulation cycle. The event-based simulation in PeerSim maintains the concept of simulation cycle, but in each cycle are executed only the protocols instances that have some pending events. Events (or messages) are sent by controllers or protocols and they are managed with a handler associated to the protocol that allows to handle a message properly. Messages exchange is simulated by a transport protocol that provides the communication primitives and allows you to add more realism to the simulation. The nodes are scheduled according to the list of pending messages. PeerSim models the set of nodes as a collection through the class `peersim.core.Network`. The overlay network is represented by a adjacency list where every peer n of the network is connected to a set of neighbors $N(n)$.

I have implemented in the simulator the SupplyChainSLA-LBP protocol based on the event driven model in PeerSim. The implemented `processEvent` method in Listing 1, using the message passing mechanism with two types of messages (`RequestMessage` and `ResponseMessage`) along with order of passing mechanism starting from the leaf nodes ensures that the selection of the partners is according to the selection term equation in (10).

Listing 1. Nodes Sending Messages and Updating Values

```

public void processEvent(Node node,int pid,Object message)
{
double nutility;

if ( message instanceof RequestMessage )
{

    RequestMessage reqMessage=(RequestMessage) message ;
    Transport t= (Transport)node.getProtocol(transportId);
    ResponseMessage respMessage;
    respMessage=new ResponseMessage(t_value ,c_value ,node);
    t.send(node , reqMessage.src ,respMessage ,pid);

}

else if ( message instanceof ResponseMessage )
{

    ResponseMessage respMessage=(ResponseMessage ) message ;
    nutility=getUtilityValue(respMessage.t_value ,
    respMessage.c_value);
    if ( nutility>maxutility)
    {
        t_value=respMessage.t_value;
        c_value=respMessage.c_value;
        maxutility=nutility;

    }
}
}

```

The scenario that the present simulation addresses is that each node in the supply chain has multiple options of partner agents to trade with. The number of these options are configurable in the configuration file that is send as a parameter when the simulation starts. All of this partner agents are preselected on the basis that they have the capability to provide the components, more specifically they are able to provide the system components as its own scope it is defined. Even if the issue Scope it is part of the weighted utility function and has a certain weight for each agent, the agents don't negotiate on this issue because they either provide the scope or the component or they are not being preselected a potential partner, this being accordingly with the equality term in Eq. (10).

Furthermore, each node v has two numeric values “t value” and “c value” that are the preferred states of the two issues: Time and Cost. Also each node has a utility function which is computed as a weighted sum from each of the

issues that the agents in the supply chain are discussing on. At the beginning of the simulation all the agents in the network are being initialized with random preferred values for time and cost variables and also for the weights used at computing every agent utility. Each node that is not a leaf is receiving messages from its neighbors, composing new messages and sending them upward to its neighbors. The message exchange between a pair of nodes is a vector of real numbers, one for each possible state of the variable shared by both nodes. Each node will assess messages received from the corresponding neighbor according to his own utility function and will chose among all the possible partners the one that maximizes his utility function. The procedure above, is repeated until after none of the messages change from one iteration to another or a maximum number of iterations is reached (Table 1).

Invoking the simulator with the configuration file above, the maximum SLA expresses as a utility function, that can be obtained at the root node, giving the constraints of time and costs in the underlying network, is 33.81 and the states of the variable shared by agents became stable at the following final utility values:

Table 1. Utility obtained by simulation

Node	Initial utility	Intermediate utility values	Final utility
0	6.325188371300536	22.46334767085241	33.816967145516315
1	2.4806385045290917	3.7658475023426337	5.9102205089020075
2	13.139763118138983	19.05462493705339	19.05462493705339
3	28.712757912880438	31.89339121189109	34.07402451090173
4	2.665934636137622	6.274423250339464	6.274423250339464
5	12.56190696603451	7.737144179620706	7.737144179620706
6	5.725432091050816	5.725432091050816	5.725432091050816
7	1.3976917977590864	1.3976917977590864	1.3976917977590864
8	27.9575241095819	27.9575241095819	27.9575241095819
9	55.39176948841717	55.39176948841717	55.39176948841717
10	36.66119166541283	36.66119166541283	36.66119166541283
11	14.788420689297935	14.788420689297935	14.788420689297935
12	29.54023493801846	29.54023493801846	29.54023493801846
13	12.816045296967191	12.816045296967191	12.816045296967191
14	3.746115621145735	3.746115621145735	3.746115621145735
15	24.741574241434865	24.741574241434865	24.741574241434865
16	3.3828050529549234	3.3828050529549234	3.3828050529549234
17	5.8003154798198695	5.8003154798198695	5.8003154798198695

My simulation was run having the following values in the configuration file:

Listing 2. Configuration file

```
# number of network nodes
SIZE 18
# number of simulation cycles
CYCLES 25
network.size SIZE
# Initializers to use
include.init initED netInit
# Controls to use
include.control neighObs valueObsED
#Supply chain SLA negotiation protocol
protocol.supplychainED SupplyChainSLA-LBP
# Network generation
init.netInit WireRegRootedTree
# The linkable protocol to operate on
init.netInit.protocol lnk
# Number of outgoing edges to generate from each node
init.netInit.k 3
```

5 Conclusions and Future Work

As today's market is in constant change, software components suppliers are faced with ever-changing customer needs and resources costs and availability. Consequently in complex projects environments it is no longer possible to maintain SCs over extended periods of time. Thus, the ability to quickly form effective, mutually beneficial trading partnerships becomes increasingly important. The old approach for SCF, where SCs were assessed manually after extended negotiations is no longer viable. Furthermore, human irrationality coupled with the complexity of the problem often lead to inefficient SCs. Therefore, there is a need for agile and flexible methods that support temporal collaboration between software components providers and consumers.

In this paper we have proposed a decentralized and distributed approach for optimizing SLA in complex project scenarios SupplyChainSLA-LBP protocol, based on the RB-LBP P2P supply chain model approach for sending messages between nodes and we have translated the activation cost into a SLA vector using utility functions. The current approach that this paper presents uses message passing between agents during the supply chain formation process and is closer to real life scenarios than the previous approaches that were using only cost as a mean for pairwise agents because uses utility functions for agents to make decision. As a future work I will investigate the impact on the supply chain formation when agents make coalitions. Since components in a complex project environment can be provided either by internal teams represented by cooperative agents that are willing to contribute to global business objectives and other

components might be provided by external organizations represented by self-interested agents which are less cooperative and hence more utility oriented we propose to investigate how the Shapely value affects the supply chain formation when there is an internal or external coalition formation.

References

1. Barnes, M.: Time and money in contract control (1969)
2. ITIL Foundation Handbook, London, The Stationery Office (2012)
3. Penya-Alba, T., Vinyals, M., Cerquides, J., Rodriguez-Aguilar, J.A.: A scalable message-passing algorithm for supply chain formation. In: 26th Conference on Artificial Intelligence (AAAI 2012) (2012)
4. Winsper, M., Chli, M.: Decentralized supply chain formation using max-sum looppy belief propagation. *Comput. Intell.* **29**, 280–309 (2012)
5. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, vol. 1, 1st edn. Morgan Kaufmann, San Francisco (1988)
6. Collins, J., Ketter, W., Gini, M., Mobasher, B.: A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int. J. Electron. Commer.* **7**, 35–58 (2002)
7. Davis, R., Smith, R.G.: Negotiation as a metaphor for distributed problem solving. *Artif. Intell.* **20**(1), 63–109 (1983)
8. Walsh, W.E., Wellman, M.P., Ygge, F.: Combinatorial auctions for supply chain formation. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 260–269 (2000)
9. Walsh, W.E., Wellman, M.P.: Decentralized supply chain formation: a market protocol and competitive equilibrium analysis. *J. Artif. Intell. Res. (JAIR)* **19**, 513–567 (2003)
10. Cerquides, J., Endriss, U., Giovannucci, A., Rodriguez-Aguilar, J.A.: Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In: IJCAI, pp. 1221–1226. Morgan Kaufmann Publishers Inc. (2003)
11. Giovannucci, A., Vinyals, M., Rodriguez-Aguilar, J.A., Cerquides, J.: Computationally-efficient winner determination for mixed multi-unit combinatorial auctions. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 1071–1078 (2008)
12. Winsper, M., Chli, M.: Decentralised supply chain formation: a belief propagation-based approach. In: Agent-Mediated Electronic Commerce (2010)
13. Winsper, M., Chli, M.: Decentralized supply chain formation using max-sum looppy belief propagation. *Comput. Intell.* **29**(2), 281–309 (2013)
14. Norman, T.J., Preece, A., Chalmers, S., Jennings, N.R., Luck, M., Dang, V.D., Nguyen, T.D., Deora, V., Shao, J., Gray, W.A., et al.: Agent-based formation of virtual organisations. *Knowl. Based Syst.* **17**(2), 103–111 (2004)
15. Winsper, M.: Using min-sum looppy belief propagation for decentralised supply chain formation, Ph.D. thesis, Aston University (2012)
16. Bishop, C.M., et al.: Pattern Recognition and Machine Learning. Springer, New York (2006)
17. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.R.: Decentralised coordination of low-power embedded devices using the max-sum algorithm. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 639–646 (2012)

18. Rogers, A., Farinelli, A., Stranders, R., Jennings, N.R.: Bounded approximate decentralised coordination via the max-sum algorithm. *Artif. Intell.* **175**(2), 730–759 (2011)
19. Kim, Y., Krainin, M., Lesser, V.: Application of max-sum algorithm to radar coordination and scheduling. In: *Workshop on Distributed Constraint Reasoning* (2010)
20. Pujol-Gonzalez, M., Cerquides, J., Meseguer, P., Rodríguez-Aguilar, J.A., Tambe, M.: Engineering the decentralized coordination of UAVs with limited communication range. In: Bielza, C., Salmerón, A., Alonso-Betanzos, A., Hidalgo, J.I., Martínez, L., Troncoso, A., Corchado, E., Corchado, J.M. (eds.) *CAEPIA 2013. LNCS (LNAI)*, vol. 8109, pp. 199–208. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40643-0_21
21. Stranders, R.: Coordinating teams of mobile sensors for monitoring environmental phenomena (2009)
22. McEliece, S.M.: The generalized distributive law. *IEEE Trans. Inf. Theor.* **46**(2), 325–343 (2000)
23. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge (2009)
24. Winsper, M., Chli, M.: Using the max-sum algorithm for supply chain formation in dynamic multi-unit environments. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 3, pp. 1285–1286 (2012)
25. Weiss, Y.: Correctness of local probability propagation in graphical models with loops. *Neural Comput.* **12**(1), 1–41 (2000)
26. Weiss, Y., Freeman, W.T.: On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. Inf. Theor.* **47**(2), 736–744 (2001)
27. Vinyals, M., Cerquides, J., Farinelli, A., Rodríguez-Aguilar, J.A.: Worst-case bounds on the quality of max-product fixed-points. In: *NIPS*, pp. 2325–2333 (2010)
28. Montresor, A., Jelasity, M.: PeerSim: a scalable P2P simulator. In: *Proceedings of the 9th International Conference on Peer-to-Peer*, pp. 99–100, Seattle, WA (2009)

Informatics in Economy

15th International Conference, IE 2016, Cluj-Napoca,

Romania, June 2-3, 2016, Revised Selected Papers

Silaghi, G.; Buchmann, R.A.; Boja, C. (Eds.)

2018, XIV, 201 p. 55 illus., Softcover

ISBN: 978-3-319-73458-3