Changes in a Service Oriented Enterprise: A Game Theory Approach

Ali Habibi Badrabadi & Mohammad Jafar Tarokh*

Ali Habibi Badrabadi, is a postgraduate student at the IT group, Department of Industrial engineering, K.N. Toosi University of Technology, Tehran, Iran.
Mohammad Jafar Tarokh, is a PhD, Associate Professor at the same Department.

KEYWORDS	ABSTRACT
Service Oriented Enterprise, Web Service, Service Culture, Top-Down Change, Economic Analysis, and Game Theory	*Service Oriented Enterprises (SOEs) are subject to constant change and variation. In this paper, the changes are considered from an economic perspective based on service culture notion. Once a change is implemented, the costs of some member services may increase, whereas the costs of some other services may reduce. We construct a game theoretic model trying to capture the possible conflicting interests of different parties in a SOE. Three incentive mechanisms are applied to the model. The first incentive mechanism shares the utility equally among the services involved in the change; the second utility-sharing rule is based on the Nash's bargaining solution, which accommodates the possible biased interdependencies inside the network; and the third rule, based on the Harsanyi's modified Shapley value, takes into account the possible coalition formation among the network parties. Since the three rules are analytically solvable, the principles of utility sharing can be implemented, for instance, as ex-ante contracts.*

1. Introduction

The recent convergence of information and communication technology (ICT) design, execution, storage, transmission and reusable knowledge is creating new opportunities. They include redeploying people, reconfiguring organizations, sharing information (e.g., language, processes, metrics, prices, policies and laws), and investing in technologies. The investments are intended to yield technical solutions that adjust to a changing business environment, and effectively leverage the value of knowledge in service relationships that produce high business value [10]. These are what we call services and service-oriented thinking.

The Web has grown from a mere repository of information to a platform for service provision. Web services are gradually taking root following the convergence of business and government efforts for making the Web primary medium of inter-actions [34]. Web services are the evolution of the RPC, DCE, DCOM, CORBA, RMI, … standards of the 1990s. The main innovation is an XML base that facilitates interoperability among implementations [11]. The maturity of XML based Web service standards, such as SOAP, UDDI, and WSDL, are driving the rapid adoption of Web services [9, 5].

This trend is motivating a paradigm shift in enterprise structure from the traditional single entity to a collaboration of Web services. Such service oriented enterprises (SOE) would potentially open the door of entrepreneurship to all Web users. A SOE is a temporary and dynamic collaboration between autonomous Web services that collectively provide a value added service to users. These services are typically provided in a frequently changing environment. Service oriented enterprises are also referred to as adaptive, on demand, virtual, extended,

---

*Corresponding author. **Mohammad Jafar Tarokh**

Email: alihabibi@sina.kntu.ac.ir   mjtarokh@kntu.ac.ir
Paper first received July. 10. 2009 ,and in revised form Jan. 13, 2010.

market-driven, or Web service based enterprises. We use the terms service oriented enterprise, SOE, and enterprise interchangeably to refer to these types of enterprises.

SOEs outsource their functionalities via third-party Web services. This triggers a need for a systematic approach to manage and maintain the proper functioning and cooperation of these services. This is of significant importance and very difficult because a SOE has to perform its functions in an extremely dynamic environment (i.e., on the Web). Market requirements and business regulations may change and individual services may come and go at will. In SOEs, changes are the rule, and are not the exception, as it is the case in traditional enterprises [19]. Therefore, providing a framework for change management in SOEs is important. There are two types of changes that happen to a SOE: top-down changes and bottom-up changes [1, 2, 19]. Top-down changes refer to the changes that are initiated by SOEs' owners. Bottom-up changes refer to the changes that are initiated by the outsourced Web service providers. A SOE may frequently make top-down changes to improve business processes, enhance market competitiveness, and comply with new regulations. In this paper, we focus on top-down changes that are always triggered by either new business strategies or new regulations, from an economic point of view.

When a top-down change is implemented, the costs of some member services increase, whereas the costs of some other services reduce. The first criterion for the change is that the total amount of profits increases. The other criterion is that each of the services is better off after the change has been carried out. That is, the services whose costs increase need a compensation payment in order to accept the implementation of the change. By our model, we show that these criteria can be satisfied and that joint gains are achievable through change.

We consider three incentive mechanisms, the first incentive mechanism introduced shares the utility equally among the companies involved in the change. The second utility-sharing rule is based on the Nash's bargaining solution, and the third rule, based on the Harsanyi's modified Shapley value, takes into account the possible coalition formation among the network parties.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to SOEs. Section 3 has an overview on changes in SOEs. Section 4 discusses top-down changes and describes the problem using an example, and Section 5 presents a game theoretic model to deal with top-down changes. In this section three different ways to share the surplus utility gained through a change are proposed and then, using a numerical example, results of them are compared. Finally, Section 6 concludes the paper and outlines future work.

2. Service Oriented Enterprise (SOE)

A service oriented enterprise is an extended, virtual, real-time, and resilient enterprise. The essential characteristics of an extended enterprise are its involvement and ability to realize straight-through processing of a number of organizations to deliver goods and services to customers. Extended enterprise is about connectivity between various service providers and service requestors. Therefore, a service oriented enterprise achieves the delivery of the supply, or value, chain. As mentioned throughout the literature [15, 20, 34], service orientation deals with loose coupling. An essential feature of loose coupling is the idea that services can be developed independently and then integrated with minimum or no dependency of the bindings between various platforms that support the services. Therefore, a service oriented enterprise facilitates the integration of loosely coupled services yet at the same time appears aggregated as a functional whole. With aggregation, various applications, repositories, and even roles or organizations appear to be well integrated, providing an essential service. For instance, a production or development effort could involve many applications and different groups from potentially geographically distributed organizations. The applications need to be invoked in a particular sequence or process flow. The output of one application, such as the blueprint of a product component, needs to be the input of another application, such as an automated manufacturing plant. The data type exchanges between the various applications need to be consistent. Similarly, the different groups involved in the ultimate objective need to be part of the same production, testing, certification, and manufacturing calendar [15].

An examination of a service oriented enterprise reveals three fundamental layers (Figure1). At the foundation is the service oriented architecture (SOA) components, including the infrastructure guaranteeing service, quality of service (QoS) as well as the enterprise service bus (ESB) for intra- and inter-enterprise connectivity. The ESB provides connectivity between various systems and trading partners using standard integration interfaces, especially Web services.

At the top is enterprise performance management (EPM). Here is where the overall performance of the organization, service contracts, trading partners, and organizational interactions are dealt with. This is where different departments within an organization—and in fact different organizations—are brought together to realize business goals. For instance, parts manufacturers and assemblers can participate in a value chain involving many companies. Each on the chain adds value toward the ultimate product. Organizationally each department internally is a service department at its core—offering services to various functions in the organization. Products are services offered to customers, trading partners, or

distributors. In fact, the service oriented enterprise assembles services and publishes them as composite applications.
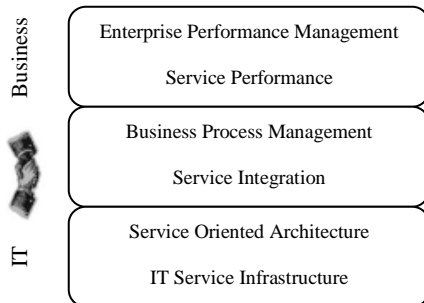
Business

IT

```
┌─────────────────────────────────────┐
│   Enterprise Performance Management  │
│                                      │
│        Service Performance           │
├─────────────────────────────────────┤
│    Business Process Management       │
│                                      │
│        Service Integration           │
├─────────────────────────────────────┤
│   Service Oriented Architecture      │
│                                      │
│      IT Service Infrastructure       │
└─────────────────────────────────────┘
```

Fig. 1. Service Oriented Enterprise

Business process management (BPM) is the middle layer of the SOE architecture. It brings business and IT together. What is a business process? A BPM system models and executes the interactions among human participants, systems, and trading partners. Business process is not just about flowcharts and pretty workflow diagrams. These are important, but not sufficient. The flowchart, or the flow, is only one aspect. Traditional workflow does a good job at guiding the work through the process but is often weak at integrating enterprise business policies, or rules, to interact with existing services, to complete work more efficiently, to reduce manual processing, and to manage enterprise as well as business-to-business (B2B) integration guided by business agreements. Agile enterprises are those that let the system manage the policy governance. Business rules engines (BREs) are becoming an essential component in the overall architecture solution stack of extended enterprises. However, these rule engines need to be part of the same system that also manages the processes. In other words, to achieve the desired agility, businesses need a unified BPM system that supports robust BPM and BRE functionality in the context of a single object model and a single system Procedural (i.e., process flows) and declarative (i.e., business rules) programming should be used in tandem, with the rules driving the processes and executed within the same system to realize the best platform for agility. The infrastructure layer supporting the business processes and carrying out all the connectivity, transformation, and QoS functions for service exchanges is the SOA layer, with ESBs as the key enabler for service architectures [8].

With this robust three layered architecture, service orientation provides the ability to loosely couple applications, trading partners, and organizations and to connect them via service calls. The coupling is often achieved through discovery. Furthermore, independent services can be composed in processes to provide even greater value than the sum of component services.

Service orientation enables internal as well as external trading partners to participate in distributed applications. Each party complies with agreed-on protocols and carries out its part in the overall execution of processes involving services from diverse organizations. The processes here are microflows typically involving only system or trading-partner service accesses. BPMS processes use the standards-based ESB transformation primitives as well as these micro-integration flows to create comprehensive business processes involving both human as well as system (i.e., back-end applications or trading partner) services. The enterprise implements its horizontal and vertical applications primarily as BPM applications. Business performance management is then enacted to make sure various business goals and service metrics are continually measured and monitored [15].

3. SOEs and Changes

Service enterprises are dynamic: partners could change; market conditions could change; new technologies could emerge in real-time. The service enterprise should take change into consideration in all its endeavors: in its organizational infrastructure and in the service oriented technologies it uses to realize its business goals. For instance, the objective could be a financial transaction involving financial institutions, custodians, brokers, contractors, legal entities, and clearing. The particular selection of a financial institution that provides a product or a service or the selection of the service could be dynamic. It could depend on price, availability, or benefits. Thus, financial processes such as purchasing securities could involve different organizations depending on the parameters or requirements of the transaction. Interfaces could also change. For instance, if a particular eXtensible Markup Language (XML) vocabulary is used for the process, the vocabulary could undergo iterations and changes, such as various versions. Exchange choreographies could also change. The only constant is change [15]. The agility required to adapt to these changes dynamically is part of the very nature of the service oriented enterprise.

Service technologies automate business processes and change as those processes respond to changing consumer, competitive, and regulatory demands. Services are thus subject to constant adaptation and variation adding new business rules and regulations, types of business-related events, operations and so forth. Services can evolve typically by accommodating a multitude of changes along the following functional trajectories [28]:

1. Structural changes: These focus on changes that occur on the service types, messages, interfaces and operations.

2. Business protocol changes: Business protocols specify the external messaging behavior of services (i.e. the rules that govern the service interaction

between service providers and clients) and, in particular, the conversations in which the services can participate in. Business protocols achieve this by describing the structure and the ordering (time sequences) of the messages that a service and its clients exchange to achieve a certain business goal. Business protocols change due to changes in policies, regulations, and changes in the operational behavior of services.

3. Policy induced changes: These describe changes in policy assertions and constraints on the service, which prescribe, limit, or specify any aspect of a business agreement that is possible, agreed to among interacting parties. Policies may describe constraints external to constraints agreed by interacting parties in a transaction and include universal legal requirements, commercial and/or international trade and contract terms, public policy (e.g., privacy/data protection, product or service labeling, consumer protection), laws and regulations that are applicable to parts of a business service. For instance, a procurement processes can codify an approval process in such a way that it can be instantly modified as corporate policies change. In most cases existing processes need to be redesigned or improved to conform to new corporate strategies and goals.

4. Operational behavior changes: These concentrate on analyzing the effects and side (cascading) effects of changing service operations. If, for example, we consider an order management service we might expect to see a service that lists "place order", "cancel-order," and "update order," as available operations. If now the "update-order" operation is modified in such a way that it includes available-to-promise functionality that dynamically allocates and reallocates resources to promise and fulfill customer orders, the modified operation must guarantee that if part of the order is outsourced to a manufacturing partner, the partner can fulfill its order on time to meet agreed upon shipment dates. This requires understanding of where time is consumed in the manufacturing process, what is normal with respect to events timeliness to the deadline, and to understand standard deviations with respect to that process events on-time performance.

Vast amount of researches have paid severe attention to flexibility, agility, and managing changes in SOEs and collaboration between web services. Previous researches on changes in Web services and SOEs have primarily focused on technical issues in the computer science field, leaving unanswered the business impacts of changes as well as optimal business strategies of a SOE. You can see some of this type of studies in table1. Compared to issues, models, and implementation of service oriented cultures, it is easier to focus on technology. Also, it is important to understand the underlying components of the technology. Think of an office building: Technology makes it possible to construct efficient, networked, and highly intelligent buildings, but what makes an enterprise successful are the people and the culture of its organization. Serving is praised as a virtue, yet it seems much more difficult to realize it in practice. Inspiration and goose bumps well up when stories are told of unselfish sacrifice and service for noble causes: in social service, in politics, in religion, and why not even in the military. However, there is a flip side. Our culture sometimes places the wrong emphasis when it rewards greed, aggrandized egos, and cut-throat approaches in climbing the corporate ladder. In a flattened world, we cannot afford to reward selfish ambitions. The service culture sees the success of customer, shareholder, employee, and partner as essential requirements to fulfillment. It is service oriented. In a service oriented enterprise, greed is not good. Success is a side effect, not the focus. Success is not just about finances; it is about how well others are served and elevated. A service oriented culture means our main function and purpose is found in serving others: helping them achieve their potential [15].

Tab. 1. Literature of Changes in SOEs

Goal	Approach/Focus	References
Work Flow & Business Process Management	BPR & Process Instance	[18], [21], [6], [30], [17], and [14]
Formalizing SOE& Modeling Changes	State Model	[31]
	Petri Net	[1], [2], [33], and [27]
	Ad hoc Model	[19]
Change Protocol	Collaboration	[26] and [16]
Web Service Presentation & Selection	Versioning	[3]
Flexible framework	Change Module	[29], [15], [25], and [7]

There are two approaches to specifying changes: top-down and bottom-up. Top-down changes are motivated by the SOE's business goals, and focus on changes that are usually government or business mandated. Unlike top-down changes, bottom-up changes are initiated by the member services and consider the uncertainty of the underlying member services. In our work, we deal with top-down changes. We have considered top-down changes in SOEs by an economic perspective based on service culture notion.

4. Problem Definition

A top-down change is expected to occur frequently during the life-cycle of a SOE due to the dynamic environment (e.g., user's requirement, marketing, laws). It is always affiliated with a new requirement on a SOE's member services and the way they cooperate with each other [19]. Let us consider a travel agency service to illustrate a top-down change. This service provides its travel packages through collaboration with other partner services such as airline, hotel and credit card services. After finding its candidate partners (e.g.,

through a discovery service) it will contact them to determine the compatibility with regards to its requirements and capabilities, and binds with desired services. Afterwards, it will interact with its partners according to the agreements and contracts made at binding time. This scenario has been depicted in Figure2.

Motahari Nejad et al. (2008) [22] identify the following security dimensions for collaboration between services: secure messaging, resources protection, security properties binding, contractual interactions, and federated trust management.

Now consider that, the Travel agency to satisfy demand of its customers has made a decision to improve the security level of its service. For this reason all of the above security dimensions may need to change. It can have an impact on the contract specified between the parties involved. Thus, a change (in this example security improvement) may need to adapt to meet a new contract.

Khriss et al. (2008) [16] propose a new approach to support adaptability for such collaborative processes. This approach uses a protocol, called Change Protocol for Collaboration (CPC), for managing the changes that can have incidence on the contract. The CPC protocol is a two-phase commit-like protocol and consists of five messages: Notify, Accept, Deny, Proceed, and Cancel (see Figure3 and Figure4). When a trading partner (playing the Master role) wants to change its business process and this change can affect its partners (playing the Slave role), it sends a message notifying its partners of this change (1). Upon receiving this message, the slaves enter the Notified state (a slave is initially in state Idle). The slaves can then accept (2) or deny (2') adapting their business processes. When a slave agrees to the change,
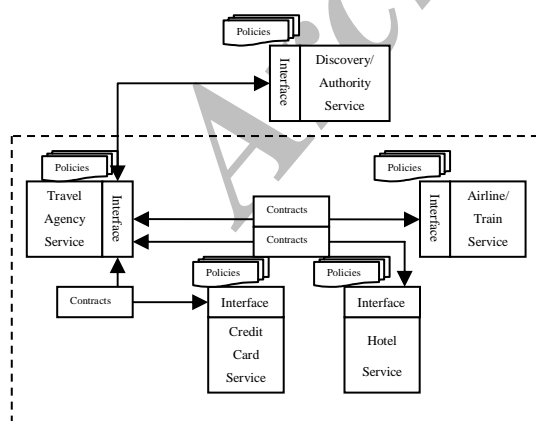

Fig. 2. Travel Agency SOE

it enters the Accepted state; otherwise, it enters the Denied state. For the master, two cases are then to be considered. If it receives an Accept message from all its slaves, it sends a Proceed message to inform its partners that they should adapt their business processes

(3). Upon receiving this message, a slave enters the Proceed state. The second case is when the master receives a Deny message from a slave: in this case, it sends a Cancel message to all its slaves informing them that the change is canceled (3'). All slaves then enter the Canceled state. Note that canceling a change does not mean that a slave has a power to veto; it only means that this slave cannot adapt its business process in order to meet the new requirement. The master will then simply react by resubmitting the change after replacing the partners (slaves) that denied.

We believe this approach (using CPC protocol) can be useful in dealing with top-down changes, but the elements which affect the services' decisions must be considered. Khriss et al. (2008) [16] mentioned that the partners' activity has two aspects: The first is purely business related, while the second is technical. But they didn't turn to these aspects.

However, for decision making about changes, economic justification is one of the major elements of business related aspect. We have an investigation in this field with an economic perspective. Our paper contributes to the existing literature by considering explicit methods for motivating services to non-contracted changes.
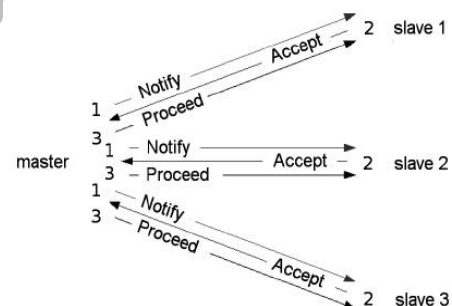

Fig. 3. Messages exchanged between Master and Slave partners – when change is accepted
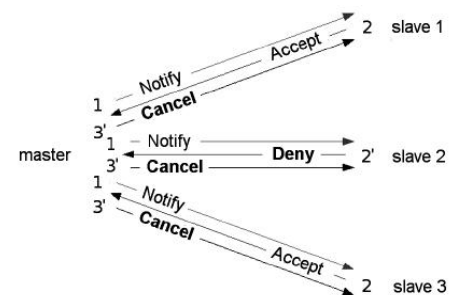

Fig. 4. Messages exchanged between Master and Slave partners – when change is denied

5. Proposed Approach

We utilize game theoretic modeling to study top-down change incentives for member services. In this paper, top-down change incentives denote the guarantee of joint gains among the services whenever the total payoff of the SOE increases.

5.1. Top-Down Change Model

Our approach to the problem is ex-ante contracting. The idea is that, if all parties can be guaranteed an increase in benefit whenever a change is implemented, then the services have the incentive to improve. Now, we shall construct a game theoretic model that captures the case. Players of the game are $N = \{1, \ldots, n\}$, where $N_S = \{1, \ldots, n-1\}$ denote the services and n is the client (core of SOE). In the status quo, each service i once he serves the client, receives a positive payment p_i from the client n and shells out non-negative costs v_i.

Hence, profit of i is $p_i - v_i$.

Let Δv_i denote change of i's costs. Furthermore, let Δp_i denote change of i's fixed payment that the client would conduct due to the change in his costs, and let Δp denote the vector $(\Delta p_1, \cdots, \Delta p_{n-1})$. Hence, the change in the service i's profit is:

$$\pi_i = \Delta p_i - \Delta v_i \qquad \forall i \in N_S \qquad (1)$$

Since the client n makes the payments Δp_i to the services $1, \ldots, n-1$, the client's profit change is:

$$\pi_n = \Delta p_n - \sum_{i=1}^{n-1} \Delta p_i \qquad (2)$$

Where Δp_n denote the change in consumers' willingness to pay caused by offering an improved service (e.g. with higher security level). Equation (1) denote utility to services and (2) denote utility to the client n (We assume that, in the status quo all the players' utilities are equal to zero).

We present the problem as a two-stage game (see Figure5). In the first stage, the client (player n) defines a utility-sharing rule, □, by which the services' payments (p_i's) will be redefined if the services' costs (v_i's) change due to a security improvement. Hence, the set of (pure) strategies available to the client n is the family f of functions which map the changes in the services' costs (the Δv_i's) to changes in the payments (the Δp_i's):

$$\phi \in f, f = \{\phi \mid \phi : R^{n-1} \mapsto R^{n-1}\}$$

Where $\square_i$ denote the rule that concerns service i, i.e. $\Delta p_i = \square_i$.

In the second stage of the game, one or more of the services $1, \ldots, n-1$ discover(s) an efficiency-improving option. The implementation of the idea would improve the security of the network but would also require transfer of costs inside the network. Knowing the utility-sharing rule $\square_i$, the service may now choose his strategy c_i between coming up with the idea ($c_i = a$) or withholding the idea ($c_i = b$). Let us denote the set of strategies available to the services $i \in N_S$ by $C_i = \{a, b\}$.

Formally, we can demonstrate the security improvement game as follow:

$$\Gamma = (N, (C_i)_{i \in N_S}, \phi, (\pi_i)_{i \in N}) \qquad (3)$$

Where

N is the set of players $N = \{1, \ldots, n\}$;
N_S is the set of services $N_S = \{1, \ldots, n-1\}$;
C_i is a set of strategies available to the services $i \in N_S$, $C_i = \{a, b\}$;
□ is a function, $\phi : R^{n-1} \mapsto R^{n-1}$, which denotes the strategy of the client n, i.e. the utility-sharing rule;
π_i denotes the utility to player $i \in N$. For service $i \in N_S$, π_i is defined in (1) and for the client n, is defined in (2).
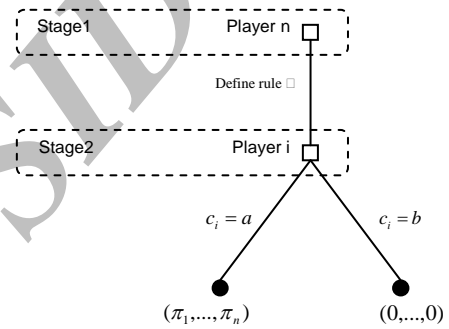

Fig. 5. General Form of Game

In stage one, the client n may act as a Stackelberg leader [4, 16, 36] and try choosing his strategy □ in such a way that the services $i \in N_S$ are encouraged to propose improvement ideas. In Stackelberg games the players are categorized as leaders and followers (or masters and slaves).

The objective of the leader is to give the follower such incentives to play optimally from the viewpoint of the leader. It is therefore crucial to the outcome of the game that the services $i \in N_S$ have the information of the rule □ at the moment of their decision-making in stage two. This can be implemented by an ex-ante contract [13] where the utility-sharing rule □ is explicitly defined.

In stage two, a rational service $i \in N_S$ chooses to propose improvement ideas (play $c_i = a$) if its consequences are profitable, i.e. if

$$\pi_i \geq 0 \qquad (4)$$

5.2. Creating a Win-Win Situation Through a change

We formulate the following proposition:

Proposition1: in a top-down change, a no-loss situation among the members of the network is possible whenever the change is raised from demand of consumers.

Proof: assume security level of the service is improved and it is valuable for consumers, so the SOE can gain a higher price from consumers. Formally:

$$\sum_{i=1}^{n-1} \Delta v_i \le \Delta p_n \tag{5}$$

A no-loss situation requires that the utility to each party is non-negative (from (4)). Using (1) and (2):

$$\pi_i \ge 0$$
$$\Rightarrow \Delta p_i - \Delta v_i \ge 0 \qquad \Rightarrow \Delta p_i \ge \Delta v_i \qquad \forall i \in N_S \tag{6}$$

$$\sum_{i=1}^{n-1} \Delta p_i \le \Delta p_n \tag{7}$$

If we let $\Delta p_i = \Delta v_i$ then we obtain:

$$\sum_{i=1}^{n-1} \Delta v_i = \sum_{i=1}^{n-1} \Delta p_i \overset{(5)}{\le} \Delta p_n$$

That is, both conditions (6) and (7) are satisfied, which shows our proposition is correct.

It is noteworthy, that the total utility derived from the security improvement is:

$$\pi = \Delta p_n - \sum_{i=1}^{n-1} \Delta v_i \overset{(5)}{\ge} 0 \tag{8}$$

(8) denotes total surplus that can be shared among the members of the network, after the no-loss conditions (6) and (7) are satisfied. Hence, instead of a mere no-loss situation, actually a win-win situation is created.

We apply three different ways to share the surplus utility gained through a change.

First, we apply the egalitarian rule, which reflects fairness and complete cooperation within the network. Second, we use the utility-sharing rule according to Nash's relative threats solution. It, in addition to the egalitarian rule, models the interdependencies between the members of network. Third, the rule according to Harsanyi's modified Shapley value takes into account possibility of coalition formation inside the network [12, 16]. We shall utilize the following definitions throughout the analysis: Let $N_v \subseteq N_S$ be the set of services whose costs change due to the security improvement, i.e. $N_V = \{i \in N_S | \Delta v_i \ne 0\}$. We demonstrate the cardinality of N_V by $|N_v|$.

5.2.1. Egalitarian Solution

The egalitarian rule implies that the utility-sharing rule, □, is constructed based on the following two conditions:

$$\pi_n = \pi_i \qquad \forall i \in N_V \tag{9}$$

$$\pi_i = 0 \qquad \forall i \in N_S - N_V \tag{10}$$

In the other words, condition (9) says that the involved services ($i \in N_V$) and the client n benefit equally.

Condition (10) denotes that the payoff of other services, whose costs are not affected, stay unchanged. The involved services experience a change in costs (Δv_i), which can be positive or negative. In the egalitarian solution, the total surplus utility, π, should be shared equally among the involved services and client. Therefore, the conditions (9) and (10) imply the following utility-sharing rule □:

$$\phi_i = \begin{cases} \Delta v_i + \dfrac{\pi}{|N_V| + 1} & \forall i \in N_V \\ 0 & \forall i \notin N_V \end{cases} \tag{11}$$

In game theoretic terms, the allocation (11) is called a λ-egalitarian solution since (see e.g. [23], p. 382):
(I) It satisfies the weak efficiency condition;
(II) The player's gains are weighted.
First, weak efficiency guarantees that all the available utility will be shared among the players. Second, the conditions (9) and (10) can be interpreted so that the utilities to the involved players are equally weighted, whereas the weights of the players not involved are zero. Furthermore, solution (11) satisfies the conditions (6) and (7) if condition (5) be true.

5.2.2. Use of Threats in Contract Negotiation

It may be useful to examine what happens if the suggested alteration in prices is not commonly accepted. Therefore, assume that each player $i \in N$ has an additional possible strategy, threat τ_i, which is the termination of the partnership. If a player executes the threat strategy, the game ends in disagreement. The payoffs to the players in disagreement are denoted by $\tau_1, \ldots, \tau_n$. Since the termination of partnership normally causes additional transaction costs to each party, the τ's are usually negative.

Because of the nature of the game, it is not relevant to consider that the players would threaten each other, unless the game has advanced beyond Stage2 and service $i \in N_S$, who is making the decision in Stage2, has decided to come up with his idea, i.e. play $c_i = a$. Thus, Figure6 expands the game tree of Figure5 to include the possibility of threatening to terminate the partnership. One possible way to take the threat strategies into account is to generalize Nash's theory to n players (for two player games, see [24]). Mathematically this is straightforward. The Nash product for n players becomes:

$$\prod_{i=1}^{n} x_i - \delta_i \tag{12}$$

Where x_i is the share of utility to player i in cooperation, and δ_i is the disagreement payoff to player i. The maximization of the Nash product (12) defines a unique strongly-efficient vector x, which is the Nash solution to the n-player bargaining problem (see [23], p. 417).

Hence, the share of utility in the threat game presented in Figure6 can be defined by the unique strongly-efficient vector π that maximizes the Nash product [16]:

$$\prod_{i \in N_V \cup \{n\}} (\pi_i - \tau_i) \qquad (13)$$

Maximization of (13) is equivalent to solving the following conditions:

$$\pi_i - \tau_i = \pi_j - \tau_j \qquad \forall i, j \in N_V \cup \{n\} \qquad (14)$$

$$\sum_{i \in N_V \cup \{n\}} \pi_i = \Delta p_n - \sum_{i \in N_V \cup \{n\}} \Delta v_i \qquad (15)$$

Condition (14) denotes that the utility to each player is related to the amount of losses in disagreement. Condition (15) takes care that all the available utility is shared. Conditions (14) and (15) form a linear system of $|N_V| + 1$ equations containing the same number of unknown variables (the π_i's). Thus, solving the system for π_i's defines vector π uniquely. The utility-sharing rule $\square$ can then be calculated from (1):

$$\phi_i(\tau) = \begin{cases} \pi_i(\tau) + \Delta v_i & \forall i \in N_V \\ 0 & \forall i \notin N_V \end{cases} \qquad (16)$$

Where τ denotes the vector that consists of τ's , $i \in N_V \cup \{n\}$.
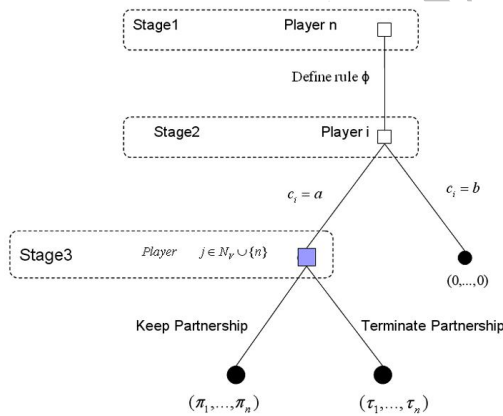

Fig. 6. Game Including Threats

5.2.3. Coalitions in Contract Negotiation

In Section 5.2.2, the game model has been constructed for n players without considering coalitions. In the following a coalitional analysis of the game is presented. For each $i \in N_V$ and $j \in N$, let $\tau_j(i)$ denote the utility (or cost) to player j if the contract between i and n is terminated. For convenience, we write $\tau_i(i) = \tau_i, i \in N_V$. We assume that $\sum_{i \in N_V} \tau_i < 0$,

that is, there is always at least one service $i \in N_V$ such that $\tau_i < 0$. This assumption eliminates the possibility that all the services ally against the client.

Let us define the coalitional threat game as a generalization of the game (3) as follows:
$$\Psi = (S \subseteq N, C_S, v(S))$$

Where S is a coalition, C_S is the set of strategies of the players in S, and $v(S)$ is a characteristic function. Originally, von Neumann and Morgenstern (1944) [35] defined $v(S)$ by a minimax representation. We shall, however, use the definition presented by Harsanyi (1963) [12]. The idea is that, instead of maximizing merely the total utility, a coalition should maximize the difference between its own total utility and the competitors' total utility. Thus, the coalitions' optimal strategies become

$$C_S^* = \{C_S \mid \max \sum\nolimits_{i \in S} \pi_i(C_S, C_{N-S}^*) - \sum\nolimits_{j \in N-S} \pi_j(C_S, C_{N-S}^*))\} \quad (17)$$

$$C_{N-S}^* = \{C_{N-S} \mid \min(\sum\nolimits_{i \in S} \pi_i(C_S^*, C_{N-S}) - \sum\nolimits_{j \in N-S} \pi_j(C_S^*, C_{N-S}))\} \quad (18)$$

The characteristic function is defined as:

$$v(s) = \sum_{i \in S} \pi_i(C_S^*, C_{N-S}^*) \qquad (19)$$

Where the strategies (C_S^*, C_{N-S}^*) are obtained from (17) and (18).

An elegant means for finding an outcome for n-player bargaining is the Shapley value, which was introduced by Shapley (1953) [32]. The Shapley value for player i of a coalitional game Ψ is [16]:

$$\varphi_i(\Psi) = \sum_{S \subseteq N-\{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}(v(S \cup \{i\}) - v(S)) \qquad (20)$$

Where $v(X)$ is the characteristic function (the worth) of coalition X. As has been discussed earlier, the possibility to use threats is an essential part of the game. For this purpose, Harsanyi (1963) [12] presents a modified Shapley value, which is calculated from the original formula (20) but with the characteristic function defined in (17), (18) and (19).

In our game, the modified Shapley values for the services $(1, \ldots, n-1)$ and the client n are

$$\varphi_i = \frac{\pi}{|N_V|+1} - \frac{\tau_n(i)}{2} + \frac{\tau_i}{2}, \qquad i \in N_V \qquad (a)$$

$$\varphi_j = 0, \qquad\qquad j \in N_S - N_V \qquad (b) \qquad (21)$$

$$\varphi_n = \frac{\pi}{|N_V|+1} + \frac{\sum_{i \in N_V} \tau_n(i)}{2} - \frac{\sum_{i \in N_V} \tau_i}{2} \qquad (c)$$

It is straightforward to verify that the players' modified Shapley values (21) sum up to the total available utility:

$$\sum_{i \in N_V} \varphi_i + \sum_{j \in N_S - N_V} \varphi_j + \varphi_n = \pi$$

That is, the allocation $\Phi = (\varphi_1, ..., \varphi_n)$ is efficient.

The services with $\tau_i < 0$ are in a weaker bargaining position than the other services. We could say that the weak partners are more dependent on the client than the other services. Thus, the modified Shapley value for the weak services is strictly less than that for the other services. In fact, the services not dependent on the client obtain the same amount of utility that the egalitarian solution (12) would give, and the services dependent on the client forfeit an amount of utility, proportional to the strength of the client's threat $(\tau_n(i), \tau_i)$, to the client.

The utility-sharing rule $\Box$ according to the modified Shapley value is obtained by replacing π_i in (1) by φ_i:

$$\varphi_i = \Delta v_i + \frac{\pi}{|N_V| + 1} - \frac{\tau_n(i)}{2} + \frac{\tau_i}{2}, \qquad i \in N_S \qquad (22)$$

Rule (22) takes into account this fact that some of the services are more dependent on the client, *n*, than the others. Hence, it is reasonable that the incentive for the former is lower than latter.

5.3. Numerical Example

This section applies the results of the proposed model to a numerical example of a given SOE with two member web services. All the numerical values are fictitious. Following the notation of previews sections, Assume that:

1 and 2 denote the services and 3 denotes the client;
$\Delta p_3 = 10\$$ per service
$\Delta v_1 = 8\$$ per service
$\Delta v_2 = -2\$$ per service

From (8) total surplus that can be shared among the members of the network is

$$\pi = \Delta p_3 - \sum_{i=1}^{2} \Delta v_i = 10 - (8 - 2) = 4\$ \text{ per service}$$

From equation (11), for egalitarian solution we have:

$$\Delta p_1 = \Delta v_1 + \frac{\pi}{|N_V| + 1} = 8 + \frac{4}{2 + 1} = \frac{28}{3}\$$$

$$\Delta p_2 = \Delta v_2 + \frac{\pi}{|N_V| + 1} = -2 + \frac{4}{2 + 1} = -\frac{2}{3}\$$$

Now, from (1) we can calculate services' corresponding utilities:

$$\pi_1 = \Delta p_1 - \Delta v_1 = \frac{28}{3} - 8 = \frac{4}{3}\$$$

$$\pi_2 = \Delta p_2 - \Delta v_2 = -\frac{2}{3} - (-2) = \frac{4}{3}\$$$

From (2), utility to the client is

$$\pi_3 = \Delta p_3 - \sum_{i=1}^{2} \Delta p_i = 10 - \frac{26}{3} = \frac{4}{3}\$$$

As you can see, in the egalitarian solution, all the participants' utilities are equal. It is noteworthy that, when no threats exist, in the determination of Δp_i's it is sufficient to know the values of Δv_i's; no additional information is needed.

The relative threats solution enables the use of threats, i.e. such actions that can harm a party if committed by another party. To illustrate how threats can affect the reallocation of the payments, let us assume that the client can terminate the contract with the service 1. Furthermore assume that the client can easily find a substitute service, whereas for the service 1, it is difficult to find a new customer. Hence, if the contract is terminated, the losses to service 1 are valued at $\tau_1 = -1.5\$$, proportioned to the income of service 1 from the present client. The client would not suffer any losses from the termination of the contract $(\tau_3(1) = 0\$)$. Hence, the client possesses a credible threat against service 1.

The utilities according to the relative threats solution are calculated from the system of linear equations (15) and (16), which in this example consists of three equations:

(*a*) $\pi_1 - \tau_1 = \pi_2$

(*b*) $\pi_2 = \pi_3 - \tau_3$

(*c*) $\pi_1 + \pi_2 + \pi_3 = \pi$

By solving equations (a)-(c) simultaneously, we obtain:

$$\pi_1 = \frac{1}{3}\$, \quad and \quad \pi_2 = \pi_3 = \frac{11}{6}\$$$

Therefore, changes in the payments to services become:

$$\Delta p_1 = \pi_1 + \Delta v_1 = \frac{1}{3} + 8 = \frac{25}{3}\$$$

$$\Delta p_2 = \pi_2 + \Delta v_2 = \frac{11}{6} - 2 = -\frac{1}{6}\$$$

That is, in consequence of service 1's dependence on the client, service 1 loses 1\$ in comparison with the egalitarian solution. This 1\$ is divided evenly among the client and service 2. Figure 7 shows the shares of each party with respect to different values of τ_1.
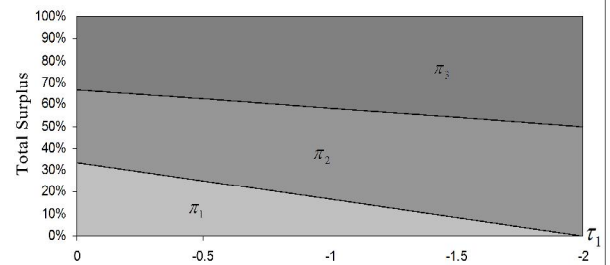

Fig. 7. Sharing total surplus between players using Nash threat solution

In determining the solution according to the modified Shapley value, we assume interdependencies inside the network same as the previous solution. That is, the

client possesses a credible threat, $\tau_1 = -1.5\$$, against service 1. Changes in the payments are calculated from (22):

$$\Delta p_1 = \Delta v_1 + \frac{\pi}{|N_v|+1} - \frac{\tau_3(1)}{2} + \frac{\tau_1}{2} = 8 + \frac{4}{3} - \frac{3}{4} = \frac{103}{12}$$

$$\Delta p_2 = \Delta v_2 + \frac{\pi}{|N_v|+1} = -2 + \frac{4}{3} = -\frac{2}{3}$$

With this reallocation of payments, the surplus is shared as follows (from (1) and (2)):

$$\pi_1 = \Delta p_1 - \Delta v_1 = \frac{103}{12} - 8 = \frac{7}{12}\$$$

$$\pi_2 = \Delta p_2 - \Delta v_2 = -\frac{2}{3} - (-2) = \frac{4}{3}\$$$

$$\pi_3 = \Delta p_3 - \sum_{i=1}^{2}\Delta p_i = 10 - (\frac{103}{12} - \frac{2}{3}) = \frac{25}{12}\$$$

As it shown in figure8, in this solution (modified Shapley value), dependency of service is taken into account. However, only one participant who benefits from weakness of service 1 has the potency to execute the threat (i.e. the client).
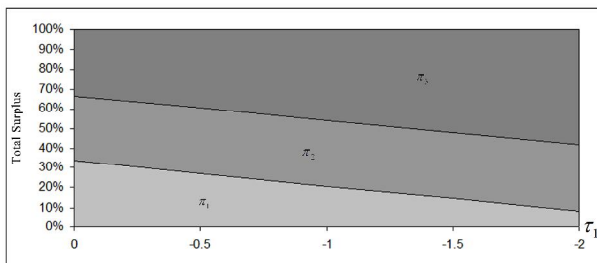

Fig. 8. Sharing total surplus using modified Shapley value

6. Conclusion and Future Work

We may distinguish between two kinds of changes in SOEs: top-down and bottom-up changes. In this paper, we have introduced a game theoretic model for the identification of top-down changes in SOEs. Often, when a top-down change is implemented, the costs of some member services increase, whereas the costs of some other services reduce. The first criterion for the change is that the total amount of profits increases. The other criterion is that each of the services is better off after the change has been carried out. That is, the services whose costs increase need a compensation payment in order to accept the implementation of the change. By our model, we show that these criteria can be satisfied and that joint gains are achievable through change.

This paper considers three incentive mechanisms, the first incentive mechanism introduced shares the utility equally among the companies involved in the change. The second utility-sharing rule is based on the Nash's

bargaining solution, which accommodates the possible biased interdependencies inside the network. Additionally, the third rule, based on the Harsanyi's modified Shapley value, takes into account the possible coalition formation among the network parties. Since the three rules are analytically solvable, the principles of utility sharing can be implemented, for instance, as ex-ante contracts.

As a summary, the main contributions of this paper are as follows. First, we have constructed a game theoretic model, which can be used to study top-down changes in SOEs. Second, we have designed explicit mechanisms that encourage services to innovations. Third, our approach takes into account the biased interdependencies inside SOEs.

A target for further development could be to extend the model to contain uncertainties of cost changes, disagreement outcomes (i.e. different types of the players), network externalities, and presence of SOEs that use same services.

References

[1] Akram, M.S., Bouguettaya, A., *"Managing Changes to Virtual Enterprises on the Semantic Web,"* WISE 2004, Springer, LNCS 3306, 2004, pp. 472–478.

[2] Akram, M.S., Medjahed, B., Bouguettaya, A., *"Supporting Dynamic Changes in Web Service Environments,"* ICSOC 2003, Springer, LNCS 2910, 2003, pp. 319-334.

[3] Bachmann, R., *"Challenges of Web Service Change Management,"* The SAP Developer Network: http://sdn.sap.com, 2006.

[4] Basar, T., Olsder, G.J., *Dynamic Noncooperative Game Theory*, Academic Press, 1982.

[5] Casati, F., Shan, E., Dayal, U., Shan, M., *"Business-Oriented Management of web Services,"* Communications of the ACM, 46(10), 2003, pp. 55-60.

[6] Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M., *"Adaptive and Dynamic Service Composition in eFlow,"* Technical report, Software Technology Inc, 2001.

[7] Chen, H.M., *"Towards Service Engineering: Service Orientation and Business-IT Alignment,"* 41st IEEE International Conference on System Sciences, 2008.

[8] Chesbrough, H., Spohrer, J., *"A Research Manifesto for Services Science,"* Communications of the ACM, 49(7), 2006, pp. 35-42.

[9] Curbera, F., Duftler, M., Khalaf, R., Nagy ,W., Mukhi, N., Weerawarana, S., *"Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI,"* IEEE Internet Computing, 6(2), 2002, pp. 86-93.

[10] Demirkan, H., Kauffman, R.J., Vayghan, J.A., Fill, H.G., Karagiannis, D., Maglio, P.P., *"Service-Oriented Technology and Management: Perspectives on Research and Practice for the Coming Decade*," Electronic Commerce Research and Applications, 2008, doi:10.1016/j.elerap.2008.07.002.

[11] Gray, J., "*Distributed Computing Economics*," ACM Queue, May/June 2003, pp. 63-68.

[12] Harsanyi, J.C., *"A Simplified Bargaining Model for the n-Person Cooperative Game*," International Economic Review, 4(2), 1963, pp. 194–220.

[13] Jarimo, T., Pulkkinen, U., Salo, A., "*Encouraging Suppliers to Process Innovations: A Game Theory Approach,"* International Journal of Technology Intelligence and Planning, 1(4), 2006, pp. 403-423.

[14] Khalaf, R., Keller, A., Leymann, F., *"Business Processes for Web Services: Principles and Applications*," IBM Systems Journal, 45(2), 2002, pp. 425-446.

[15] Khoshafian, S., *Service Oriented Enterprises*, New York: Taylor & Francis Group, 2007.

[16] Khriss, I., Lévesque, E., Tremblay, G., Jacques, A., *"Towards Adaptability Support in Collaborative Business Processes,"* IEEE Computer Society, 2008.

[17] Lee, S., Kim, T., Kang, D., Kim, K., Lee, J.Y., *"Composition of Executable Business Process Models by Combining Business Rules and Process Flows*," Expert Systems with Applications, 33, 2007, pp. 221-229.

[18] Leymann, F., Roller, D., Schmidt, M.T., *"Web Services and Business Process Management,"* IBM Systems Journal, 41(2), 2002, pp. 198-211.

[19] Liu, X., Bouguettaya, A., "*Managing Top-down Changes in Service-Oriented Enterprises*," IEEE International Conference on Web Services (ICWS), 2007.

[20] Magilo, P.P., Srinivasan, S., Kreulen, J.T., Spohrer, J., *"Service Systems, Service Scientists, SSME, and Innovation*," Communications of the ACM, 49(7), 2006, pp. 81-87.

[21] Meng, J., Krithivasan, R., Su, S.Y.W., Helal, S., "*Flexible Inter-enterprise Workflow Management using E-Services*," Proceedings of the 4th IEEE Int'l Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS), 2002.

[22] Motahari Nejad, H.R., Skogsrud, H., Benatallah, B., Casati, F., "*Securing Service-Based Interactions: Issues and Directions*," IEEE Distributed Systems Online, http://dsonline.computer.org, 2008.

[23] Myerson, R.B., *Game Theory: Analysis of Conflict*, First Harvard University Press, 1997.

[24] Nash, J.F., *"Two-Person Cooperative Games*," Econometrica, 21(1), 1953, pp. 128–140.

[25] Nayak, N., Linehan, M., Nigam, A., Marston, D., Jeng, J.J., Wu, F.Y., Boullery, D., White, L., Nandi, P., Sanz, J.L.C., *"Core Business Architecture for a Service-Oriented Enterprise,"* IBM System Journal, 46(4), 2007, pp. 723-742.

[26] Orriens, B., Yang, J., Papazoglou, M., *"A Rule Driven Approach for Developing Adaptive Service Oriented Business Collaboration*," ICSOC 2005, Springer, LNCS 3826, 2005, pp. 61–72.

[27] Ouyanga, C., Verbeekb, E., van der Aalsta, W.M.P., Breutela, S., Dumasa, M., ter Hofstedea, A.H.M., *"Formal Semantics and Analysis of Control Flow in WS-BPEL*," Science of Computer Programming, 67, 2007, pp. 162-198.

[28] Papazoglou, M.P., *"The Challenges of Service Evolution,"* CAiSE 2008, Springer, LNCS 5074, 2008, pp. 1-15.

[29] Peltz, C, "*Web Services Orchestration and Choreography,"* IEEE Computer Society, 2003.

[30] Rinderle, S., Reichert, M., Dadam, P., "*On Dealing with Structural Conflicts Between Process Type and Instance Changes,*" BPM 2004, Springer, LNCS 3080, 2004, pp. 274-289.

[31] Robinson, P., Karabulut, Y., Haller, J., *"Dynamic Virtual Organization Management for Service Oriented Enterprise Applications*," IEEE Computer Society, 2005.

[32] Shapley, L.S., "*A Value for N-Person Games*," in Annals of Mathematics Studies 28, editors H. W. Kuhn and A. W. Tucker, Princeton University Press, 1953, pp. 307–317.

[33] van der Aalst, W.M.P., Lassen, K.B., "*Translating Unstructured Workflow Processes to Readable BPEL: Theory and Implementation*," Information and Software Technology, 50, 2008, pp. 131-159.

[34] Varadan, R., Channabasavaiah, K., Simpson, S., Holley, K., Allam, A., "*Increasing Business Flexibility and SOA Adoption Through Effective SOA Governance*," IBM Systems Journal, 47(3), 2008, pp. 473-488.

[35] von Neumann, J., Morgenstern, O., *Theory of Games and Economic Behavior,* Princeton: Princeton University Press, 1944.

[36] von Stackelberg, H., *Market Structure and Equilibrium*, Vienna: Julius Springer, 1934.