Annals of Emerging Technologies in Computing (AETiC)
Vol. 6, No. 3, 2022

Research Article

Dynamic Context Driven Re-configurable Business Process

Priyanka Chakraborty* and Anirban Sarkar

National Institute of Technology Durgapur, India
priyanka1nitdgp@gmail.com; sarkar.anirban@gmail.com
*Correspondence: priyanka1nitdgp@gmail.com

Received: 31st October 2021; 24th June 2022; Published: 1st July 2022

Abstract: The building of a re-configurable business process (BP) has gained importance in business organizations. It helps the organization to adapt to the agility in business goals. A proper context-driven re-configurable BP should be capable of integrating dynamic context information. However, this is absent in the existing studies. As a result, providing a suitable, expressive and re-configurable BP to the business organization stakeholders has become a challenging issue. The prevailing research works lack the proper consideration and suitable incorporation of the context-driven services to make a BP re-configurable. And then it can quickly respond and change its behavior to adapt to the rapid and unpredictable changing business environment. In addition, those methods hardly come up with any appropriate technique to use the set of specified goals to extract context-driven services. Those business goals are determined by the group of stakeholders of a business organization. This paper proposes a new method of re-configuring context-driven from a defined goal to sort out these vital challenges. Present context data is included in an existing BP to achieve a modified goal which immensely benefits the end-users. Thus, this approach is intrinsically highly user-centric, reusable, fast and inexpensive. To achieve this, an algorithm called Context-driven Re-configurable Business Process Achievement Algorithm (CDRBPA) is introduced and implemented. Based on Primary Context (PC), three software metrics, namely, Degree of re-usability (D_{RUPC}), Degree of re-appropriation (D_{RAPC}) and Degree of re-configurability (D_{RPC}) have been proposed to measure the modifications done to the existing BP. In conclusion, various case studies with different complexities have been performed to show the strength of the proposed algorithm.

Keywords: ***Context-driven Re-configurable business process; Degree of re-usability; Degree of re-appropriation; Degree of re-configurability; context-driven services***

---

1. Introduction

A context-driven business process (BP) is a BP that includes the current context information, describing the surrounding environmental situation associated with its BP elements. Consequently, the BP can easily work on the dynamic context data and modify its functioning as required.

Context [1] in BP is defined by the surrounding knowledge about the associated entities. The surrounding environment may include information about the physical world (like information about location, time, speed, temperature, device information etc.) or the logical world (like the identity, emotional state, personal choices etc.). Physical-world information is mostly obtained from the sensor, data while the logical world information is derived from the users' interaction pattern with various applications or by browsing users' profiles. Thus, the dynamic context data can be obtained from distinct sources and it is system-specific. Context is categorized into the Primary Context (PC) and the Secondary Context (SC). The PC is the obligatory information and describes the present surrounding environmental situation of an entity. Hence, the flow of BP is determined majorly by the PC values. The corresponding services, gateways and data within a BP are also highly influenced by those values. Secondary or auxiliary context provides only the additional information to describe the situation of an entity more accurately.

Priyanka Chakraborty and Anirban Sarkar, "Dynamic Context-driven Re-configurable Business Process", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 37-55, Vol. 6, No. 3, 1st July 2022, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2022.03.004, Available: http://aetic.theiaer.org/archive/v6/v6n3/p4.html.

Context-driven services [2] can quickly respond to their surrounding environmental information termed as the context data that is dynamic and system-specific. Then it can easily adapt to this ever-changing surrounding environment. Let us consider a service app named as the Market Analysis App that is context-driven and used by the business organizers to analyze the current market situation to increase the amount of sales. The end-user location, the app usage time and date, the current availability of the product and present demand of the product are the four determining factors based upon which the app gives the preferable and profitable value range to its end-user. Thus, the business organizer is benefitted from this context-driven service app for the decision-making process to determine the selling price of that particular product. Another real-life scenario can be taken as the Airline Fair App that can provide its end-users with a suitable list of flight tickets considering their preferences. The four relevant contexts to be considered by the app are Date and time of flight booking, preference of the class, current pandemic situation and availability of seats. Among them, the date and time of flight booking is the primary context because this is the prime decisive factor for fixing the fare of the flight. Thus, this app facilitates its users by providing a suitable and relevant list of flight ticket details.

A BP [3] is a collection of a set of services (represented as the activities), gateways, sequence flows, events and the data sets where multiple user inputs are provided and expressive output is produced. The suitable incorporation of the services guided by contexts to a pre-existing BP forms it re-configurable. Then the BP can rapidly respond to the dynamic business scenario and change its behaviour suitably. Thus, the BP can satisfy the newly modified goal with the minimal changes in the BP elements. Thus, a re-configurable BP driven by contexts can deliver expressive, appropriate and reliable services to customers.

Dynamic context information refers to the ever-changing context data that describes the continuously varying surrounding environmental situation in which the BP elements operates. Thus, the context data also persistently changes with time according to the rapidly changing environmental situation associated with the BP elements. In the prevailing research works, the dynamic context information was generated from different sources like sensor devices, various mobile devices or deduced from the different kinds of communication pattern of the users using several applications. Also, present research works collect data from browsing users' profiles in social media. Those processes of gathering and processing the ever-changing context data are both time-consuming and complex work. Additionally, for making a BP re-configurable, those approaches are unable to give a clear conception of the proper inclusion process of the dynamic context data into a BP. Furthermore, those approaches barely give the proper methodologies for deriving the re-configurable BP driven by contexts from a particular goal, specified by a group of stakeholders of a business organization. Consequently, there exists several research questions to be solved like, (i) In what ways can the dynamic context data be assimilated within a BP to facilitate them more responsive to its surrounding environment and change its behaviour in the same way? (ii) From the provided goal, what is the process by which a context-guided re-configurable BP can be derived so that the BP can satisfy both the earlier goal and the newly updated goal at the same time.

A novel and unique approach is proposed in this paper to settle the above challenges. The SCORE [4, 5] framework contains a number of weakly associated layers like essence, role, object, context, and subject. Among which in the subject layer the hierarchical structure of goal and its corresponding sub-goal sets are outlined by the help of a feature graph. This feature graph is the most abstract and also the most expressive method to represent the associations between goal and sub-goal. The principle and secondary contexts set associated with the sub-goals set is contained in the context layer. All the obligatory information to describe the condition and an entity's surrounding environment, involved in the fulfillment of a sub-goal is considered as the primary context, while the secondary context provides the additional knowledge to define the background of that entity in more detail. The object layer comprises the set of services that is driven by the set of primary contexts that belong to the context layer. These context-driven services and related BPs are the main constructs of the object layer. The complete set of the context-driven services, obtained in the object layer is segmented into distinct sections based on the associated roles that are responsible for the execution of a particular set of services.

In the proposed method, the sub-goals which are needed to be reformed are identified first. In the next step, the set of primary contexts, secondary contexts and those context-driven services are to be recognized for each of the sub-goal. Common sequence flows belong to the earlier BP (BP_1), related to the earlier sub-goal and the new BP (BP'), corresponding to the newly modified sub-goal are identified. In the subsequent step, the preceding and succeeding BP element pairs of those sequence flows are merged which are common. Finally, the equivalent sequence flows of *BP1* are modified accordingly. Therefore, a re-configurable BP guided by contexts is proficient in successfully reflecting any sorts of changes in its associated sub-goal. Consequently, the BP can quickly respond to the rapid and constantly changing business environment in which it operates, by the proper inclusion of the ever-changing context data that describes the surrounding business environment. The primary part of this study has been reported in [6]. And that is now made richer and accomplished by the successful inclusion of three software metrics for the measurement of the amount of changes needed to be done in the earlier BP. Moreover, the previous version is now enriched through the addition of three real-life case studies and the completion of a detailed comparative study of this approach with the other earlier approaches.

2. Related Works

In [7], a goal-driven method with the consideration of the context-aware approach to provide the set of suitable services to its end users has been proposed. The goal tree is used for the representation of the goal of the users more systematically and uniquely. The system architecture comprises three important layers like (i) service layer, (ii) Context Reasoner Layer, and (iii) runner and planner layer. All types of contexts data (Physical context, User Information and computing context) and all related services are stored by the service layer, whereas, all reasoning-related modules are contained in the reasoned context layer. Planner and Runner Layer are designed and implemented to constitute the proper service sequence path dynamically and invoke those services depending on the present context values. The planner module contains four components like goal loader, goaltree builder, goaltree optimizer and service path builder whereas, the runner module includes four other components, such as, runner manager, service path execution unit, web service handler and runner adapters. The presented architecture is validated through a series of studies on a service-based medical diagnostic system using the CASC (Context-Aware service composition) system which offers a proper service sequence plan to be executed depending on the existing context data.

In [8], how the service discovery and the service composition processes are guided by the context information is demonstrated in detail. Three types of context information like location, budget and time are taken into account for the determination of proper service discovery and service composition process. An ontology has been constructed to represent the context information with the help of a unique constraint model. The two types of constraints introduced are hard and soft constraint. Hard constraints are compulsory conditions to be fulfilled, whereas, soft constraints only enforce a penalty value, ranging from zero to one, on the pre-assigned condition. Moreover, they have proposed a method for discovering and composing the non-electronic services, by assigning the pre-conditions (modelled as contexts) and the post-conditions (modelled as constraints) of the services. Finally, a service composition algorithm has been proposed by using the branch and bound algorithm technique. The entire approach is illustrated through a case study on an entertainment planner.

A scalable software architecture is proposed in [9] to manage a massive amount of heterogeneous data collected from diverse platforms. This architecture comprises three functional components such as, (a) Distributed complex event integrated system for event detection, (b) module dedicated to derive contextual knowledge and correlate them with identified events which are obtained from the event detection module, (c) a notification module for giving alert information. CEP (Complex Event Processing) [10] engine is used for tracking and analyzing the huge streaming data in real-time. The SEDA-SOA [9] model includes seven modules like, (i) Push and Pull module, (ii) Subscribers Registration module, (iii) Complex Event Processing module, (iv) Context Detection module, (v) Distributed Database System, (vi) Notification module and (vii) Subscribers. In addition, an algorithm is also proposed to automate the overall flow of the SEDA-SOA system. Finally, a real-life case study on the health care system considering how the heath of the citizens is affected by air pollution is demonstrated in detail for the validation of the proposed context-aware architecture.

An event-driven SOA which is aware of the contexts is the prime focus of the paper [11]. To achieve the purpose three types of contexts are considered here. These are (i) location context (the living place, working place or the current place where the user is present at that moment), (ii) physical context (speed, noise level, weather etc.) and (iii) personal context (any kind of disease, weight, blood pressure, age etc.). The proposed mechanism, named CARED-SOA, encompasses several significant steps like (i) acquisition of a huge amount of data which is heterogeneous in nature from the IoT devices, (ii) analysis and processing of those streaming data using the CEP engine for better decision making and (iii) the provision of REST services those are both context-aware as well as the lightweight services. CARED-SOA components are (i) Enterprise Service Bus, (ii) Data Transformation, (iii) Receiving Module, (iv) Domain DB (v) Domain REST Web Service, (v) CEP engine, (vi) Context Broker, (vii) Context DB, (vii) Context and Notification REST Web Service, (viii) Complex Event Notification Message Broker, (ix) Domain data, (x) Iot and (xi) Mobile App. The Context Broker contains three central modules such as, (a) Context-Based Adviser, (b) Context Reasoner and(c) Context Knowledge Manager. In [12], A framework named CAS-Mine is proposed to investigate the associations between the context data and the invoked services of users by deducing the common association rules. The current context is considered to shape service provisioning. A new algorithm called GENIO is presented to find associations rules. In [13], a model-driven approach termed PerCAS gives the provision of the inclusion of the web service to user personal context preferences. It can dynamically alter the logic of context-awareness based on users' demand.

A context-aware ESB is proposed in [14]. This approach provides the service orchestration process by selecting the closest services and performing the service composition depending on the location of the client. The graph-based method is aware of the contexts and it designs the processes along with their location information of the users. The vertices of the graph represent the main services while the edges denote the interconnection between the services. The services are identified efficiently by this approach and the location-wise choreography of services are unveiled dynamically. In [15], several distinct context modelling such as Markup Scheme Models, Key-Value Models, Graphical Models, Spatial context model, Ontology-Based Models, Object-Oriented Models and Logic-Based Models techniques are surveyed in detail. The need of the context-aware research, challenges involved in this and the future directions are also discussed. In [16], a context model framework is developed based on different abstraction level of ontology such as Domain-specific Ontology and Meta Ontology level. Moreover, an algorithm is established for the conversion of the context model to the knowledge with the help of the reasoning tool. Enhancement of the context model re-usability and reasoning atomization is the prime focuses of this paper. A hybrid spatial model is introduced in [17] for the navigation systems which is context-aware. Here, spatial model requirements are categorized into efficiency-related and service-oriented requirements. An ontology-based context-aware approach is proposed to solve issues of uncertain models with automatic reasoning. The model is divided into domain-specific and meta levels following a hierarchy. The basic context knowledge is extracted by the meta ontology while the domain knowledge is obtained from the domain-specific model. Finally, an algorithm is designed to get the knowledge from the context model. In [18], an ontology-based framework is designed, implemented and validated by taking into account a real-life case study for the construction of NFC-based context-aware applications. In this proposed framework the development overhead of the context data collection, aggregation and storing processes are hugely abridged as compared to the previous context-aware frameworks.

A prototype of a generic probabilistic framework is developed to provide the context-aware scalable authentication in [19]. A framework based on ontology is proposed to develop context-aware results triggered by user contexts. The framework produces service security for NFC-based data integration. An adaptive access control framework is proposed in [20] for wireless area network and for measuring medical data in the body. This framework can deal with unexpected situations and provide superior security decisions by the inclusion of user behaviour into this model. In [21], a system is introduced to provide better user authentication through a touch-based identity protection service for collecting contextual data. Two types of features like contextual and behavioural features are included in the system to provide 90% accuracy for usability and security aspects in real-life data. A layered conceptual framework is developed in [22] to describe the communal elements in the most context-driven architecture. A detailed survey on the existing context-driven applications and context-aware framework is also performed in the paper. In [23], a detailed survey is done on different types of procedures and also the service engineering considering the context-aware solutions. A set of solutions that was not discussed in the pre-existing research works is also proposed depending on three major categories: message interception methods, source code level and model-driven handling. Finally, a series of studies has been done to describe the method and compared to each other to exhibit the strength and the weakness of those three categories depending on several features. An exhaustive survey on the existing context-driven methods for mobile platforms is accomplished in [2, 24]. Moreover, a rule-based context reasoning platform is developed and validated through the use cases in the paper.

Above pre-existing works are less efficient to provide a proper method to include dynamic context data within an existing BP so that it can obtain re-configurability. Additionally, those works hardly describe how a re-configurable BP which is driven by contexts can be effectively derived from a specified business goal.

3. Proposed Methodology

The context-driven services are becoming the fundamental requirement to its users in order to efficiently deal with the rapidly-changing surrounding environmental situation of those users. Consequently, the appropriate incorporation of those context-driven services makes an existing BP more relevant, reliable, meaningful and appropriate in nature through the assimilation of the current dynamic context information. Thus, the re-configurable BP becomes capable of satisfying the previous as well as the new sub-goal at the same time. The algorithm described below delivers a unique way to attain re-configurability in BP driven by contexts through the proper insertion of the current context information associated with the BP elements, contained in that BP.

3.1. Algorithm

The Context-Driven Re-configurable Business Process Achievement (CDRBPA) algorithm is given the dynamic context information about the BP elements and a re-configurable BP is attained.

Algorithm 1: Algorithm for CDRBPA

Input: Earlier Goal (*G*), Earlier BP (*BP*), Goal to be achieved (*EG*).

Output: Modified BP (*BP'*);

```
Set SFs[] = NULL; C[]← contexts; R[]← roles; SR[]← services; SG[] ← sub-goals;
PE[]← preceding elements of sequence flows; NE[]← succeeding elements of sequence flows;
begin
    Initialize number of existing goals n
    for i=1 to n do
        Find SGi corresponding to BPi;
        SGi[] ← getSubgoal(Gi);
        for j=1 to m do
            SRij[]←getService(SGij);
            for k=1 to r do
                Rijk[ ]←getRole(SRij);
                Cijk[ ]←getContext(SRij);
            end
        end
        SFs[] ←SFij;
    end
    for i=1 to m do
        SFUi[]←Unique Sequence flows prior to SFs[i];
        for p=1 to s do
            PEi[p ] ←getPE(SFUi[p]);
            NEi[p] ←getNE(SFUi[p]);
        end
    end
    C(PE, NE) ← (PE1 [], NE1 []) ∩ (PE2 [], NE2 []) ∩ … ∩ (PEn [], NEn []);
    Z← C(PE, NE);
    SFUf ← Union SFU1[], SFU2[] …..SFUn[] associated with Z;
    Update SFU1[] related to Z, by SFUf;
end
```

The algorithm starts with the recognition of the sub-goals and its associated BPs that are required to be changed. The succeeding step contains the process of recognition of the mutual sequence flows between the earlier BP and the new BP. Then those common sequence flows are merged together within the newly modified BP (*BP'*) in the last step.

Three inputs like a) the existing BP, b) the existing goal (*G*) and c) the goal to be achieved termed as the Enhanced goal (*EG*) are given into the algorithm and the output is generated as the newly modified BP (*BP'*). In the beginning, a global empty list *SFs[]* is initialized where all sequence flows from the existing BP is stored. Six distinct lists for storing sub-goals (*SG[]*), services (*SR[]*), *roles (R[])*, contexts (*C[]*), preceding entities linked to the sequence flows *(PE[])* and succeeding entities linked to the sequence flows (*NE[]*) are also created.

In the first step, all the sub-goals from the existing goal G are retrieved using a function *getSubgoal()* and stored in the list *SG[].* The second step contains the iterations through all the sub-goals to identify and retrieve all the services, associated with those sub-goals through *getService()* method and store them in the list *SG[].* Then, another for loop is used for the iterations through all the services to retrieve all the roles and all contexts, linked with those services through two methods: *getRole()* and *getService()* and store those results in two separate lists *R[]* and *C[]* respectively. Then, for each sub-goal, all the sequence flows are retrieved and stored in the global list *SFs[].*

In the next step, for each sub-goal, the sequence flows which are unique to that sub-goal (i.e not associated with other sub-goals) are recognized and preserved in a list named *SFU*. For each sequence flow in the *SFU[]* list, all the BP elements which precede (PE) and succeed (NE) that sequence flow are retrieved using *getPE()* and *getNE()* methods and stored in two different lists: *PE[]* and *NE[]* respectively. Having done this, the (*PE*, *NE*) pair which is common to all the sub-goals is recognized and stored in *Z*. The union of all the sequence flows which is associated with *Z* is taken and stored into *SFUf*. Finally, the sequence flow, associated with Z, is replaced by *SFUf* to get the enhanced goal. Thus, earlier BPs corresponding to earlier sub-goals are re-configured and both the new and existing goals are sufficed.

The algorithm, in step 4-13, is required to traverse through three lists, a list of *n* number of goals (*G[]*), a list of *m* number of sub-goals (*SG[]*) and a list of r number of services per sub-goal (*SR[]*). Thus, here, the time complexity can be computed as *O(nmr).* As the number of goals are decided by the stakeholders then the complexity would be *O(mr).* In the similar manner, in steps 16-22, the algorithm is required to traverse through two loops, the outer loop for a list of *m* number of subgoals and the inner loop for a list of sequence flows of interest (*SFU[]*) containing s number of elements. Therefore, the time complexity here is calculated as *O(ms).* Hence, the overall time complexity of the algorithm can be computed as *O(mr)* + *O(ms).* As *s* is always less than r, the Algorithm's time complexity is stated as *O(mr).*

3.2. Re-configurability Metrics

Three primary context (PC)-based metrics like, Degree of re-usability based (D_{RUPC}), Degree of re-appropriation (D_{RAPC}) and Degree of re-configurability (D_{RCPC}) are introduced. D_{RUPC} is used to measure the number of those PC-driven services which can be re-used from the pre-existing BP (*BP1*) to the newly re-configured BP (*BP'*). D_{RAPC} is formulated to measure the amount of re-appropriation required to be done in each of those PC-driven services obtained in D_{RUPC}. D_{RCPC} is introduced for the calculation of the number of newly added PC-driven services in BP'. It also measures the complexity exists for this re-configuration in BP'.

(a) Degree of Re-usability based on PC (D_{RUPC})- This is formulated as the ratio of the number of PC-driven services those can be re-used from *BP1*(*NRUPC*) and the total number of PCs and those PC-driven services belong to *BP'*(*NTPC*)

$$D_{RUPC} = \frac{NRUPC}{NTPC} \quad (1)$$

$D_{RUPC} <= v$, where $v <= 1$ and v is user defined. Stakeholders can change the value of v depending on the current business environment and the customer needs. If the ($D_{RUPC} <= v$) condition is satisfied then only D_{RAPC} and D_{RUPC} are calculated. The set of all PC-driven services, to be re-used is stored in a vector named as, Reusable Vector (V_{RU}).

(b) Degree of Re-appropriation based on PC (D_{RAPC}) - For each element, contained in V_{RU}, D_{RAPC} value is evaluated. In a BP, the services (*Sr*) mean the activities are context-driven and it leads to the fact that the corresponding gateways (*Ga*), sequence flows (*Sq*) and the data (*Da*) are also context-driven.

Measurement of the effect of the modification of the primary contexts on the related *Sr, Ga, Sq* and *Da* elements, contained in the V_{RU}, is performed here. Suppose, a context-driven service is driven by a set of contexts in BP1 and that service can be re-used in the re-configured BP (*BP'*) with the modification in its several contexts. Changes in the primary contexts (PC) affect that context-driven service most and consequently, the associated data and gateways are also influenced to a certain extent whereas, the modification in the secondary contexts (SC) or auxiliary context has less impact on its associated services, gateway and data. As a result, modifications in PC result in the changes in the services of complex or high (*Hi*) level and variations in SC result in the changes in the services of simple or low (*Lw*) level. Similarly, gateways and data are also immensely affected by the changes in PCs and moderately affected by SCs. Hence, certain weightage values should be given to the four types of BP elements (*Sr, Ga, Sq* and *Da*) and their corresponding level of complexity based upon the amount of modification effect on the complete BP flow. Here the services are directly modified by the corresponding PCs and SCs, whereas the gateways and the data sets are affected by those context-driven services. Consequently, the weightage values are assigned as [{(*WvC(Hi(Sr)*))>(*WvC(Lw(Sr)*))} > {(*WvC(Hi(Ga)*))>(*WvC(Lw(Ga)*))}> {(*WvC(Hi(Sq)*))>(*WvC(Lw(Sq)*))} > {(*WvC(Hi(Da)*))>(*WvC(Lw(Da)*))}], where, *WvC(Hi(Sr)*) denotes the weightage value of the services with high level of modifications.

Here, {(x_1>x_2)>(y_1>y_2)> (w_1>w_2) (z_1>z_2)}, where, x_1, x_2, y_1, y_2, w_1, w_2, z_1, z_2 all variables are user-defined, so that the different type of stakeholders can assign the values of those variables based on the present business scenario of an Enterprise Architecture Framework. Thus, a 4-D vector is obtained in this step and it comprises four elements. Each of those four elements is the summation of the weightage values of high- and low-level complexity of the four BP elements *Sr, Ga, Sq* and *Da* respectively.

$$D_{RAPC} = \begin{bmatrix} \Big(WvC\big(Hi(Sr)\big)\Big) + (WvC(Lw(Sr))) \\ \Big(WvC\big(Hi(Ga)\big)\Big) + (WvC(Lw(Ga))) \\ \Big(WvC\big(Hi(Sq)\big)\Big) + (WvC(Lw(Sq))) \\ \Big(WvC\big(Hi(Da)\big)\Big) + (WvC(Lw(Da))) \end{bmatrix} = \begin{bmatrix} (x1 + x2) \\ (y1 + y2) \\ (w1 + w2) \\ (z1 + z2) \end{bmatrix} \tag{2}$$

In Eq 2, the 1st element is the summation of *WvC(Hi(Sr*)) and *WvC(Lw(Sr*)), that denotes the total weightage value of context-driven services of level *Hi(complex)* and *Lw(simple)*. Similarly, the 2nd element is the summation of *WvC(Hi(Ga*)) and *WvC(Lw(Ga*)), that signifies the total weightage value of gateways of level *Hi(complex)* and *Lw(simple).* The third element is the summation of *WvC(Hi(Sq*)) and *WvC(Lw(Sq*)) and the fourth element is the summation of *WvC(Hi(Da*)) and *WvC(Lw(Da*)). Those elements denote the total weightage value of sequence flows of level *Hi(complex)* and *Lw(simple)* as well as the total weightage value of data of level *Hi(complex)* and *Lw(simple)* respectively.

(c) Degree of Re-configurability based on PC (*D_{RCPC}*) - This metric is basically the ratio of the magnitude of two vectors. A calculation is required to measure the number of BP elements (*Sr, Ga, Sq* and *Da*), needed to be generated in the newly generated BP (*BP2*) associated with the new sub-goal. Finally, the newly obtained BP (*BP2*) and the pre-existing BP (*BP1*) are merged together to produce the newly modified BP (*BP'*). A 4-D vector is needed to be generated to calculate the weightage values of the BP elements, contained in *BP2*.

So, the 4-D weightage vector is obtained as

$$WvN = \begin{bmatrix} \Big(WvCN\big(Hi(Sr)\big)\Big) + (WvCN(Lw(Sr))) \\ \Big(WvCN\big(Hi(Ga)\big)\Big) + (WvCN(Lw(Ga))) \\ \Big(WvCN\big(Hi(Sq)\big)\Big) + (WvCN(Lw(Sq))) \\ \Big(WvCN\big(Hi(Da)\big)\Big) + (WvCN(Lw(Da))) \end{bmatrix} = \begin{bmatrix} (a1 + a2) \\ (b1 + b2) \\ (c1 + c2) \\ (d1 + d2) \end{bmatrix} \tag{3}$$

Finally, these two vectors D_{RAPC} and *WvN* are added to get the new 4-D vector denoted as

$$TW = \begin{bmatrix} (x1 + x2) + (a1 + a2) \\ (y1 + y2) + (b1 + b2) \\ (w1 + w2) + (c1 + c2) \\ (z1 + z2) + (d1 + d2) \end{bmatrix} \tag{4}$$

Each of the four elements of *TW* represents the total weightage values of the four types of BP elements like services, gateways, sequence flows and data those are either belongs to *BP1* and re-used in

BP′ or newly generated in *BP2* and re-used in *BP′*. Moreover, a new vector named as *NTWv* is produced to measure the weightage values of those BP elements if the re-configured BP is newly generated in its wholeness. This matrix is represented as follows

$$NTWv = \begin{bmatrix} WvCN(Hi(Sr)) \\ WvCN(Hi(Ga)) \\ WvCN(Hi(Da)) \end{bmatrix} \quad (5)$$

WvCN(Hi(Ga)) denotes the weightage value of the gateways which would be newly generated if *BP′* would be produced in its totality. Thus, D_{RCPC} is formulated as the ratio of the magnitude of the two vectors like, *TW* and the summation of *WvN* and *NTWv*. The expression is like below

$$D_{RCPC} = \frac{|TW|}{|WvN|+|NTWv|} \leq m < 1 \quad (6)$$

where, m is a user-defined variable and the value of it should be less than 1. The value of m can be varied by the stakeholders of an EAF depending on the present context data regarding the business environment and the customer needs. The value of D_{RCPC} increases with the higher values of *TW* and the higher values of *TW* signifies that the modification amount in the four types of elements, belonged to either *BP1* or *BP2* is at a large volume. On the contrary, lower value of D_{RCPC} represents the value of *TW* is also of less. Smaller value of *TW* signifies that the amount of modifications, required to be performed in the pre-existing BP elements in *BP1* and *BP2* is also less. Consequently, the development and the maintenance costs are reduced to a large extent. So, the stakeholder's main focus is to decrease the value of D_{RCPC} as much as possible at the time of satisfying the new goal *G′*. Moreover, the user-defined value of D_{RCPC} gives the privilege to its end-users to assign the value based on the on-going customers' requirements and also on the present business scenario.

3.2.1 Illustration of three metrices

Here, an example has been described to explain the above three software metrics in detail. Suppose, a *BP′* is comprised of seventeen BP elements (5 *Sr*, 2*Ga*, 8 *Sq*, 2 *Da*) within which sixteen elements have been re-used from the pre-existing BP (*BP1*). So, $D_{RU} = \frac{16}{17} =$ 0.941>=*k*, where the value of *k* is assigned as 0.3 by the stakeholders. Suppose, the stakeholders have assigned the values of x_1, x_2, y_1, y_2, w_1, w_2, z_1 and z_2 as 8,7, 6, 5, 4, 3, 2 and 1 respectively.

The sequence flow *Sq3* is changed in *BP1*.Therefore, the associated services like, Sr_3, Ga_2 and Da_2 are supposed to be changed by changing the Primary contexts (level *Hi*). As a result, the associated Primary Context-driven services Sr_3, Sq_3, Ga_2 and Da_2 are also changed in its high (level *Hi*). So, the resulting 4-D vector

$$D_{RAPC} = \begin{bmatrix} (8+0) \\ (6+0) \\ (4+0) \\ (2+0) \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \\ 4 \\ 2 \end{bmatrix}.$$

Suppose, five *Sr,* two *Ga*, eight *Sq* and two *Da* are taken from *BP2* with the modification of complex (*Hi*) level. Thus,

$$\text{WvN} = \begin{bmatrix} (5*8)+0 \\ (2*6)+0 \\ (8*4)+0 \\ (2*2)+0 \end{bmatrix} = \begin{bmatrix} 40 \\ 12 \\ 32 \\ 4 \end{bmatrix}, |WvN| = 52.763.$$

So, the desired 4-D vector TW is attained after the vector addition of D_{RAPC} and WvN

$$\text{TW} = \begin{bmatrix} 8 \\ 6 \\ 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 40 \\ 12 \\ 32 \\ 4 \end{bmatrix} = \begin{bmatrix} 48 \\ 18 \\ 36 \\ 6 \end{bmatrix}.$$

The magnitude of the vector *TW* is $\sqrt{48^2+18^2+36^2+6^2}$= 62.928. If *BP′* would be newly produced entirely then the *NTWv* would be

$$NTWv = \begin{bmatrix} 40 \\ 12 \\ 32 \\ 4 \end{bmatrix}. \ |\text{NTWv}|=52.763.$$

Hence, then $WvN\, and\, |NTWv|$ would be added and the value would be = 52.763+52.763 =105.527.

Thus, $D_{RCPC} = \frac{62.928}{105.527} = 0.596 <= x = 0.75$. Therefore, the BP can be seen as re-configurable.

If the condition (D_{RCPC} <1) is satisfied, then only BPs are re-configurable. The boundary-value of D_{RCPC} is specified by the end-user. Thus, the stakeholders can make variations of the boundary value of D_{RCPC} based on the current business situation and the on-going customer needs of an enterprise architecture framework.

4. Case Studies for the illustration of the Proposed Algorithm

Three detailed case studies are devised to study the impact of contexts and context-driven processes on the re-configurability property of BPs. All case studies are distinct by the number of contexts, context-driven BP elements (like, services, gateways and sequence flows), degree of re-usability, degree of re-configurability and complexity level. One case study named as Hospital Management System is discussed in this section and others (Online Admission System and Online Shopping System) are appended in the Supplementary files.

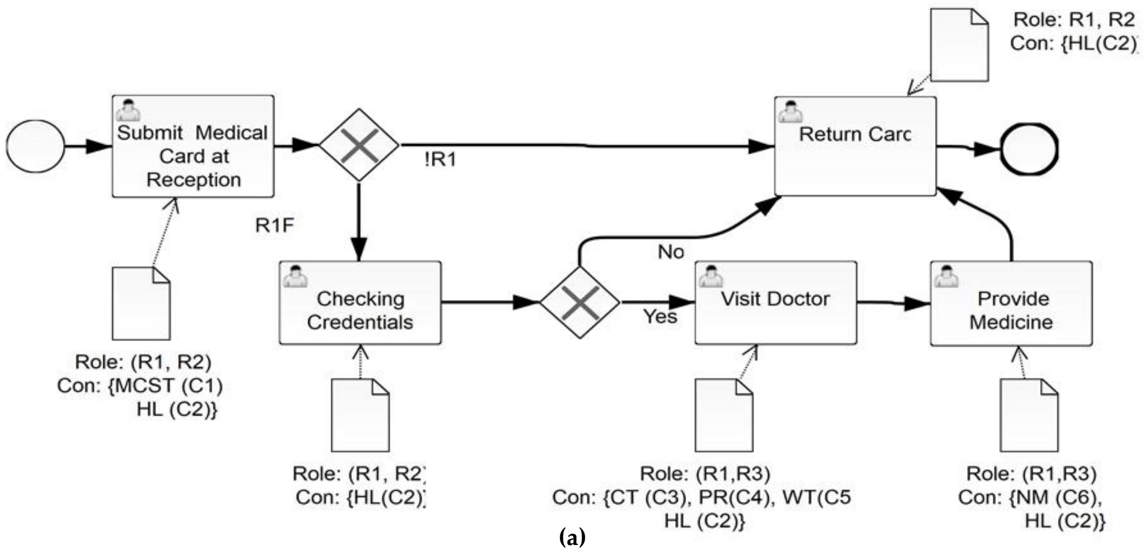
4.1. Hospital Management System

The medical unit of an institution gives the free treatment facilities to its faculties. Later the management group of the institution decided to provide the medical services to its students also.

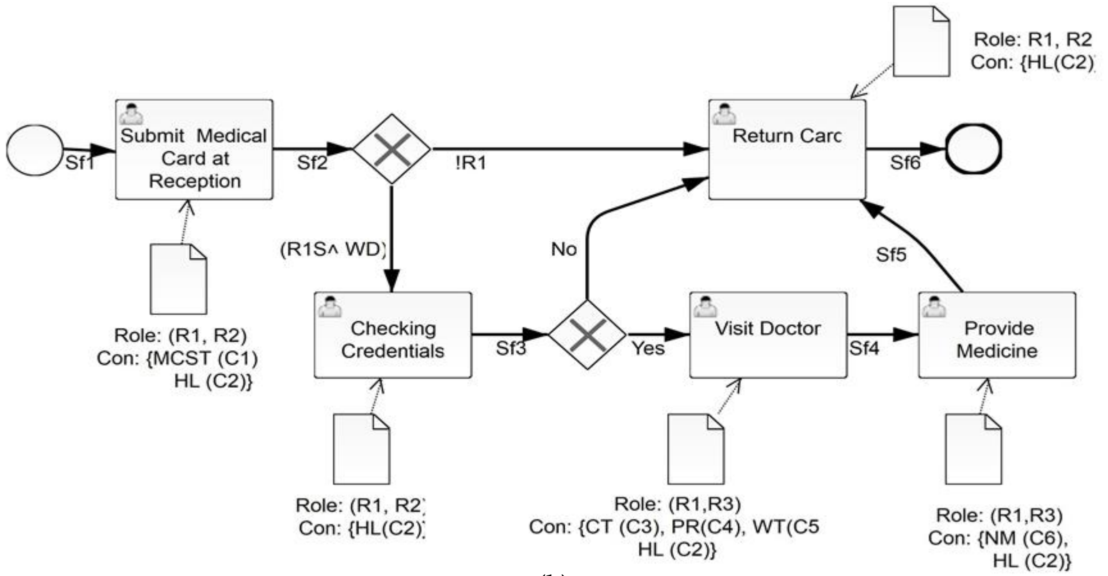
So, the previous goal is to offer free medical service to faculties (*G*) is enriched by offering the free medical services to students also (*EG*). *BP1*, associated with *G* is comprised of five context-driven services like Submit Medical Card at Reception (*Sr1*), Return Card (*Sr2*), Check Credentials (*Sr3*), Visit Doctor (*Sr4*) and Provide Medicines (*Sr5*). Service *Sr1* is realized by the primary context like, Medical Card Submission Time (MCST)-*C1* and the secondary context like, Hospital Location (HL)-*C2*. Similarly, Service *Sr2* is realized by the context *C2* and *Sr3* is realized by *C2*. Service *Sr4* is realized by a set of primary contexts like *C2*, Checking Time (CT)-*C3*, Pressure of Patient (PR)- (*C4*) and Weight of Patient (WT)- *C5* while for service *S5* is realized by *C2* and Number of Medicines (NM)- *C6*. Figure 1(a) represents the BP (*BP1*) containing the Hospital Management System (for Faculty). Figure 1(b) represents the BP (*BP2*) containing the Hospital Management System (for Students). Figure 1(c) represents the BP (*BP'*) containing the Hospital Management System (for Faculty and Student both).

According to the proposed algorithm, for both the BPs like, *BP1* and *BP2* the global list *SFs[1]* and *SFs[2]*, holding all the sequence flows (*SF*), contain the SFs labelled as (($!R_1$), R_1F, Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, No, Yes) and (($!R_1$), ($R_1S \wedge WD$) , Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, No, Yes) respectively. In *BP1*, *WD* represents Working Days. Subsequently, another list, *SFUi[]* for *i*=1 and 2, includes the unique *SFs* which exist in *BPi* but do not exist in *BPj*, where, *i*≠*j*. Hence, $SFU_1[]$ in *BP1* encloses SFs like R_1F. Similarly, $SFU_2[]$ in *BP2* encloses the SF labelled as ($R_1S \wedge WD$).

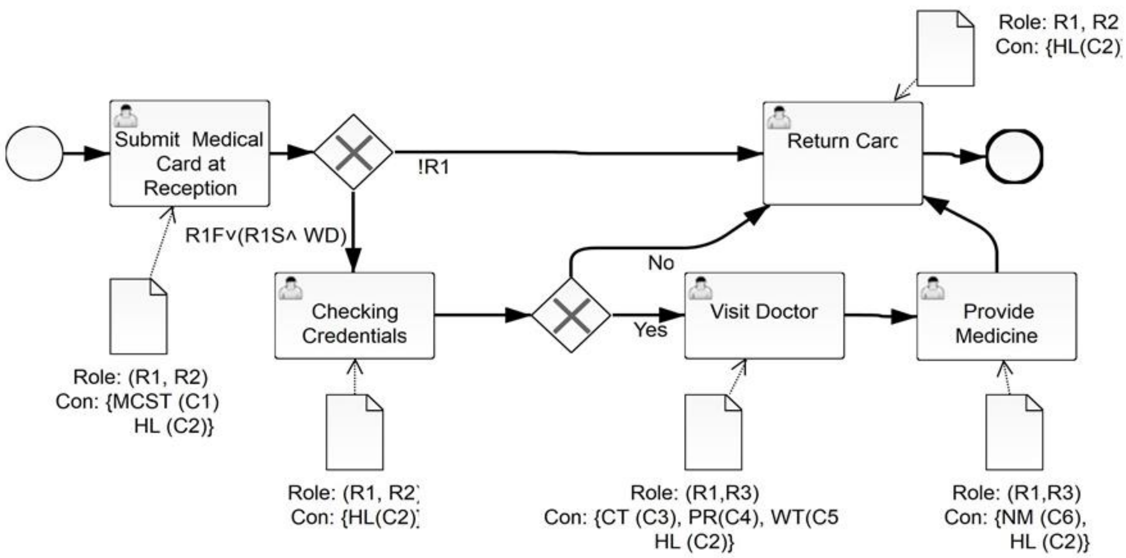
In *BP1*, for, $SFU_1[1]$, *p*=1, i.e., for R_1F sequence flow, the PE such as gateway (Ga_1) is kept in $PE_1[1]$ and the NE named as Check Credentials (Sr_3) is stored in $NE_1[1]$. Similarly, in BP2, for, $SFU_2[1]$, *p*=1, i.e., for ($R_1S \wedge WD$) sequence, the PE expressed as the gateway (Ga_1) is stored in $PE_2[1]$and the NE represented as Check Credentials (Sr_3) is stored in $NE_2[1]$. Common (PE, NE) pair such as (Ga_1, SR_3) is kept in the variable *Z*. The sequence flows associated with (Ga_1, SR_3) in $SFU_1[1]$ and $SFU_2[1]$ are R_1F and ($R_1S \wedge WD$) respectively. Hence, ($R_1F \vee (R_1S \wedge WD)$) is kept in *SFUf*. Finally, all the sequence flows in $SFU_1[1]$ of BP_1 is updated with ($R_1F \vee (R_1S \wedge WD)$) kept in *SFUf*. Consequently, the modified re-configured *BP'* fulfils the earlier sub-goal (SG_1) as well as the new sub-goal (SG_2) which means it can serve both the faculties and the students in the Hospital management system.



(a)



(b)



(c)

Figure 1. (a) Diagram for a Hospital Management System (Faculty) BP, **(b)** Diagram for a Hospital Management System (Student) BP and **(c)** Diagram for a Hospital Management System (Faculty as well as Student both) BP

BP' is comprised of nineteen BP elements (5 *Sr*, 2 *Ga*, 2 *Da*, 10 *Sf*) within which eighteen elements have been re-used from the pre-existing BP (*BP1*). So, $D_{RUPC} = \frac{18}{19}$ = 0.947>=k, where the value of k is assigned as 0.7 by the stakeholders. Suppose, the stakeholders have assigned the values of x_1, x_2, y_1, y_2, w_1, w_2, z_1 and z_2 as 8, 7, 6, 5, 4, 3, 2 and 1 respectively.

In this case study, since *BP1* is re-configurable, *BP'* is generated by including the new sequence flows belonged to *BP2*. The calculation is to measure the changes in *BP1*. Those changes are expressed as

$$D_{RAPC} = \begin{bmatrix} (1*8) \\ (1*6) \\ (1*4) \\ (1*2) \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \\ 4 \\ 2 \end{bmatrix}$$

In $BP2$, five Sr, two Ga, ten Sf and two Da are created with the modification of complex (Hi) level which are re-used in BP'.

So, the $WvN = \begin{bmatrix} (5*8)+0 \\ (2*6)+0 \\ (10*4)+0 \\ (2*2)+0 \end{bmatrix} = \begin{bmatrix} 40 \\ 12 \\ 40 \\ 4 \end{bmatrix}$.

The magnitude of the vector WvN is $\sqrt{40^2 + 12^2 + 40^2 + 4^2}$= 57.965. If $BP1$ could not be re-configured, then BP' would be generated in its entirety. Then the $NTWv$ would be

$$NTWv = \begin{bmatrix} (5*8) \\ (2*6) \\ (10*4) \\ (2*2) \end{bmatrix} = \begin{bmatrix} (40) \\ (12) \\ (40) \\ (4) \end{bmatrix}.$$

Now, $|NTWv| = \sqrt{40^2 + 12^2 + 40^2 + 4^2}$=57.965. Hence, $|WvN| and |NTWv|$ would be added and the value would be (57.965*2) = 115.93 in case *BP1* would not be re-configured. But, here *BP1* is re-configured. So the total amount of modifications to be done in the process of re-configuration is

$$TW = (D_{RAPC} + WvN) = \begin{bmatrix} 8 \\ 6 \\ 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 40 \\ 12 \\ 40 \\ 4 \end{bmatrix} = \begin{bmatrix} 48 \\ 18 \\ 44 \\ 6 \end{bmatrix}.$$

Hence, $|TW| = \sqrt{48^2 + 18^2 + 44^2 + 6^2}$ = 67.823. Therefore,

$D_{RCPC} = \frac{|TW|}{|WvN| + |NTWv||} = \frac{67.823}{115.93}$ =0.585<= x =0.75. Thus, the *BP* is re-configurable.

4.2. Online Admission System

An institution provides the Courses for undergraduate (UG) Students. Later the management group of the institute decides to offer various courses to the postgraduate (PG) students also. So, the goal (*G*) of giving education remains the same, only the new sub-goal (SG_2) of providing courses to PG students is added to the previous sub-goal (SG_1) of offering the courses to the UG students. The previous BP (BP_1) associated with SG_1 comprises seven primary context-driven services like Student Registration (Sr_1), Check Age (Sr_2), Not Eligible (Sr_3), Return Home Page (Sr_4), Eligible for Admission (Sr_5), Check Marks (Sr_6) and Get Enrolled (Sr_7). Figure 2(a) represents the BP (*BP1*) containing the Online Admission System (for UG students), Figure 2(b) represents the BP (*BP2*) containing the Online Admission System (for PG students) and Figure 2(c) represents the BP (*BP'*) containing the Online Admission System (for UG and PG students both).

According to the proposed algorithm, for both the BPs like, *BP1* and *BP2* the global lists *SFs[1]* and *SFs[2]*, holding all the sequence flows (*SF*), contain the SFs labeled as {Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, Sf_7, Sf_8, Invalid, (18<=age<=21), ((18<=age<=21)∧(marks>=80%)), Not in Range} and {Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, Sf_7, Sf_8, Invalid, (21<age<=25), Not in Range, ((21<age<=25)∧(marks>=60%))} respectively. Subsequently, another list, *SFUi[]* for *i*=1 and 2, stores the unique *SFs* which exist in *BPi* but do not exist in *BPj*, where, *i*≠*j*. Hence, $SFU_1[]$ in *BP1* contains SFs like (18<=age<=21) and ((18<=age<=21)∧(marks>=80%)). Similarly, $SFU_2[]$ in *BP2* encloses the SFs labelled as (21<age<=25) and ((21<age<=25)∧(marks>=60%)).

In BP1, for, $SFU_1[1]$, *p*=1, marked as (18<=age<=21), the PE i.e., gateway (Ga_1) is kept in $PE_1[1]$ and the NE named as Eligible for Admission (Sr_5) is stored in $NE_1[1]$. In the same way, for, $SFU_1[2]$, *p*=2, for

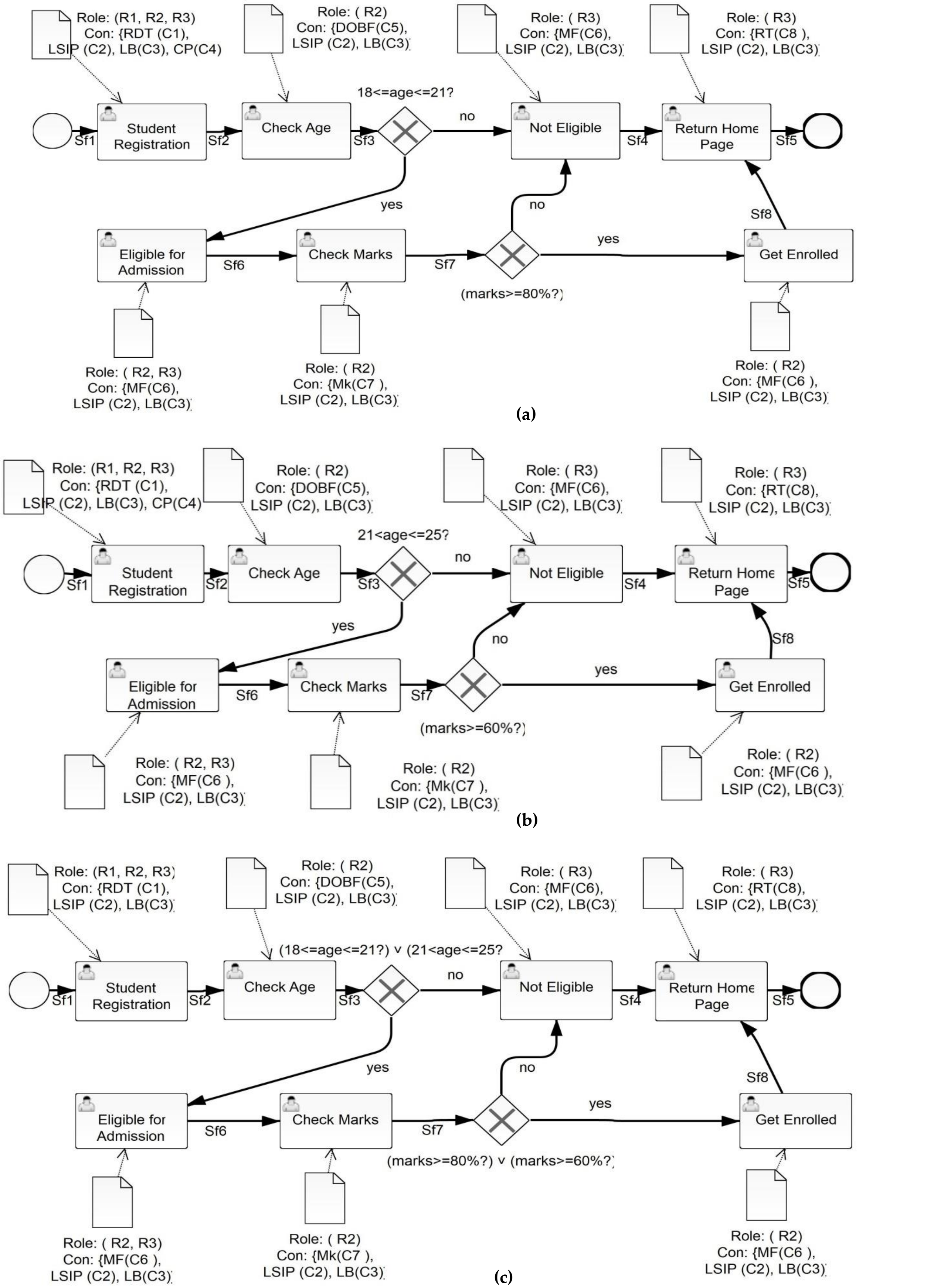


Figure 2. (a) Diagram for an Online Admission System (UG students) Business process, **(b)** Diagram for an Online Admission System (PG students) Business process and **(c)** Diagram for an Online Admission System (UG and PG both students) Business process

((18<=age<=21)∧(marks>=80%)), the *PE* i.e., gateway (Ga_2) is kept in $PE_1[2]$and *NE* like Get Enrolled (Sr_7) is stored in $NE_1[2]$. Equally, in *BP2*, for, $SFU_2[1]$, *p*=1, marked as (21<age<=25), *PE* expressed as the gateway(Ga_1) is stored in $PE_2[1]$and the *NE* represented as Eligible for Admission (Sr_5) is stored in $NE_2[1]$. For, $SFU_1[2]$, *p*=2, for ((21<age<=25)∧(marks>=60%)), gateway (Ga_2) is kept in $PE_1[2]$and NE like Get Enrolled (Sr_7) is kept in $NE_1[2]$. Common (*PE*, *NE*) pairs like (Ga_1, SR_5) and (Ga_2, SR_7) are kept in the variable *Z*.

The sequence flows associated with (Ga_1, SR_5) in $SFU_1[1]$ and $SFU_2[1]$ are (18<=age<=21) and (21<age<=25) respectively. The other sequence flows attached to (Ga_2, SR_7) in $SFU_1[2]$ and $SFU_2[2]$ are ((18<=age<=21)∧(marks>=80%)) and ((21<age<=25)∧(marks>=60%)) respectively. Hence, ((18<=age<=21) ∨ (21<age<=25)) and (((18<=age<=21)∧(marks>=80%))∨((21<age<=25)∧(marks>=60%))) are kept in *SFUf*. Finally, the sequence flows, contained in $SFU_1[1]$ and $SFU_1[2]$ in *BP1* are modified by ((18<=age<=21) ∨ (21<age<=25)) and (((18<=age<=21)∧(marks>=80%))∨((21<age<=25)∧(marks>=60%))), kept in *SFUf*. Consequently, the modified *BP′* fulfils the prior sub-goal (SG_1) and the modified sub-goal (SG_2) means, Online Admission Process for UG students as well as for PG students concurrently.

BP′ is comprised of twenty-three BP elements (7 *Sr*, 2 *Ga*, 2 *Da*, 12 *Sf*) within which twenty-two elements have been re-used from the pre-existing BP (*BP1*). So, $D_{RUPC} = \frac{22}{23} = 0.956>=k$, where the value of *k* is assigned as 0.7 by the stakeholders. Suppose, the stakeholders have assigned the values of x_1, x_2, y_1, y_2, w_1, w_2, z_1 and z_2 as 8, 7, 6, 5, 4, 3, 2 and1 respectively.

In this case study, since *BP1* is re-configurable, *BP′* is generated with the modification of *BP1* by the inclusion of only the new sequence flows belonged to *BP2*. So, the calculation is required, to measure the amount of changes in *BP1* for the inclusion of the new sequence flows. The changes are expressed as

$$D_{RAPC}=\begin{bmatrix}(1*8)\\(1*6)\\(1*4)\\(1*2)\end{bmatrix}=\begin{bmatrix}8\\6\\4\\2\end{bmatrix}$$

In *BP2*, seven *Sr*, two *Ga*, twelve *Sf* and two *Da* are created with the modification of complex (*Hi*) level and those BP elements are re-used in *BP′*.

$$\text{So, the } WvN=\begin{bmatrix}(7*8)+0\\(2*6)+0\\(12*4)+0\\(2*2)+0\end{bmatrix}=\begin{bmatrix}56\\12\\48\\4\end{bmatrix}.$$

The magnitude of the vector *WvN* is $\sqrt{56^2+12^2+48^2+4^2}$=74.833. If *BP1* could not be re-configured, then *BP′* would be generated in its entirety. Then the *NTWv* would be

$$NTWv=\begin{bmatrix}(7*8)\\(2*6)\\(12*4)\\(2*2)\end{bmatrix}=\begin{bmatrix}(56)\\(12)\\(48)\\(4)\end{bmatrix}.$$

Now, $|NTWv| = \sqrt{56^2+12^2+48^2+4^2} = 74.833$. Hence, $|WvN| and |NTWv|$ would be added and the value would be (74.833*2) = 149.666 in case of if *BP1* would not be re-configured. But, here *BP1* is re-configured. So the total amount of modifications to be done in the process of re-configuration is

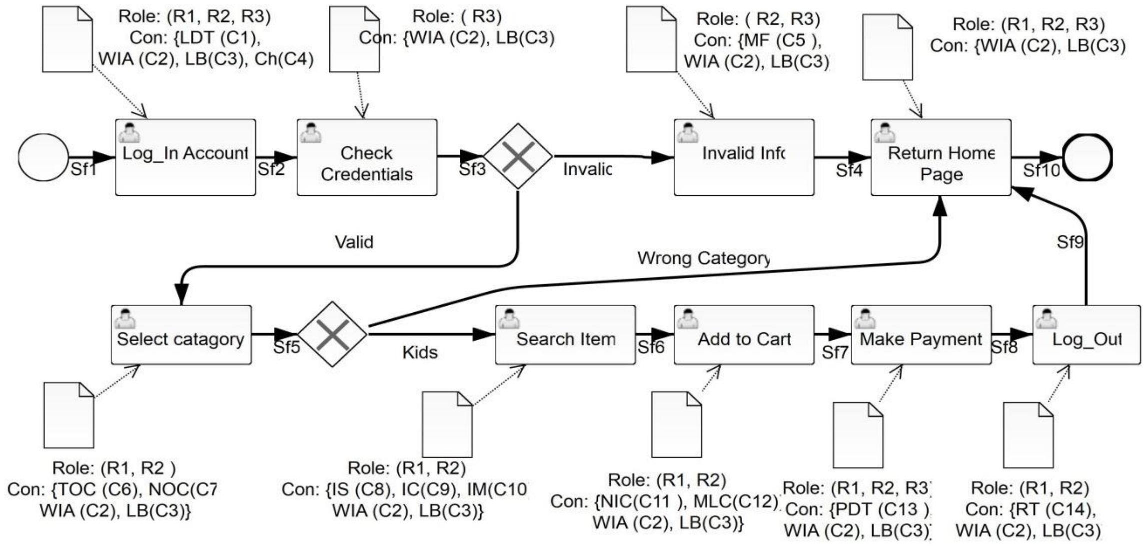
$$TW=(D_{RAPC}+WvN)=\begin{bmatrix}8\\6\\4\\2\end{bmatrix}+\begin{bmatrix}56\\12\\48\\4\end{bmatrix}=\begin{bmatrix}64\\18\\52\\6\end{bmatrix}.$$

Hence, $|TW|=\sqrt{64^2+18^2+52^2+6^2}$=84.616. Therefore, $D_{RCPC}=\frac{|TW|}{|WvN|+|NTWv|}=\frac{84.616}{149.666}$=0.565<=*x*=0.75

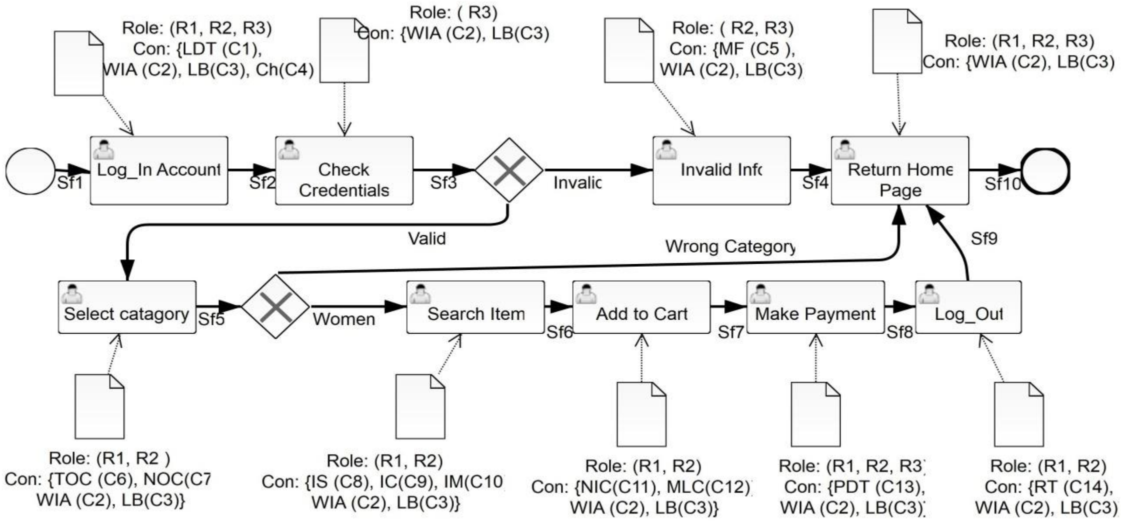
Thus, the BP is re-configurable.

4.3. Online Shopping System

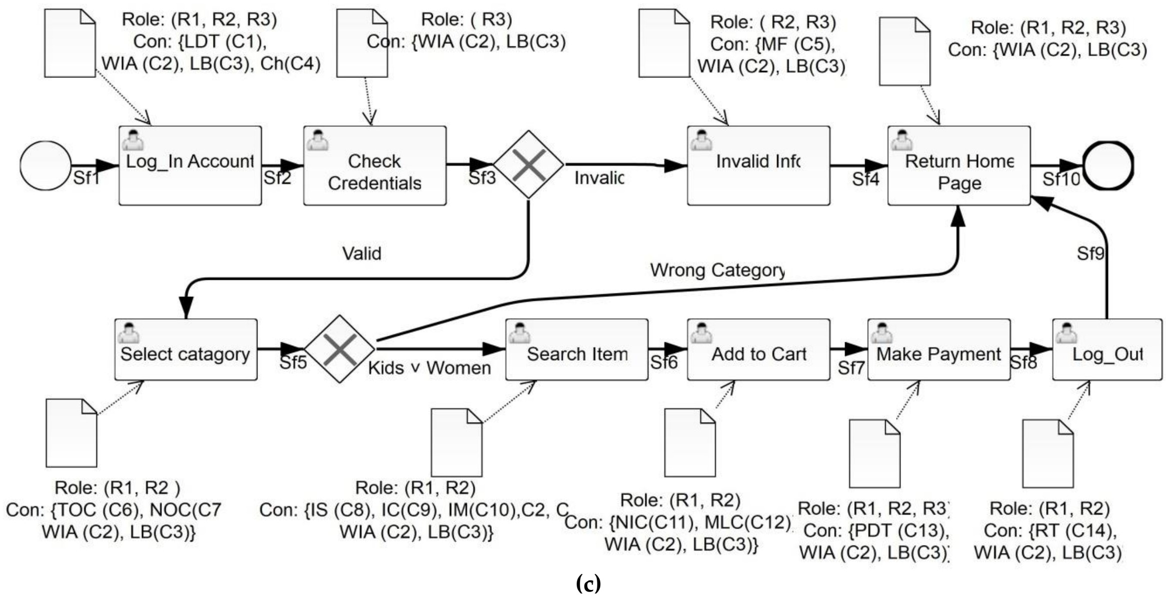
An online shopping site delivers the dresses for only Kids. Later the management group of that shopping site decides to open a new section for selling the dresses for women also. So, the goal (*G*) of providing dresses to the customers remains the same, only the new sub-goal (SG_2) of selling dresses for women section is added to the previous sub-goal (SG_1) of selling various dresses for the kids only. The previous BP (BP_1) associated with SG_1 comprises eight primary context-driven services like Login Account (Sr_1), Invalid Information (Sr_2), Returning to the Homepage (Sr_3), Select Category (Sr_4), Search Item (*Sr5*),



(a)



(b)



(c)

Figure 3. (a) Diagram for an Online Shopping System (Kids only) Business process, **(b)** Diagram for an Online Shopping System (Women only) Business process and **(c)** Diagram for an Online Shopping System (Kids as well as Women Section) Business process

Add To Cart(Sr_6), Make Payment (Sr_7) and Log Out (Sr_8). Figure 3(a) represents the BP (*BP1*) containing the Online Shopping System (for Kids only), Figure 3(b) represents the BP (*BP2*) containing the Online Shopping System (for Women only), and Figure 3(c) deals with the BP (*BP′*) containing the Online Shopping System (for Kids and Women both).

According to the proposed algorithm, for both *BP1* and *BP2* the global lists *SFs[1]* and *SFs[2]*, holding all the sequence flows (SF), contain the SFs labelled as (Valid, Invalid, Wrong Category, Kids, Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, Sf_7, Sf_8, Sf_9, Sf_{10}) and (Valid, Invalid, Wrong Category, Women, Sf_1, Sf_2, Sf_3, Sf_4, Sf_5, Sf_6, Sf_7, Sf_8, Sf_9, Sf_{10}) respectively. Subsequently, another list, *SFUi[]* for *i*=1 and 2, includes the SFs of interest. Therefore, $SFU_1[]$ in *BP1* encloses SFs like Kids. Similarly, $SFU_2[]$ in *BP2* encloses the *SF* labelled as Women.

In *BP1*, for, $SFU_1[1]$, *p*=1, the PE i.e., gateway (Ga_2) is kept in $PE_1[1]$ and the NE named as Search Item (Sr_5) is kept in $NE_1[1]$. In the same manner, in *BP2*, for, $SFU_2[1]$, *p*=1, PE expressed as the gateway (Ga_2) is stored in $PE_2[1]$and the NE represented as Search Item (Sr_5) is stored in $NE_2[1]$. Common (*PE, NE*) pair like (Ga_2, SR_5) is kept in the variable Z. The sequence flows attached to (Ga_2, SR_5) in $SFU_1[1]$ and $SFU_2[1]$ are Kids and Women respectively. Hence, (Kids ∨ Women) is kept in *SFUf*. Sequence flows in $SFU_1[1]$ of *BP1* is then updated with (Kids ∨ Women) kept in *SFUf*. Consequently, the modified *BP′* fulfils the prior sub-goal (SG_1) and the new sub-goal (SG_2) means, Online Shopping system for Kids Section as well as for Women section concurrently.

BP′ is comprised of twenty-seven BP elements (9 *Sr*, 2 *Ga*, 2 *Da*, 14 *Sf*) within which twenty-six elements have been re-used from the pre-existing BP (*BP1*). So, $D_{RUPC} = \frac{26}{27} = 0.962 >= k$, where the value of *k* is assigned as 0.7 by the stakeholders. Suppose, the stakeholders have assigned the values of x_1, x_2, y_1, y_2, w_1, w_2, z_1 and z_2 as 8, 7, 6, 5, 4, 3, 2 and 1 respectively.

In this case study, since *BP1* is re-configurable, *BP′* is generated with the modification of *BP1* by inclusion of only the new sequence flows belong to *BP2*. So, the calculation is required, to measure the amount of changes, in *BP1* for the inclusion of the new sequence flows is expressed as

$$D_{RAPC}=\begin{bmatrix}(1*8)\\(1*6)\\(1*4)\\(1*2)\end{bmatrix}=\begin{bmatrix}8\\6\\4\\2\end{bmatrix}.$$

In *BP2*, five *Sr*, two *Ga*, ten *Sf* and two *Da* are created with the modification of complex (*Hi*) level and those BP elements are re-used in *BP′*.

$$\text{So, the } WvN=\begin{bmatrix}(9*8)+0\\(2*6)+0\\(14*4)+0\\(2*2)+0\end{bmatrix}=\begin{bmatrix}72\\12\\56\\4\end{bmatrix}.$$

The magnitude of the vector *WvN* is $\sqrt{72^2+12^2+56^2+4^2}$= 92.086. If *BP1* could not be re-configured, then *BP′* would be generated in its entirety. Then the *NTWv* would be

$$NTWv=\begin{bmatrix}(9*8)\\(2*6)\\(14*4)\\(2*2)\end{bmatrix}=\begin{bmatrix}(72)\\(12)\\(56)\\(4)\end{bmatrix}.$$

Now, $|NTWv| = \sqrt{72^2+12^2+56^2+4^2}$= 92.086. Hence, $|WvN| and |NTWv|$ would be added and the value would be (92.086*2) = 184.173, if *BP1* would not be re-configured. But, here *BP1* is re-configured. So the total amount of modifications to be done in the process of re-configuration is

$$TW=(D_{RAPC}+WvN)=\begin{bmatrix}8\\6\\4\\2\end{bmatrix}+\begin{bmatrix}72\\12\\56\\4\end{bmatrix}=\begin{bmatrix}80\\18\\60\\6\end{bmatrix}.$$

Hence, $|TW|= \sqrt{80^2+18^2+60^2+6^2}$=101.784. Therefore, $D_{RCPC}=\frac{|TW|}{|WvN|+|NTWv||}=\frac{101.784}{184.173}$=0.552<=$x$=0.75. Thus, the *BP* is re-configurable.

5. Experimental Result and Analysis

Here, a detailed analysis of the algorithm is performed using the explained case studies. Variations of the D_{RCPC}, D_{RUPC} values and the total execution time (*Et)* taken by this proposed algorithm to re-configure those BPs with the different contexts (*Nc*) and context-driven services within the BPs have been discussed here. Certain benchmark has been set for this experiment, like D_{RCPC} and D_{RUPC} values should be less than 1. Three distinct case studies having different contexts and context-driven services have been discussed in detail to illustrate the proposed algorithm named as CDRBPA.

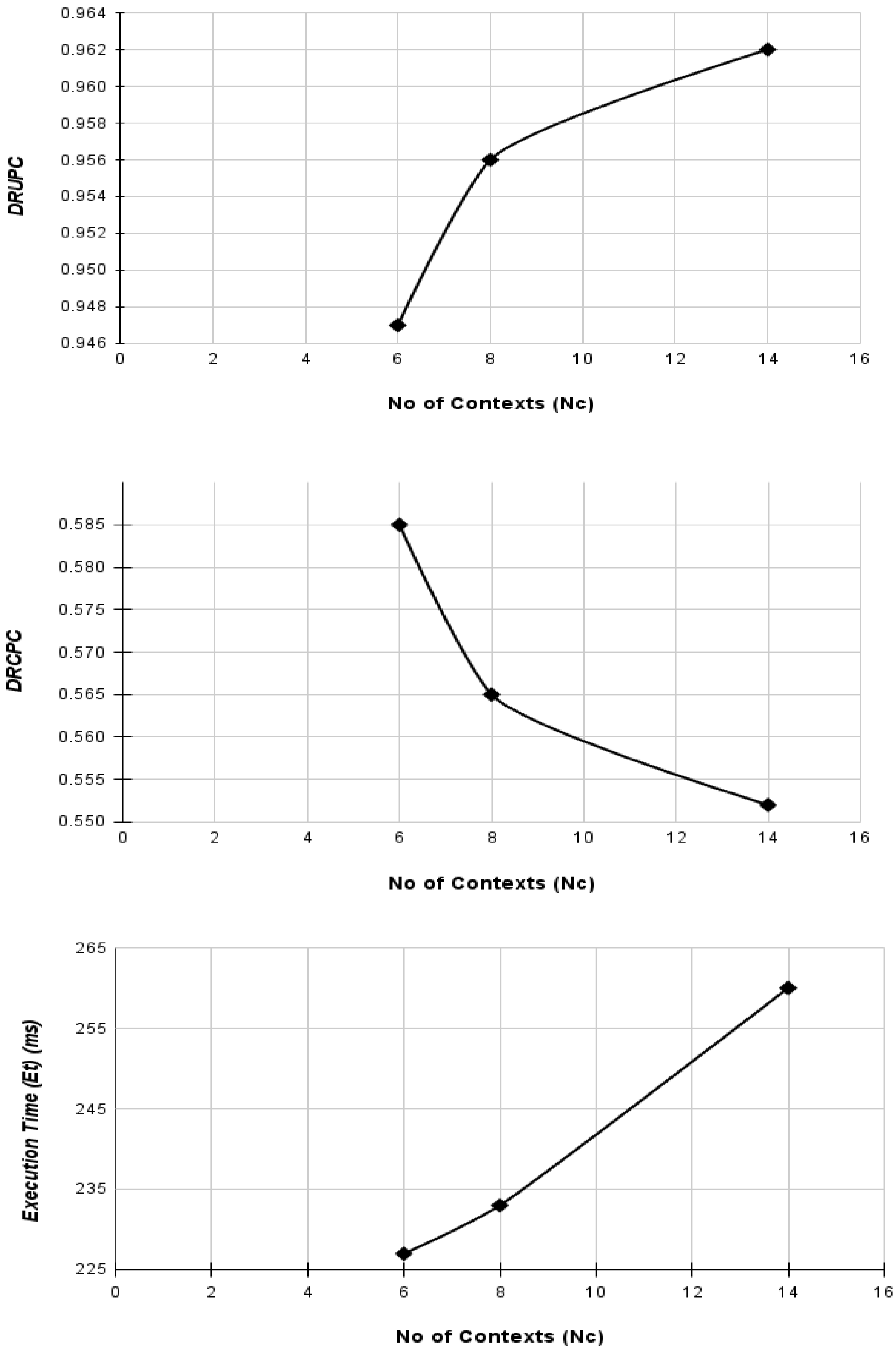


Figure 4. (a) Graphical Representation of D_{RUPC} Versus *Nc* for the three Case Studies, **(b)** Graphical Representation of D_{RCPC} versus *Nc* for Three Case Studies **(c)** Graphical Representation of Execution Time *Et* (in ms) Versus *Nc* for Three Case Studies

The implementation of the algorithm has been done on the Java Eclipse platform using Document Object Model (DOM) [25] and Parser (used to parse XML document) [26]. Camunda BPMN Modeler[1] has been used for creating input files with BPMN diagrams.

Table 1. D_{RUPC}, D_{RCPC}, *Nc* and *Et* values for Three Distinct Case Studies

Case Studies	D_{RUPC}	D_{RCPC}	No of Contexts(*Nc*)	No of Context-driven services	No of context-driven Gateways	Execution Time(*Et*)(ms)
Hospital	0.947	0.585	6	5	2	227
Admission	0.956	0.565	8	7	2	233
Shopping	0.962	0.552	14	9	2	260

The above table (Table 1) exhibits how the D_{RUPC}, D_{RCPC} and the corresponding *Et* values are changed with the different number of Contexts (*Nc*), Context-driven services and context-driven Gateways within a BP for three distinct case studies. Those three real-life case studies deal with the BPs containing the distinct number of BP elements, different number of primary contexts and various levels of complexity. In Table 1, D_{RCPC} value varies from 0.552 to 0.585, which satisfies the condition, mentioned in Section 3.2. Hence, the D_{RCPC} value implies that each BP is re-configurable.

The greater value of D_{RUPC} indicates higher usability of the pre-existing BP (*BP1*). The prior BP can be re-used to serve the newly modified sub-goal and the earlier sub goal together with the minor changes in *BP1*. As shown in the Table 1, D_{RUPC} value increases with the increment of *Nc* values. It signifies that the performance of the algorithm enhances with the higher number of contexts because as the number of contexts and context-driven services increases in *BP1*, most of them can be re-used in *BP'* and thus the percentage of the re-usable context-driven services also increases. Consequently, the requirement of reconfiguration expressed as the degree of re-configurability is reduced with the increment of the *Nc* value. In contrast, the total execution time value (sum of the time to take the two input files, time for re-configuring the first input file and time for producing the output file) of the algorithm increases with the increasing value of *Nc* due to the higher level of complexity to handle the larger number of context-driven services in *BP1*. The variation of D_{RUPC} values, D_{RCPC} values and *Et* values (in ms) with the *Nc* value is shown in the Figure 4(a), 4(b) and 4(c) respectively.

6. Conclusion

The derivation process of a re-configurable BP from a specified goal, driven by contexts and decided by the stakeholders of an organization is described in detail in this paper. A pre-existing BP can be made re-configurable by the suitable integration of the context information, describing the surrounding environmental situation related to the BP elements and those context-driven services within that BP. Thus, those re-configurable BPs are more relevant, meaningful, suitable and flexible in nature. Consequently, the context-driven re-configurability property makes the pre-existing BP capable of satisfying the earlier and the updated sub-goal at the same time without changing its elements rather than only the inclusion of the dynamic context knowledge into the BP. Hence, both the development and maintenance cost are highly minimized through the development of the context-driven re-configurable BP as there is no need of the replacement of the pre-existing BP by a new BP to suffice the newly modified goal.

This unique approach to develop the context-driven re-configurable BP starts with identifying the set of contexts and those context-driven services related to the sub-goal to be modified. Later steps are contained with the finding of the common sequence flows between the previous BP (*BP1*) and the new BP (*BP2*) and their associated previous and next BP element pairs. Finally, the common sequence flows along with their next and previous elements are integrated within the previous BP. In addition, three software metrics like, D_{RUPC}, D_{RAPC} and D_{RCPC} are introduced for the measurement of the amount of changes in the BP elements within the pre-existing BP. Moreover, three real life scenarios like, Hospital Management System, Online Admission System and Online Shopping System are discussed in detail.

The effect of the primary contexts on the BP elements within a BP is taken into account in this work. Though the primary contexts affect the flows of the BP the most, but still the effect of the secondary contexts also cannot be ignored. Hence, the future work will include the impact of the secondary contexts on the BP flows. Another part of the future work will concentrate on improving three Software metrics by

[1] https://camunda.com/bpmn/

including more features of the BP elements like distinct weightage values that may be assigned to different types of gateways based on their amount of modification effects on the BP flow. Finally, the proposed algorithm's performance may be improved by incorporating the secondary context information and the other properties of the BP elements.

The prime objective of the future work will be the improvement of the three proposed primary context based metrics (D_{RUPC}. D_{RAPC}, D_{RCPC}) through the insertion of more properties of a BP. Another enhancement of this work is to assign different weightage values to different gateways having distinct number of branches associated with them. The higher weightage value will be given to the gateways having more number of branches due to their larger impact on the BP flow. In contrast, the lower weightage value will be assigned to the gateways those possess the lower branching paths because of their lesser impact on the flow within a BP.

References

[1] Anind K. Dey, "Understanding and using context", *Personal and Ubiquitous Computing*, Print ISSN: 1617-4909, Online ISSN: 1617-4917, pp. 4-7, Vol. 5, No. 1, 1st February 2001, Published by Springer, DOI: 10.1007/s007790170019, Available: https://link.springer.com/article/10.1007/s007790170019.

[2] Kazem Haki and Maia W. Forte, "Service Oriented Enterprise Architecture Framework", in *Proceedings of the 2010 6th World Congress on Services*, 5-10 July 2010, Miami, FL, USA, Online ISBN: 978-1-4244-8199-6, DOI: 10.1109/SERVICES.2010.39, pp. 391-398, Available: https://ieeexplore.ieee.org/document/5577265.

[3] Maria Marques, Carlos Agostinho, Gregory Zacharewicz and Ricardo J. Goncalves, "Reconfigurable and updatable product-service systems: the path for sustainability and personalization", in *Proceedings of the Symposium on Model driven Approaches for Simulation Engineering*, 23-26 April 2017, VA, USA, Online ISBN: 978-1-5108-3826-0, DOI: 10.22360/springsim.2017.mod4sim.016, pp. 1–12, Published by ACM Digital Library, Available: https://dl.acm.org/doi/10.5555/3108244.3108255.

[4] Priyanka Chakraborty and Anirban Sarkar, "SCORE framework: a layered approach for enterprise architecture", *International Journal of Business and Systems Research*, Print ISSN: 1751-200X, Online ISSN: 1751-2018, pp. 438–467, Vol. 13, No. 4, 22nd May 2019, Published by Inderscience Publishers, DOI: 10.1504/IJBSR.2019.102524, Available: https://www.inderscience.com/offer.php?id=102524.

[5] Priyanka Chakraborty and Anirban Sarkar, "Context Driven Approach for Enterprise Architecture Framework", in *Proceedings of the CISIM 2017: Computer Information Systems and Industrial Management*, 16-18 June 2017, Bialystok, Poland, Online ISBN: 978-3-319-59105-6, DOI: 10.1007/978-3-319-59105-6_24, pp. 277-289, Published by Springer, Available: https://link.springer.com/chapter/10.1007/978-3-319-59105-6_24.

[6] Priyanka Chakraborty and Anirban Sarkar, "Conceptualizing Reconfigurable Business Process: A Context Driven Approach", In *Advanced Computing and Systems for Security*, Singapore: Springer, 2021, Vol. 14, ch. 2, pp 61-74, Print ISBN: 978-981-16-4293-7, Online ISBN: 978-981-16-4294-4, DOI: 10.1007/978-981-16-4294-4_5, Available: https://link.springer.com/chapter/10.1007/978-981-16-4294-4_5.

[7] Lian Yu, Arne Glenstrup, Yang Zhang and Shuang Su, "Goal-driven Context-aware Service Composition", in *Proceedings of the 5th International Conference on Pervasive Computing and Applications*, 1-3 December 2010, Maribor, Slovenia, Online ISBN: 978-1-4244-9144-5, E- ISBN: 978-1-4244-9143-8, DOI: 10.1109/ICPCA.2010.5704124, pp. 342-347, Published by IEEE, Available: https://ieeexplore.ieee.org/document/5704124.

[8] Yifei Zhang, Jianing Wang and Yuhong Yan, "Context-Aware Generic Service Discovery and Service Composition", in *Proceedings of the IEEE International Conference on Mobile Services*, 27 June-2 July 2014, Anchorage, AK, USA, E-ISBN: 978-1-4799-5060-7, DOI: 10.1109/MobServ.2014.27, pp. 132-139, Published by IEEE, Available: https://ieeexplore.ieee.org/document/6924304.

[9] Majid Ayoubi and Sadam Ravichandra, "SEDA-SOA: A Scalable Event-Driven Context-Aware Service Oriented Architecture", in *Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 10-12 July 2018, Bengaluru, India, E-ISBN: 978-1-5386-4430-0, DOI: 10.1109/ICCCNT.2018.8493724, pp. 1-7, Available: https://ieeexplore.ieee.org/document/8493724.

[10] David C. Luckham, *Event processing for business: organizing the real-time enterprise*. New York, USA: Hoboken, John Wiley & Sons Publications, 2012.

[11] Alfonso García D. Prado, Guadalupe Ortiz and Juan Boubeta-Puig, "CARED-SOA: A Context-Aware Event-Driven Service-Oriented Architecture", *IEEE Access*, E-ISSN: 2169-3536, pp. 4646-4663, Vol. 5, No., 8th March 2017, Published by IEEE, DOI: 10.1109/ACCESS.2017.2679338, Available: https://ieeexplore.ieee.org/document/7874075.

[12] Elena Baralis, Luca Cagliero, Tania Cerquitelli, Paolo Garza and Marco Marchetti, "CAS-Mine: providing personalized services in context-aware applications by means of generalized rules", *Knowledge and Information Systems*, Print ISSN: 0219-1377, Online ISSN: 0219-3116, pp. 283–310, Vol. 28, No. 2, 12th November 2010, DOI: 10.1007/s10115-010-0359-z, Available: https://link.springer.com/article/10.1007/s10115-010-0359-z.

[13] Jian Yu, Jun Han, Quan Z. Sheng and Steven O. Gunarso, "PerCAS: An Approach to Enabling Dynamic and Personalized Adaptation for Context-Aware Services", in *Proceedings of the 10th international conference on Service-Oriented Computing*, 12-16 November 2012, Online ISBN: 978-3-642-34321-6, DOI: 10.1007/978-3-642-34321-6_12, pp. 173-190, Available: https://link.springer.com/chapter/10.1007%2F978-3-642-34321-6_12.

[14] Jayeeta Chanda, Sabnam Sengupta, Ananya Kanjilal and Swapan Bhattacharya, "CAESB: Context Aware Enterprise Service Bus", *International Journal of Computer Applications*, Online ISSN: 0975-8887, pp. 1-8, Vol. 30, No. 3, 1st September 2011, Published by Citeseer, DOI: 10.5120/3626-5062, Available: https://www.ijcaonline.org/archives/volume30/number3/3626-5062.

[15] Unai Alegre, Juan Carlos Augusto and Tony Clark, "Engineering context-aware systems and applications: A survey", *Journal of Systems and Software*, Online ISSN: 0164-1212, pp. 55-83, Vol. 117, No. 1, 1st July 2016, Published by Elsevier, DOI: https://doi.org/10.1016/j.jss.2016.02.010, Available: https://www.sciencedirect.com/science/article/pii/S0164121216000467.

[16] Muhammad A. Kabir, "Modeling, Managing and Reasoning about Social Contexts for Socially-Aware Applications", in *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 18-22 March 2013, San Diego, CA, USA, Online ISBN: 978-1-4673-5076-1, E-ISBN: 978-1-4673-5077-8, DOI: 10.1109/PerComW.2013.6529531, Published by IEEE, Available: https://ieeexplore.ieee.org/document/6529531.

[17] Bo Hu, Zhi-Xue Wang and Qing-Chao Dong, "A Novel Context-aware Modeling and Reasoning Method based on OWL", *Journal of Computers*, Online ISSN: 1796-203X, pp. 943-950, Vol. 8, No. 4, April 2013, Published by Academy, DOI: 10.4304/jcp.8.4.943-950., Available: http://www.jcomputers.us/vol8/jcp0804-15.pdf

[18] Imad Afyouni, Cyril Ray and Christophe Claramunt, "Spatial models for context-aware indoor navigation systems: A survey", *Journal of Spatial Information Science*, Online ISSN: 1948-660X, pp. 85–123, No. 4, May 2012, DOI: 10.5311/JOSIS.2012.4.73., Available: https://digitalcommons.library.umaine.edu/josis/vol2012/iss4/5/

[19] Jianchao Luo and Hao Feng, "A Framework for NFC-based Context-aware Applications", *International Journal of Smart Home*, Print ISSN: 1975-4094, Online ISSN: 2383-725X, pp. 111-122, Vol. 9, No. 1, January 2015, Published by SAGE, DOI: 10.14257/ijsh.2015.9.1.12, Available: https://gvpress.com/journals/IJSH/vol9_no1/12.pdf.

[20] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason I. Hong and Ian Oakley, "CASA: context-aware scalable authentication", in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, 24-26 July 2013, Newcastle, UK, Online ISBN: 978-1-4503-2319-2, pp 1–10, DOI: https://doi.org/10.1145/2501604, Published by ACM Digital Library, Available: https://dl.acm.org/doi/10.1145/2501604.2501607.

[21] Htoo A. Maw, Hannan Xiao and Bruce Christianson, "An adaptive access control model for medical data in Wireless Sensor Networks", in *Proceedings of the IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, 9-12 October 2013, Lisbon, Portugal, E-ISBN:978-1-4673-5801-9, DOI: 10.1109/HealthCom.2013.6720690, pp.303-309, Available: https://ieeexplore.ieee.org/document/6720690.

[22] Tao Feng, Jun Yang, Zhixian Yan, Emmanuel M. Tapia and Weidong Shi, "TIPS: context-aware implicit user identification using touch screen in uncontrolled environments", in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, 26-27 February 2014, California, USA, Online ISBN: 978-1-4503-2742-8, DOI: https://doi.org/10.1145/2565585.2565592, pp. 1–6, Available: https://dl.acm.org/doi/10.1145/2565585.2565592.

[23] Matthias Baldauf, Schahram Dustdar and Florian Rosenberg, "A survey on context-aware systems", *International Journal of Ad Hoc and Ubiquitous Computing*, Print ISSN:1743-8225, Online ISSN:1743-8233, pp. 263-277, Vol. 2, No. 4, 1st June 2007, Published by Inderscience Publishers, DOI: https://doi.org/10.1504/IJAHUC.2007.014070, Available: https://dl.acm.org/doi/10.1504/IJAHUC.2007.014070.

[24] Georgia M. Kapitsaki, George N. Prezerakos, Nikolaos D. Tselikas and Lakovos S. Venieris, "Context-aware service engineering: A survey", *Journal of Systems and Software*, Online ISSN: 0164-1212, pp. 1285-1297, Vol. 82, No. 8, 1st August 2009, Published by Elsevier Science Inc., DOI: 10.1016/j.jss.2009.02.026, Available: https://dl.acm.org/doi/10.1016/j.jss.2009.02.026.

[25] Hassan Bazoun, Youssef Bouanan, Gregory Zacharewicz, Yves Ducq, Hadrien Boye *et al.*, "Business process simulation: transformation of bpmn 2.0 to devs models (wip)", in *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative*, 13-16 April 2014, Tampa, Florida, Online ISBN: 978-1-62748-032—1, pp. 1–7. Available: https://dl.acm.org/doi/10.5555/2665008.2665028

[26] Dandan Yang, Zhiquing Wei and Yongquan Yang, "A Novel Implementation of a Hash Function Based on XML DOM Parser", in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 17-19 September 2015, Xi'an, China, E-ISBN: 978-1-4673-9200-6, DOI: 10.1109/CyberC.2015.21, pp. 5-8, Published by IEEE, Available: https://ieeexplore.ieee.org/document/7307779.