A Requirements Analysis Framework for Development of Service Oriented Systems

Shreya Banerjee  
Department of Computer Applications,  
National Institute of Technology,  
Durgapur, India  
shreya.banerjee85@gmail.com

Anirban Sarkar  
Department of Computer Applications,  
National Institute of Technology,  
Durgapur, India  
sarkar.anirban@gmail.com

ABSTRACT
In Service Oriented Systems (SOS), implementation of business processes is accomplished through services in distributed, loosely coupled manner based on business process requirements of the users. Consequently, importance of business process requirements analysis for development of SOS is strongly highlighted in both academia and industry. Usually, traditional requirement engineering is competent enough to specify and analysis business requirements for development of software systems efficiently. However, Service Oriented Requirement Engineering (SORE) emerging for SOS development is differ from traditional requirement engineering due to complex nature of services. Yet, a serious gap is still exist between early and detailed specification of business process requirements in SORE and further mapping towards design of SOS from set of business processes. To address this issue, in this paper, a requirements analysis framework is proposed for development of SOS systems. The contribution of the proposed work is formal representation of business process requirements for SOS based on business scenario and Cause-Effect-Dependency (CED) graph in dimensions of six aspects of services – What, Why, How, Who, When and Where (5W1H). Both early and detailed level requirements analysis in the context of SORE is facilitated by the proposed approach. Beside, traceability of proposed approach towards design of business processes for development of SOS is also exhibited in this paper. Moreover, the practical utility of the proposed approach is demonstrated using a suitable case study.

Keywords
Business Process Requirement, Service Oriented Requirement Engineering, Consumer Viewpoint, Provider Viewpoint, Requirements Analysis Framework

1. INTRODUCTION
Over the years, automation of business process is driven by emerging roles of stakeholders and their respective requests. Consequently, the need for specification and analysis of business requirements as per stakeholders’ demands is widely recognized [1]. To meet such necessity, service oriented systems (SOS) are developed to represent and implement business processes as services in loosely coupled manner depending on specific goals of stakeholders [5]. Service Oriented Requirement Engineering (SORE) is evolved as a crucial step specific towards development of SOS. In contrast to traditional requirements engineering, SORE is different due to existence of two fold viewpoints like service consumer and service producer [12, 13, 14], dynamic nature (identification of visibility, capability, interaction and effect) and business process driven approach (identification of business process flow). Further, a six steps process consisting of concepts - What, Why, How, When and Where (5W1H) - is broadly followed for architecting services in SOS [19]. In this context, scope of services (Capability and Effect) is defined by the word What; the word Who is used for external actors driving or interacting with services (both consumer and producer). The concept of Why is identified via the reason of communications between two services (Interaction) or service and external actors (Visibility); The term How is recognized using details of processes coordinating the services and how a service itself will be [19]. Besides, using the word When, the happening time of business activities relative to each other is acknowledged. Next, through the notion of Where, locations of service factors relative to one another are signify [19]. Further, using What aspect the producer viewpoint and through why aspect the consumer viewpoint is recognized. Thus, a SOS-based system is developed based on the fact that how actors are able to interact with services in order to deliver a real world effect towards attaining some business process requirements. As capability, the target service should exhibits the business process requirement. A Real World Effect is delivered to a consumer by means of this Capability when accessed via a service, and it will be accomplished through evidences provided via the service description [4]. Hence, in the context of SOS, 5W1H characteristics of services should be reflected in specification and analysis of business process requirements.

Usually, in business process driven approach, business process requirements are synthesized based on stakeholders’ objectives or goals. Business processes are comprising of a defined set of business activities recognizing details of steps required to achieve business process requirements. Simultaneously, a good business scenario is competent to enable individual business activities as well as business processes to be viewed in relation to one another coordinated by distinct business rules and overall business objectives [3]. Further, business scenarios are refined towards workflows. In this context, workflow management system is defined to coordinate and allocate work through various stages for execution of business processes [2].

In general, development of SOS from a set of business processes through services are done in three generic steps. The First step is specification and analysis of business process requirements. This step is consisting of two parts. In the first part, business process requirements are captured and specified in a meaningful way. In the second part, captured requirements should be analyzed in an effective way so that identification of a service in later stages can be conducted efficiently. Second step is design of business processes and workflows. In this step, those analyzed

![Figure 1. Steps in service oriented system (SOS) development.](image-url)
business process requirements are represented in terms of business processes and workflow models. Finally, the last step is implementation of services based on designed business processes and workflows. Figure 1 is the illustration of those steps for development of SOS. Distinct tools and approaches are exist in both literature and industry for aiding different steps of business process automation. Such as, BPEL (Business Process Modelling Notation) [6], YAWL (Yet Another Workflow Language) [8], XPDL (XML Process Definition Language) [9], UML Activity Diagram [10] etc. are the most well-known approaches for design step. Likewise, Web Service Business Process Execution Language (WS-BPEL) [7] and Web Service Description Language (WS-BPEL) [11] are tools for implementation of services from set of business processes.

Yet, several challenges are exist in representation and analysis of business process requirements for development of SOS systems. Firstly, serious gaps are still exist between detailed representation (What aspect) of end-users’ (Who and Where aspect of service) business goals (Why aspect of service) and corresponding business scenarios consisting of distinct flow sequences and business rules to achieve those business goals (How and When aspect of service) [15]. Secondly, distinct characteristics of services are absent in current workflow systems those are important to represent SOS rigorously and unambiguously [16, 17]. Thirdly, dependencies and flow sequences (How aspect of service), those are important to obtain an effective business process flow among distinct business objectives, are not captured efficiently. Fourthly, there is a lack of efficient methods for matching and selecting available services against users’ requirements (matching between What and Why aspect of service) [18]. Finally, however, early analysis of requirements is proficiently carried out, less attentions are paid towards detailed analysis of business process requirements in the context of SORE.

To address these challenges, this paper propose a requirements analysis framework is proposed for development of SOS that is capable to address suitable mechanisms for identification of, (i) business scenarios (How and When aspect) in terms of set of business activities and its related effects which will in turn realize service functionalities as capability and set of real world effects (What aspect of service); (ii) actors and their roles (Who and Where aspect) who will interact with a service to address a set of business objectives (Why aspect of service); (iii) granularities in business activities and its effects to address abstraction in services; (iv) dependencies among business activities to establish a well-formed business process flow (How and When aspect of service) to attain specific business process requirements. In this context, a business scenario is intended to describe a service use-case that is proficient to represent encapsulated business function which is comprising of set of absent/present business activities determining the absence/presence of a business objective. The business scenario will ensure that the described service is able to accomplish a business activity in a single invocation to achieve desired real world effect and exhibit right level of granularity. Further, this will facilitate the provisioning of service capabilities in right level of abstraction. All these mechanisms together are able to unify consumer viewpoints and producer viewpoints and further are capable to reduce the semantic gaps between business process requirements and service functionalities.

2. RELATED WORK

Researches related to specification and analysis of business process requirements for development of SOS can be classified in two directions. One type of research direction is envisioned to focus on SORE towards aligning business process requirements and service descriptions. Another type of research direction is shaped to emphasis on specification of business scenarios as well as workflows within a service.

In the first type of research initiative, several approaches are proposed in literature. A requirements engineering framework for SOS is described in [12] where service requirements engineering process is comprised of two phases as (i) “initial service requirements specification as a value driven, creative and innovative approach” and (ii) “service requirements evolution management during service usage in practice as a change management approach”. In another proposal [20], service requirements are realized using Core Ontology for REquirements (CORE) by specifying the stakeholders concerns on domain level requirements of the target system. Further, a mapping mechanism from CORE based service requirements description to web service description language (WSDL) is described in this approach. The approach is effective towards the requirements specification from provider point of view but is lacking from any requirement modelling language in order to capture the requirements in consumer viewpoints. Several recent SORE approaches are capable to extend the concepts of RGPS meta-model [21]. The RGPS meta-models are ontology driven hierarchical approaches towards modelling requirements for SOC paradigm. It is consisting of four meta-model layers namely, Role, Goal, Process and Service to provide a unified view towards requirements elicitation, analysis and domain level modelling of Networked Software [21]. However, in RGPS proposals no methodologies are defined for using the meta-model layers in order to analyze and model requirements [22]. The RGPS meta-models are further extended by several proposals [22, 23] using i* Framework and Goal Oriented Requirements Engineering approach to fetch the positive features of both the approaches towards SORE. These approaches are mainly devised to describe the methodology for Role-Goal, Process and Service meta-model layer of RGPS using i* Framework, Business Process Execution Language (BPEL) and Web Service Description Language (WSDL) respectively. The goal oriented approach is efficient to rationalize the different facets of early requirements analysis and design phases of large scale software system but not suitable for detailed level requirements analysis of such software. Moreover goal oriented approach in requirements analysis in general is not efficient to promote models’ reuse and heterogeneity [22].

In the second type of research initiative, few approaches are purposed to pay attentions towards representations of business scenarios and further workflows according to distinct characteristics of a service. In [24], a method is described to facilitate “better understanding of execution order and integration dependencies of user goals by making use of business process models”. In this approach, user stories are integrated with BPMN models. However, this method is based on requirement of two documents. One is list of users’ goals and another is corresponding BPMN models. But, BPMN models are not semantically enrich to specify several relevant concepts of business process and further consisting of various ambiguities due to lack of strong mathematical foundation. In [26], a tool called GPMN-Edit (Goal-oriented Process Modelling Notation Editor) is described for modelling goal-oriented workflows. In this approach representation of users’ goals (Why and Who aspect) is highlighted, yet, flow sequences and dependencies related to those objectives (How and When aspect) are not stressed. Further, goal oriented approaches are not suitable for detailed analysis of requirements. In [25], a mechanism is described to develop business processes using scenario driven approach and further such business processes are specified as WS-BPEL (Web Service – Business Processing Execution Language) descriptions. In this approach, a tool named as Sketch is also devised that is able to realize the described approach. Initially, scenarios are illustrated as UML Sequence diagrams in this methodology. Then those diagrams are systematically transformed towards state-machines, WS-BPEL flavored UML Activity diagrams and further towards WS-BPEL descriptions. However, dependencies between distinct objectives are not expressed in a suitable way (How and When aspect). In [27], a distributed workflow based approach is described based on goal-oriented workflow to generate, gather, distribute and store events those are resulting from specific business workflow. The primary focus of this approach is on establishment of cause-source hierarchy relationships between business objectives and actions of workflow. Yet, this focus is also confined towards the representation of distinct goals. There is no clear descriptions about how to achieve the flexibility of flow sequences to accomplish business goals and further dependencies among distinct goals (How and When aspect).
3. SCOPE OF THE WORK
In this paper, a business requirements analysis framework is proposed for development of SOS in order to address challenges and meet objectives mentioned in the introduction section. The proposed requirements analysis framework is consisting of three phases – (i) Early Requirements analysis, (ii) Detailed Requirements analysis and (iii) Mapping. A formal concept of Business Scenario is proposed in early requirements analysis phase. The concept of Business Scenario is intended to address the fact that a business activity can be accomplished through a service in a single invocation to achieve desired real world effects. Further, users’ objectives are also represented in early requirements analysis phase. Thus, the early requirements analysis phase belongs to the proposed requirements analysis framework is able to specify business process requirements in respect of consumer viewpoints. Besides, Cause-Effect-Dependency (CED) graph is proposed in detailed requirements analysis phase in the purpose of detailed level analysis of business requirements for devising of SOS. The proposed CED graph is a major extension of Cause-Effect Graph [28]. This proposed graph is based on the consideration that in SOS, a set of real world effects are caused by each set of business activities to meet certain business process requirements. The proposed CED graph is primarily capable to represent and identify granularities and dependencies among activities pertain in some business process flow in order to match with consumer viewpoints specified in early requirements analysis phase. For any example system, the proposed CED graph can be generated from the description of specific Business Scenario. Thus, proposed CED graph is capable to represent business process requirements in respect of provider viewpoints and further facilitate in matching of consumer and provider viewpoints. Beside, in each of stage, corresponding facets are represented in mathematical logic. Moreover, to prove the traceability of the proposed framework towards design, in the Mapping phase of the proposed framework business processes are realized from CED graphs using BPMN.

The novelty of the proposed work is several. Firstly, the concept of Business Scenario along with CED graph is capable to represent and identify services and its descriptions, service abstraction, service capabilities and real world effects. Secondly, through representation of users’ goals and corresponding business scenarios in early requirements analysis phase, the proposed methodology is competent to represent consumer viewpoints. Thirdly, through specification of granularities and dependencies among activities relating with business process flows in detailed requirements analysis phase, the proposed methodology is able to represent provider viewpoints. Thus, proposed approach is capable to match provider viewpoints (What aspect of service) with stakeholders’ objectives (Who, Why and Where aspect of service) and related business scenarios (How and When aspect of service) in connection to consumer viewpoints. Fourthly, distinct facets of both early and detailed requirements analysis phase are represented in mathematical logic. Thus the proposed model is competent to provide unambiguous representations and precise specification of business process requirements. Fifthly, the gap between early and detailed specification of business process requirements in SORE are reduced with the aid of simultaneous representations of stakeholders’ goals and related business scenarios in the prescribed approach. This ability is also meant to facilitate matching and selecting available services against users’ requirements. Sixthly, the proposed work is capable to represent both consumer and provider using concept of users’ roles (Who aspect of service). In the same time, locations of service provider and consumer as well as distinct service factors can be represented through users’ roles implicitly (Where aspect of service). Finally, the proposed methodology is useful towards validation and verification of service oriented requirements. Moreover, the proposed approach is expressive enough for both early and detailed level analysis of business requirements. Besides, the proposed approach is efficient to exhibit traceability towards design of business processes for development of SOS.

With the aforementioned objectives, the paper is organized in following way. The proposed requirements analysis framework is specified in section 4. Next, analysis of business process requirements in several directions based on CED graph is specified in section 5. Afterwards, in section 6, proposed approach is illustrated using a suitable case study. Finally, the paper is concluded in section 7 along with several future works.

4. PROPOSED REQUIREMENTS ANALYSIS FRAMEWORK: CAUSE-EFFECT-DEPENDENCY (CED) GRAPH BASED
Analysis of business process requirements for development of SOS should consider six characteristics of service architecture – Who, What, Why, How, When and Where. To incorporate these six criteria with requirements analysis, in this section, a requirements analysis framework is proposed. The proposed analysis framework is consisting of three phases – (i) Early Requirements analysis phase, (ii) Detailed Requirements analysis and (iii) Mapping Phase. These three phases and their corresponding descriptions are specified in following sub-sections.

In the first phase, users’ requirements are represented based on formal concept of Business Scenario. Further, in the second phase, those business scenarios are analyzed in detailed level based on proposed CED graph. Besides, facets in both phases are represented formally in mathematical logic. Further, in the third phase, business processes are realized from CED using BPMN to facilitate the traceability towards design from requirements analysis of business processes for development of SOS. Furthermore, those business processes may be more refined to obtain workflows.

4.1 Early Requirements Analysis Phase
The Early Requirements Analysis phase is intended to identify stakeholders’ roles and goals, corresponding business activities, and real world effects related with a service. To achieve these objectives in early requirements analysis phase, the formal concept of business scenario is proposed in this section. Business activities, corresponding real world effects and business rules can be identified through this formal concept of business scenario. Simultaneously, six characteristics of services can be handled effectively. This phase is consisting of distinct parts – (i) Identification of User Roles, (ii) Identification of User Goals, (iii) Identification of distinct business activities and their effects and (iv) Specification of Business scenario. Consequently, this phase is competent to represent business process requirements from consumer viewpoints with the identification of users’ goals, roles, and corresponding business scenario.

(i) Identification of User Roles (Who and Where aspect): In this step, roles of users are identified. In the context of service interaction, Roles are separation of users’ concerns when a service is interacted by them. Regarding this, through users’ roles, Who aspect of service is recognized. Usually, external actors, by whom services are provided and used, are represented by Who aspect of service. Thus, who is associated with roles played by external actors (including service provider and service consumer) who drive or interact with services. Likewise, where aspect of service architecture is identified by the location of users which is inherent in users’ roles. Users’ roles may be represented as a set RL. The formalization is specified below.

$$F1: \forall rl((rl \in RL) \rightarrow (L(rl(RL))))$$

Explanation: In this axiom, rl is a single role belongs to set of Role (RL). L is a predicate over function rl(RL) which takes RL as arguments and returns roles.

(ii) Identification of User Goals (Why aspect): Users have different objectives or goals when distinct roles are played by them. Based on those objectives services are consumed by users. Usually, the cause of communication between services and external actors or between numbers of services are recognized by Why aspect. Thus, with the concept of user goals Why aspect of service is represented. The formalization of Users’ goals is specified below.
**F2:** \( \forall r \exists G(r \in RL) \rightarrow (G \in Goals) \)

**Explanation:** Goals are specified as a set of Goal. G is a member of Goals and \( r \) is a member of RL. This formalization is intended to express the fact that users have certain goals if and only if in the same time roles are played by them.

(iii) **Identification of distinct business activities and their effects (What aspect):** A single user goal can be represented as a real-world effect and those can be achieved through sequences of business activities. Those business activities can be recognized as task lists. Consequently, real-world effects can be specified as set of effects. Hence, it can be stated that a specific goal \( G \) may be achieved through an effect (E) that is resulted from completion of distinct tasks. Generally, the scope of the service that is consisting of the capability and real world effect of service is specified using What aspect of service. Thus, through distinct task lists and their corresponding effects What aspect of service is achieved. Consequently, the formalization specified below is capable to realize What aspect in terms of the relationship between Goals(G), the corresponding business activities and their effects.

\[
F3: \forall G \exists T \exists E ((G \in Goals) \rightarrow (Effectlist(E) \land (Tasklist(TA) \land (TA \rightarrow E))))
\]

**Explanation:** In this formalization, business activities are specified as a predicate Tasklist and TA is an instantiation of that predicate. Likewise, real-world effects are specified as a predicate Effectlist and Eff is an instance of that predicate. \( (TA \rightarrow E) \) – this fragment of \( F3 \) is considered to specify that effects are entailed from tasks. Individual formalism of Tasklist and Effectlist are specified below.

\[
F4: \forall TA \exists ta((Tasklist(TA) \land Task(ta)) \rightarrow (ta \in TA))
\]

**Explanation:** In this formalization, Tasklist is a set of tasks (ta). Thus, ta is representation of a single business task or activity and Tasklist is the representation of set of business activities.

\[
F5: \forall Eff \exists E((Effectlist(Eff) \land Effect(E)) \rightarrow (E \in Eff))
\]

**Explanation:** In this formalization, Effectlist is a set of Effects (E). Thus, E is represented as a single effect and Effectlist is represented as a set of real world effects.

(iv) **Specification of Business Scenario (How and When aspect):** Sequences of business activities and corresponding effects can be represented as a business scenario through which a user goal is accomplished. Several effects may be endeavor from a set of business activities by the means of implementation. Beside, a business activity may be dependent on effects of previously accomplished business activities for its own implementation. Thus, a business scenario can be formally expressed as flow sequences and dependencies between tasks and corresponding resulted effects. Generally, the detail of the processes that coordinate the service is specified by How aspect. Hence, using both dependencies and flow sequences among task lists, How aspect of service can be achieved. Besides, the timing of service is usually realized by When aspect of service. In the proposed framework, When aspect of service can be acquired through dependency relationships among business activities because through dependency one task is dependent on a task that is previously created. Further, dependency relationships can be of different types and time dependency is one of them. The formalization of business scenario is specified below.

\[F6: \]

Sequence1: \((\neg Tasklist(TA1) \land \neg ON\_IMPLEMENTATION\_OF(TA1)) \rightarrow Effectlist(E1))

Sequence2: \((\neg Tasklist(TA2) \land \neg ON\_IMPLEMENTATION\_OF(TA2) \land \neg DEPENDENT\_ON(E1) \land \neg DEPENDENT\_TYPE(Message \lor Data \lor Time) \rightarrow Effectlist(E2))

Sequence3: \((\neg Tasklist(TA3) \land \neg ON\_IMPLEMENTATION\_OF(TA3) \land \neg DEPENDENT\_ON(E2) \land \neg DEPENDENT\_TYPE(Message \lor Data \lor Time) \rightarrow Effectlist(E3))

**Explanation:** In this formalization, \( ON\_IMPLEMENTATION\_OF() \) is a predicate presenting implementation of business activities (TA). Similarly, \( DEPENDENT\_ON() \) is a predicate implying dependencies of business activities (TA) on previously generated effects (E) resulted from another business activity. \( Sequence \ 1, \ 2, \ \& \ 3 \) are represented as flow sequences of business activities in a specific scenario. Further, \( DEPENDENT\_TYPE() \) is a predicate indicating types of dependency of business activities on previously generated effects.

4.2 Detailed Requirements analysis Phase

The purpose of detailed requirements analysis phase is to describe the service within its operational environment, along with its detail functionality, business rules. In early requirements analysis phase of the proposed framework, distinct business constraints, granularity among the business activities and type of dependencies are not identified in full. Consequently, the necessity of detailed requirements analysis phase is raised. To facilitate objectives of this phase, Cause Effect-Dependency (CED) graph is proposed. Propose CED graphs are intended to represent effect list, task list and business scenarios identified in early requirements analysis phase along with detail level representation of business constraints, rules and dependencies for detailed level analysis of business process requirements. In this way, proposed CED graph is proficient to represent and identify the granularities and dependencies among the activities pertain in some business process flow. Accordingly, CED graph is capable to represent provider viewpoints and further matching of consumer and provider viewpoints. Distinct facets of the proposed graph is illustrated below in both graphical notations and mathematical logic.

4.2.1 **Formal Specification of Cause-Effect-Dependency(CED) graph**

A Cause-Effect-Dependency graph is consisting of set of nodes (V), edges (E) and can be represented as a seven tuple – \( \{C, Eff, TR, DR, G, L\} \). The set of nodes \( V \) of two types – one is Cause (C) and another is Effect (Eff). The set of edges \( E \) of two types – one is Transformation Relationship (TR) and another is Dependency Relationship (DR). Further, using Constraint Function (G), constraints are imposed on Causes or Effects. Beside, through Label Function (L) associated with dependency relationships corresponding dependency types are specified.

(a) **Cause (C):** Cause can be defined as a set of input entities bringing changes in a business process. These input entities can be of different types such as a single task, activity, event or a user input. Every Cause is related with a specific creation time. Tasks specified in section 4.1 is mapped towards these Cause nodes. Cause (C) can be formally represented as a member of the set Cause List. The formalization is specified below.

\[F7: \forall Cause\_List \exists C((\neg Cause(C) \land (c \in Cause\_List)) \rightarrow (Q(Cause\_List(C)) \leftrightarrow C))\]

**Explanation:** In this formalization, Cause is a predicate implying a single cause \( C \). \( Q \) is a predicate over function \( Cause\_List() \) taking \( C \) as arguments and returns causes.

The graphical notation of a Cause can be a circle where “C” is written inside the circle.
(b) Effect (E): Effect can be defined as a set of output states those are resulted from a combination of causes. Every effect is related with a specific creation time. Effects specified in section 4.1 is mapped towards these Effect nodes. Effect (E) can be formally represented as a member of set Effect_List. The formalization is specified below.

\[ F8: \forall \text{Effect}_\text{List} \exists E((\text{Effect}(E) \land (E \in \text{Effect}_\text{List})) \rightarrow (Q(\text{Effect}_\text{List}(E)) \rightarrow E)) \]

**Explanation:** In this formalization, Effect is a predicate implying a single effect E. Q is a predicate over function Effect_List() taking E as arguments and returns effects.

The graphical notation of an Effect can be a circle where “E” is written inside that circle.

(c) Transformation Relationship (TR): This is a relationship directing a set of Causes towards its corresponding Effects by different means. However, in this relationship, related effects are synthesized after causes are created. A transformation relationship can be formally represented as follows.

\[ F9: \forall \exists 2 \exists \text{Con}((\text{Cause}(C) \land \text{Effect}(E) \land \text{TR}(C, E) \land \text{Time}(t1) \land \text{Time}(t2)) \rightarrow ((C \rightarrow E) \land \text{Starting}_\text{time}(C, t1) \land \text{Starting}_\text{time}(E, t2) \land (((t2 < t1) \lor (t2 = t1)) \land \neg((t2 < t1)))) \]

**Explanation:** In this formalization, TR() is a predicate specifying the Transformation Relationship between instances of Cause and Effects. Beside, Time is a predicate implying the time. Likewise, Starting_time is a predicate indicating creation time of a cause or an effect. Thus, two pairs of arguments – Cause(C), time(t) and Effect(E), time(t) – can be taken by this predicate.

The graphical notation of this relationship is a solid line arrow.

(d) Dependency Relationship (DR): This is a relationship between Cause and Effect when causes may be dependent on effects those are created before creation time of that cause by different types such as Message/Data/Time. This relationship is associated with dependency constraint label (L). A dependency relationship can be formally represented as follows.

\[ F10: \forall \exists 2 \exists \text{Message} \exists \text{Data} \exists \text{Time}((\text{Cause}(C) \land \text{Effect}(E) \land \text{DR}(C, E) \land \text{Effect}(E) \land \text{Time}(t1) \land \text{Time}(t2)) \leftrightarrow ((C \rightarrow E) \land \text{Starting}_\text{time}(C, t1) \land \text{Starting}_\text{time}(E, t2) \land (((t2 < t1) \lor (t2 = t1)) \land \neg((t2 < t1)))) \]

**Explanation:** In this formalization, DR() is a predicate specifying the Dependency Relationship between instances of Cause and Effects. Further, Time() is a predicate specifying time; Starting_time() is a predicate specifying synthesis time of Causes and Effects. Beside, DM(), DD() and DT() are predicates denoting Message Dependency type, Data Dependency type and Time Dependency type respectively.

The graphical notation of this relationship is a dash line arrow with the related Label (L).

(e) Label (L): This is specification of dependency type associated with dependency relationship. A label can be message, data and time dependency.

\[ F11: \forall \text{DR}((\text{Label}(l) \land Q(\text{DR}, l)) \leftrightarrow (\text{DR}(\text{Cause}, \text{Effect}) \land \text{dependent}_\text{by}(\text{DM} \lor \text{DD} \lor \text{DT}))) \]

**Explanation:** In this formalization, Label() is a predicate implying label of DR relationship. Further, dependent_by() is a predicate implying types of dependencies of Causes on previously generated Effects.

Label (L) is graphically represented as type of label such as “Message”/“Data”/“Time” associated with DR relationship.

(f) Constraint Function (G): This function may be imposed on either Causes or Effects with distinct conditions based on which transformation from distinct causes towards corresponding effects (Transformation Relationship (TR) edge) or dependencies of distinct causes towards effects (Dependency Relationship (DR) edge) are accomplished. A constraint Function can be represented as \(G(\text{Cause}, \text{Conditions})\) or \(G(\text{Effect}, \text{Conditions})\). This function cannot be applied simultaneously on two different types of edges (Transformation Relationship (TR) and Dependency Relationship (DR)). Thus, the formal representation for this function is consisting of two parts. One part is for Transformation Relationship (TR) and another is for Dependency Relationship (DR).

The formalization for Constraint Function (G) when Transformation Relationship (TR) is occurring is as follows.

\[ F12: \forall G: \exists 2 \exists \text{TR}((Q(G(c, \text{Cnd}) \lor G(e, \text{Cnd})) \land \text{Cause}(c) \land \text{Effect}(e) \leftrightarrow (\text{TR}(c, e) \land \neg\text{DR}(c, e))) \]

Similarly, the formalization for Constraint Function (G) when Dependency Relationship (DR) is occurring is as follows.

\[ F13: \forall G: \exists 2 \exists \text{DR}((Q(G(c, \text{Cnd}) \lor G(e, \text{Cnd})) \land \text{Cause}(c) \land \text{Effect}(e) \leftrightarrow (\text{DR}(c, e) \land \neg\text{TR}(c, e))) \]

**Explanation:** In this formalization, Cnd is intended for implying distinct conditions imposed on Cause and Effect in Transformation Relationship (TR) and Dependency Relationship (DR) relationships. Further, Cnd may be of distinct types – inclusive_or, AND, NOT, exclusive_or, Require and Mask. Constraint Function (G) is graphically represented as “G(Cause/Effect, Condition)” associated with Cause or Effect nodes.

Table 1 is the summarization of the graphical notations of proposed CED graph.

<table>
<thead>
<tr>
<th>Table 1: Graphical notations of facets of CED graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cause-Effect-Dependency graph facets</strong></td>
</tr>
<tr>
<td>Cause (C)</td>
</tr>
<tr>
<td>Effect (E)</td>
</tr>
<tr>
<td>Transformation Relationship (TR)</td>
</tr>
<tr>
<td>Dependency Relationship (DR)</td>
</tr>
<tr>
<td>Label (L)</td>
</tr>
<tr>
<td>Constraint Function (G)</td>
</tr>
</tbody>
</table>
4.2.2 Illustration of Cause-Effect-Dependency (CED) Graph using a case study

In this section, a case study is specified based on a business process of hiring new four employees in an organization. This example case is consisting of several sequences of tasks. The process is initiated with the publication of hiring advertisement. After that, candidates are introduced and registered in the payroll. Then four candidates are chosen through interview. Finally, those are hired in the organization.

Distinct steps related to this scenario are specified below. Equivalent CED graph is illustrated in figure 2. In this figure, distinct causes related to the example case study are represented as Causes C1, C2, C3… and distinct effects are represented as Effects E1, E2, E3… in CED graph. The sequences of steps related to the example case is specified below.

Step 1. Publication of an Advertisement.
Step 2. Candidates Orientation Process is started.
Step 3. Candidates are introduced around the company.
Step 4. Candidates are registered in the payroll.
Step 5. Four People are chosen through interview.
Step 6. Information about the candidates not selected are deleted from payroll.
Step 7. Training of new employees are begun.

4.3 Mapping Phase: Realization of Business Processes from proposed CED graph

In this section, business processes are realized from proposed CED graphs (using BPMN). This realization is specified in Table 2. From this comprehension, the fact is demonstrated that formally specified business process requirements in proposed requirements analysis framework is easily traceable towards design of business processes for development of SOS.

5. ANALYZED FEATURES OF SERVICES BASED ON CED GRAPH

Using CED graph, requirements of a SOS can be analyzed in detailed form in several directions such as – Business Process Flow in services, granularity, and six aspects of service architecture – what, who, why, how, when and where. In this section, those detailed level analysis is specified. Further, based on both early and detailed level analysis services can be identified from set of business processes. To realize this fact, in this section a step wise method is proposed for identification of services from the proposed CED graph.

(a) Business Process Flow in Service: A business scenario may be consisting of different types of flows in a business process depending on distinct business constraints. Those distinct types of flows can be identified using Constraint Functions (G) and dependency types of proposed CED graphs. Based on this, workflows in services are identified and analyzed. These variability in business process flows of a service is summarized in Table 3. Further, different variations of Inclusive_Or

---

**Figure 2. Example of Cause-Effect-dependency (CED) graph.**

**Table 2. Realization of business processes from proposed CED graph**

<table>
<thead>
<tr>
<th>Proposed CED graph notations</th>
<th>Equivalent BPMN Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause Vertex</td>
<td>Task, Activity, Event</td>
</tr>
<tr>
<td>Effect Vertex, Transformation Relationship, Dependency Relationship</td>
<td>Sequence Flow, Default Flow, Conditional Flow</td>
</tr>
<tr>
<td>Constraint Functions</td>
<td>Gateway</td>
</tr>
<tr>
<td>Label</td>
<td>Labels on flow</td>
</tr>
</tbody>
</table>
Table 3. Distinct types of Business Scenarios, Equivalent CED graph and Formalization in mathematical logic

<table>
<thead>
<tr>
<th>Business Scenario</th>
<th>CED graph Expression</th>
<th>Formalization in Mathematical Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) At least one task among set of tasks have been directed or dependent on a specific effect. (Transformation Relationship)</td>
<td>(i) Inclusive Or C1,C2,C3</td>
<td>(i)∀c1∀c2∃e1((Cause(c1) ∧ Cause(c2) ∧ Effect(e1)) ∧ ¬(c1 = c2) ∧ G((c1,c2, inclusive_or)) ↔ (TR(c1,e1) V TR(c2,e1)))</td>
</tr>
<tr>
<td>(ii) At least one effect among set of effects have been directed from or dependent by a specific cause. (Dependency Relationship)</td>
<td>(ii) Inclusive Or E1,E2,E3</td>
<td>(ii)∀c1∀c2∃e1((Cause(c1) ∧ Cause(c2) ∧ Effect(e1) ∧ ¬(c1 = c2) ∧ G((c1,c2, inclusive_or)) ↔ (DR(c1,e1) V DR(c2,e1)))</td>
</tr>
<tr>
<td>(i) All of tasks have directed towards a specific effect or dependent on a specific effect. (Transformation Relationship)</td>
<td>(i) And C1,C2,C3</td>
<td>(i)∀c1∀c2∃e1((Cause(c1) ∧ Cause(c2) ∧ Effect(e1) ∧ ¬(c1 = c2) ∧ G((c1,c2, and)) ↔ (TR(c1,e1) ∧ TR(c2,e1)))</td>
</tr>
<tr>
<td>(ii) All of effects in an effect list can be directed from or dependent by a specified task. (Dependency Relationship)</td>
<td>(ii) And E1,E2,E3</td>
<td>(ii)∀c1∀c2∃e1((Cause(c1) ∧ Cause(c2) ∧ Effect(e1) ∧ ¬(c1 = c2) ∧ G((c1,c2, and)) ↔ (DR(c1,e1) ∧ DR(c2,e1)))</td>
</tr>
<tr>
<td>(i) A specific task has not be directed towards a specific effect or dependent on a specific effect. (Transformation Relationship)</td>
<td>(i) Not C1,C2</td>
<td>(i)∀c1∀e1((Cause(c1) ∧ Effect(e1) ∧ G((c1),not)) ↔ ¬TR(c1,e1))</td>
</tr>
<tr>
<td>(ii) A specific task has not be directed from or dependent by a task. (Dependency Relationship)</td>
<td>(ii) Not E1,E2</td>
<td>(ii)∀c1∀e1((Cause(c1) ∧ Effect(e1) ∧ G((c1),not)) ↔ ¬DR(c1,e1))</td>
</tr>
<tr>
<td>(i) Only one task among numbers of tasks for directed towards effects or dependent on a specific effect. (Transformation Relationship)</td>
<td>(i) Exclusive Or C1,C2,C3</td>
<td>(i)∀c1∀c2∃e1((Cause(c1) ∧ Cause(c2) ∧ Effect(e1) ∧ ¬(c1 = c2) ∧ G((c1,c2, exclusive_or)) ↔ ((¬TR(c1,e1) ∧ TR(c2,e1)) V (TR(c1,e1) ∧ ¬TR(c2,e1))))</td>
</tr>
<tr>
<td>(ii) Only one effect among the members of effect list has been either directed from or dependent by a specific task. (Dependency Relationship)</td>
<td>(ii) Exclusive Or E1,E2,E3</td>
<td>(ii)∀c1∀c2∃e2((Cause(c1) ∧ Cause(c1) ∧ Effect(e1) ∧ ¬(c1 = c2) ∧ G((c1,c2, exclusive_or)) ↔ ((¬DR(c1,e1) ∧ DR(c1,e2)) V (DR(c1,e1) ∧ ¬DR(c1,e2))))</td>
</tr>
<tr>
<td>(i) One task cannot be directed towards a specific effect or dependent on an effect if another specified task cannot do the same. (Transformation Relationship)</td>
<td>(i) Require C1,C2</td>
<td>(i)∀c1c2c3e1e1(((c1 ∈ Cause) ∧ (c2 ∈ Cause) ∧ (c3 ∈ Cause) ∧ (e1 ∈ Effect) ∧ G(c1,c2, require)) ↔ ((TR(c1,e1) ∧ TR(c2,e1)) V (TR(c3,e1) V ¬TR(c3,e1))))</td>
</tr>
<tr>
<td>(ii) One effect cannot be directed from or dependent by a specific task if another specified effect cannot be directed or dependent in same way. (Dependency Relationship)</td>
<td>(ii) Require E1,E2</td>
<td>(ii)∀c1c2c3e2e2(((c1 ∈ Cause) ∧ (c2 ∈ Cause) ∧ (c3 ∈ Cause) ∧ (e1 ∈ Effect) ∧ G(c1,c2, require)) ↔ (DR(c1,e1) ∧ DR(c2,e1) V (DR(c3,e1) V ¬DR(c3,e1))))</td>
</tr>
<tr>
<td>(i) One task is directed or dependent towards an effect then another task cannot be directed towards the same effect. (Transformation Relationship)</td>
<td>(i) Mask C1,C2</td>
<td>(i)∀c1c2c3e2e3(((c1 ∈ Cause) ∧ (c2 ∈ Cause) ∧ (c3 ∈ Cause) ∧ (e1 ∈ Effect) ∧ (G(c1,c2,mask)) ↔ ((TR(c1,e1) ∧ ¬TR(c2,e1)) ∧ ¬(TR(c1,e1) ∧ TR(c2,e1)) V (TR(c3,e1) V ¬TR(c3,e1))))</td>
</tr>
<tr>
<td>(ii) One effect is directed from or dependent by a task then another effect is not directed from or dependent by the same task. (Dependency Relationship)</td>
<td>(ii) Mask E1,E2</td>
<td>(ii)∀c1c2c3e3e4(((c1 ∈ Cause) ∧ (c2 ∈ Cause) ∧ (c3 ∈ Cause) ∧ (e1 ∈ Effect) ∧ (G(c1,c2,mask)) ↔ ((DR(c1,e1) ∧ ¬DR(c2,e1)) ∧ ¬(DR(c1,e1) ∧ DR(c2,e1)) V (DR(c3,e1) V ¬DR(c3,e1))))</td>
</tr>
</tbody>
</table>

A specific task has not be directed from or dependent by a specified task. The scope of service functionality offered (iii) Performance: Performance of service is influenced by functionality and structural complexity of a service [30]. Based on proposed CED, services’ functionality is captured easily through combination of Causes (C) and Effects (E) pair. Further, structural complexity of a service is

(ii) Reusability: A real world effect provided by a service may be composed of set of Effects (E) those may be achieved through a combination of distinct Causes (C). Thus, using proposed CED a service’s Capability or business functionality can be specified using distinct Tasks at module level. This module level pairing of Tasks and Effects are able to be further reused in other services. In this way, process hierarchy and reusability of combination of those tasks and effects in different business scenarios are aided by proposed CED graphs.

A specific effect cannot be directed from or dependent by another specified task.

Flexibility in business processes: Using Constraint Function (G) of proposed CED, flexibility in business process flows in a business scenario can be achieved. Flexibility can be facilitated through Constraint Function (G) by moving away from rigid predefined paths for providing the users with multiple and alternate path of executions.

Condition is illustrated in Figure 3 as an example of different business process flows in a service based on CED graph.
The sequence flow (combination of cause and effect) is identified. Thus, consumer viewpoints are realized. Further, the method of variation of inclusive or condition based on transformation and dependency relationships are described in Figure 3. Equivalent services can be identified from the set of business process requirements represented as CED graphs. To fulfill this criteria a step wise method is proposed below.

Step 1: The scope of service is identified within the set of sequences of cause and effect. This is done by identifying the services that are required to achieve some business objective. The scope of service is identified within the set of sequences of business activities.

Step 2: The detailed capability of Service is identified by combination of causes (C) and effects (E) through transformation relationship (TR), dependency relationship (DR) and constraint function (G) within the set of sequences achieved in the step 1.

Figure 3. Different Variations of Inclusive Or condition based on Transformation and Dependency Relationships;
(a) All causes have transformed towards the effect; (b) One cause has transformed towards all of the effects; (c) All of the causes has dependent on one effect; (d) One cause has dependent on all of the effects.

(e) One cause has dependent on all of the effects.

Step 4: The real world effects of services are identified by set of effects (E).

Step 5: Variability in process flow within a service is identified by constraint function (G).

Step 6: The interaction between two services are recognized when one cause depends on one effect of another service by passing messages.

6. ILLUSTRATION OF THE PROPOSED FRAMEWORK USING A CASE STUDY
In this section, the proposed work is demonstrated using a case study. The case study is described in [29]. A business process is described in this case study whose goal is to help dermatologist in treatment of patients with skin cancer and premalignant skin lesions. According to the interview with dermatologist, this case study is consisting of three phases having several goals as per role of dermatologist. In the following, those phases and respective users’ roles and goals are listed. Beside, distinct phases of proposed requirements analysis framework are conducted on those goals. At the last, relative services based on the example business process are identified.

6.1 Early Requirements analysis
In this phase the case study is represented based on proposed methodology of early requirements analysis phase in section 4.1.

(a) Identification of Users’ Roles (Who and Where aspect):
Role: Dermatologist Place: City Hospital

(b) Identification of Users’ Goals (Why aspect):
(i) Phase 1: Diagnosis Phase
Dermatologist’s Goals: Meeting with patients, Collecting sample diagnoses, inform patients and arrange intervention.

(ii) Phase 2: Treatment Phase
Dermatologist’s Goals: Pre-treatment care, Patient Treatment

(iii) Phase 3: After care Phase
Dermatologist’s Goals: Post treatment care, Decision about long term check-up.

(c) Identification of distinct business activities and their corresponding effects (What aspect):
From why aspect, the tasks and corresponding effects per phase can be recognized as specified in Table 4. This combination of cause and effect...
is utilized to identify the scope of the services that may be developed from the example business process case study. Based on this Cause Effect identifications, the equivalent business scenario for phase 1 is illustrated below.

Table 4. List of tasks and corresponding effects of distinct phases in the specified case study

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consult meeting with the patient (ta1)</td>
<td>Patient’s Information have been acquired (E1)</td>
</tr>
<tr>
<td>Verification of Patient’s information (ta2)</td>
<td>Patient is known (E2)</td>
</tr>
<tr>
<td>Verification of Patient’s information (ta2)</td>
<td>Patient is unknown (E3)</td>
</tr>
<tr>
<td>Take new sample (ta3)</td>
<td>Sample is unknown (E4)</td>
</tr>
<tr>
<td>Diagnosis Sample (ta4)</td>
<td>Diagnosis result have been produced (E5)</td>
</tr>
<tr>
<td>Inform Patients and schedule intervention (ta5)</td>
<td>Patients get report and schedule chart (E6)</td>
</tr>
</tbody>
</table>

**Phase 2: Treatment Phase**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide pre-treatment care towards patients (ta6)</td>
<td>Patients are ready for getting treatment (E7)</td>
</tr>
<tr>
<td>Provide pre-treatment care towards patients (ta6)</td>
<td>Patients do not want pre-treatment (E8)</td>
</tr>
<tr>
<td>Treatment patients (ta7)</td>
<td>medication, advice (E9)</td>
</tr>
</tbody>
</table>

**Phase 3: After care Phase**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provides post-treatment care towards patients (ta8)</td>
<td>Observed and recorded vital signs of patient (E10)</td>
</tr>
<tr>
<td>Provides post-treatment care towards patients (ta8)</td>
<td>Patient skip post-treatment (E11)</td>
</tr>
<tr>
<td>Provide long-term check-up (ta9)</td>
<td>Next meeting schedule with patient is necessary (E12)</td>
</tr>
<tr>
<td>Provide long-term check-up (ta9)</td>
<td>Next meeting schedule with patient is unnecessary (E13)</td>
</tr>
</tbody>
</table>

(d) Specification of the business scenario (How and When aspect):

From the specification of what aspect of phase 1 in the case study, an equivalent business scenario can be obtained. Through this business scenario, flow sequences and dependency between task lists and effect lists are identified. Thus, representation of How and When aspect of the service that may be developed from the corresponding business process of phase 1 is further enabled.

Sequence1: \((\text{Tasklist(TA1)} \land \text{ON IMPLEMENTATION OF(TA1)}) \rightarrow \text{Effectlist(E1)})

Sequence2: \((\text{Tasklist(TA2)} \land \text{ON IMPLEMENTATION OF(TA2)} \land \text{DEPENDENT ON(E1)} \land \text{DEPENDENT TYPE(Data))} \rightarrow (\text{Effectlist(E2)} \lor \text{Effectlist(E3)}))

Sequence 3: \((\text{Tasklist(TA3)} \land \text{ON IMPLEMENTATION OF(TA3)} \land \text{DEPENDENT ON(E3)} \land \text{DEPENDENT TYPE(Message))} \rightarrow \text{Effectlist(E4)})

Sequence 4: \((\text{Tasklist(TA4)} \land \text{ON IMPLEMENTATION OF(TA4)} \land \text{DEPENDENT ON(E4)} \land \text{DEPENDENT TYPE(Data))} \rightarrow \text{Effectlist(E5)})

Sequence 5: \((\text{Tasklist(TA5)} \land \text{ON IMPLEMENTATION OF(TA5)} \land (\text{DEPENDENT ON(E5)} \land \text{DEPENDENT TYPE(Data))} \land (\text{DEPENDENT ON(E2)} \land \text{DEPENDENT TYPE(Message))}) \rightarrow \text{Effectlist(E5)})

6.2 Detailed Requirements analysis and Mapping towards Design of Business Processes

Detailed requirements analysis of the service is conducted through CED graph. The specified business scenario described in section 6.1 is
6.2.1 Analyzed features of the service in detailed requirements analysis phase

(a) Business Process Flow in the Service: Identification of the process flow in the example business process is accomplished using a set of Transformation Relationships (TR), Dependency Relationships (DR) and Constraint Functions (G). Such as phase 1 is a process flow which is consisting of flow sequences

\[ TR(C1,E1), \ DR(E1,C2,DEPENDENT_TYPE(Data)), \ TR(C2,E2, G(E2,E3),MASK) \ldots \]

(b) Granularity in the Service: Granularity in the service is achieved from the example business process as follows.

(i) Flexibility in business processes: Flexibility is acquired through distinct Constraint Function (G) and impact of those Constraint Function (G) on Transformation Relationship (TR), Dependency Relationship (DR), Causes (C) and Effects (E) in the obtained CED graph. Such as impact of \( G(E2,E3),MASK \) on \( TR(C2,E2) \) and \( TR(C2,E3) \).

(ii) Reusability: Modular level pairing of Task and its corresponding Effect in the CED can be used in different business scenarios. Such as combination of Cause C1 and Effect E1 is reused in another business scenario like hospital admission.

(iii) Performance: Performance of the service implemented from the example business process is captured through the pairing of Cause C and Effect E (service functionality) in different ways and numbers of nodes of the Cause C and Effect E in the service (structural complexity).

(c) What, Why, Who, How, When and Where aspect of service architecture: Through the set of Effect E in the CED graph What aspect is recognized. Using distinct Dependency Relationship (DR) and Transformation Relationship (TR) relationship in the CED graph How and When aspect of service is identified. Further, with the help of distinct combination of Causes C and Effects E based on Dependency Relationships (DR), Transformation Relationships (TR), and Constraint Function (G) What aspect is acclaimed.

6.2.2 Identification of services from the example business process requirements

Based on the method specified in section 5 (d) services can be identified from the example business process specified in the case study. Three distinct services may be equivalent to three phases of the business process. Those services are Patient Diagnoses, Patient Treatment and After Care. These service are interacting with each other through message passing as specified in figure 4 and 5 using Dependency Relationship. Identification of Patient Diagnoses service is demonstrated below as an example of the proposed method in section 5 (d).

Identification of Patient Diagnoses service: This service is equivalent with the business process of phase one. The scope of this service is limited towards the Transformation Relationships (TR) between Cause C5 and Effect E6. Further, interaction of this service with the service Patient Treatment is conducted by message dependency from Cause C6 to Effect E6. Beside, the capability of the service is represented through the set of Causes C1, C2, C3, C4, C5; corresponding Effects E1, E2, E3, E5, E6; related Transformation Relationships TR(C1,E1), TR(C2,E3), TR(C2,E2)\ldots; and related Dependency Relationships DR(C2,E1), DR(C3,E3), DR(C4,E4)\ldots; Functionality within the service is recognized using set of Causes (C1, C2, C3, C4, C5). Likewise, real world effects of the service is realized by set of Effects (E1, E2, E3, E5, E6). Further, variability in the process flow within the service is achieved through applying Constraint Function \( G(E2,E3),Mask \) on Effects E2 and E3 when Cause C2 is directed (Transformation Relationships (TR)) towards those Effects. Similarly, variability is also achieved when dependency (Dependency Relationships (DR)) of Cause C5 on Effects E3 and E5 is constrained by Constraint Function \( G(E3,E5),Inclusive_or \).

7. CONCLUSION AND FUTURE WORK

This paper is aimed for representation of a requirements analysis framework for development of SOS from set of business processes. The proposed requirements analysis framework is consisting of three phases - (i) Early Requirements analysis, (ii) Detailed Requirements analysis and (iii) Mapping. Early Requirements analysis phase is conducted for identification of users’ goals, roles and corresponding business scenarios. Thus, this level is related to specifications of business process requirements from consumer viewpoints. Formal concept of Business Scenario is proposed to aid this phase. Further, Detailed Requirements analysis phase is accomplished for detail representations of users’ objectives and business scenarios along with enforcing business rules, constraints on them and resulted business process flows for achieving those business objectives. Thus, this level is related to specifications of
business process requirements from provider viewpoints. Cause-Effect-Dependency (CED) graph is proposed to facilitate this phase. Beside, in this phase matching of consumer and provider viewpoints is also effectively achieved since a CED graph can be directly devised from a specified Business Scenario. Further, in mapping phase, business processes can be directly realized from business scenario/CED graphs. In this way, the proposed methodology is capable to represent six aspects of services - What, Why, How, Who, When and Where aspect (SWIH) efficiently. The entire methodology is formally presented and illustrated with suitable case studies.

The contribution of the proposed approach are manifolds. Firstly, identification and analysis of What, Why, How, Who, When and Where aspect of services are conducted in a very initial stage of SOS development. Secondly, specification and analysis of business process requirements from both consumer and provider viewpoints are accomplished effectively. Thirdly, correspondence between both types of viewpoints is achieved efficiently. Fourthly, traceability of proposed business process requirements analysis framework towards design of business processes is exhibited conveniently. Fifthly, business process flow and dependencies among business activities are captured effectively with the aid of formal representation of Business Scenario and CED graph in mathematical logic and graphical forms. Sixthly, identification of service, service functionality, capability, granularity, consumer and provider viewpoint, and real world effects are conducted definitely. Finally, business process requirements for development of SOS can be further analyzed in detail level using formal specification of CED graph in several directions such as variability in business process flow, service granularity etc. Moreover, proposed mathematical logical expressions are competent to represent semantics of business requirements rigorously and unambiguously. Thus, this proposed approach is expressible for both early and detailed analysis of business process requirements.

Future work include validation and verification of business requirements for development of SOS based on proposed CED graph. Further, implementation of case tools for the purpose of automation of distinct phases of the proposed framework is an important future work. Beside, extension of CED graph with specification of Who and Where aspect in detail requirements analysis phase is also a prime future work.

8. REFERENCES


