Adaptive Process in a Pervasive System - A Holistic Hybrid Approach

Henry Eleonu[1], Jane Oruh[2]

[1] **Computer Services Department, Abia State Planning Commission, Umuahia, Abia State, Nigeria**

[2] **Department of Computer Science, Michael Okpara University of Agriculture, Umudike Umuahia, Abia State, Nigeria**

Abstract
Currently few efficient context aware systems that dynamically adapt business processes to context information exist. This calls for the study of context data that affect business processes at runtime, and also a research on adaptive systems that will dynamically reconfigure a business process at runtime to soot the current context. Context-aware systems offer entirely new opportunities for application developers and for end users by gathering context data and adapting systems behavior accordingly, especially in combination with mobile devices these mechanisms are of high value and are used to increase usability tremendously. Our goal is to develop efficient architectural frameworks for context aware systems for business processes that will respond to changing context by adapting itself to a new context. We propose a hybrid architecture which combines both middleware technologies and web services.

Keywords: *Process Adaptive, Context Aware Systems, Pervasive Systems, Ubiquitous Systems, Context Aware Business Processes, Software Architecture, context modeling.*

1. Introduction

With the popularization of mobile devices such as notebooks, smart phones and PDA's, there is an increase in the deployment of pervasive systems globally. One research area in the field of pervasive systems is context aware systems. Context aware systems are systems that are able to adapt itself to the context of the environment. When a change to a new context occurs, the system automatically reconfigures itself to adapt to the new context. In the case of a business processes that is aware of context, a particular context has its own corresponding business process. This entails that a change in context will result to a change in some parts of the business process, but not the entire business process. This type of context aware business processes are executed by context aware systems. Context aware systems increase usability and efficiency of systems.

This paper is structured as follows. Section 2 is the background of this paper. This includes a review of context aware services and application, a study of context aware business processes and context modeling techniques. In Section 3, we present a review of existent Context-aware systems architectures. We also introduce current design principles for context-aware systems and common context models used in various context-aware systems. We also made a review of approaches for adaptive process modeling. In Section 4, we discuss the proposed architectural framework and also explain the approach and highlight disadvantages. Section 5 highlights some future work that would emanate from this research. Finally, Section 6 is a list of the references.

2. Background

2.1. Business Process

2.1.1 What is a Business Process?

A business process or business method is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers. It often can be visualized with a flowchart as a sequence of activities with interleaving decision points or with a Process Matrix as a sequence of activities with relevance rules based on the data in the process [43]. One of the best definitions of process is by [19], they define a process as

"A set of linked activities that take an input and transform it to create an output. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream."

This definition also emphasizes the constitution of links between activities and the transformation that takes place within the process. [19] also included the upstream part of the value chain as a possible recipient of the process output. From the definitions above, we can see that a process must have clearly defined boundaries, input and output, and it must consist of activities that are ordered according to their position in time and space.

2.1.2 What is Business Process Modeling?

Business Process Modeling (BPM) is the activity of representing processes of an enterprise, so that the current process may be analyzed and improved. The essence of BPM is to improve process efficiency and quality

2.2. Context-Aware Computing Background

2.2.1. What "context information" includes

[38] refer to context as location, identities of nearby people and objects, and changes to those objects. In a similar definition, [5] defines context as location, identities of the people around the user, the time of day, season, temperature, etc. [33], [23] define context as the user's location, environment, identity and time. [10] define context as any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

One popular way to classify context instances is the distinction of different context dimensions. [24] call these dimensions external and internal, and [17] refer to physical and logical context. The external (physical) dimension refers to context that can be measured by hardware sensors, i.e., location, light, sound, movement, touch, temperature or air pressure, whereas the internal (logical) dimension is mostly specified by the user, for instance, user profile. Examples of Context Information are:

1. Identity, e.g. user profile, login information etc
2. Spatial information, e.g. location, orientation, speed, and acceleration
3. Temporal information, e.g. time of the day, date, and season of the year
4. Environmental information, e.g. temperature, air quality, and light or noise level
5. Social situation, e.g. who you are with, and people nearby
6. Resources that are nearby, e.g. accessible devices, and hosts
7. Availability of resources, e.g. battery, display, network, and bandwidth
8. Physiological measurements, e.g. blood pressure, heart rate, respiration rate, muscle activity, and tone of voice
9. Activity, e.g. talking, reading, walking, and running

2.2.2. Trends in context-aware computing

In the middle 1990's, the ubiquitous computing and the human-computer interaction communities started researching on context-aware systems, i.e. software that adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. The notion of context on the basis of these systems was then extended to cover a wider variety of aspects, such as environmental conditions (e.g., light, temperature), infrastructure (e.g., available networks, computational resources), and human factors (e.g., tasks, users).

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. [9]. Context-aware systems:

- Are able to adapt their operations to the current context without explicit user intervention
- Aim at increasing usability and effectiveness by taking environmental context into account

[18] and [23] define context-aware computing to be the ability of computing devices to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices. [37], [5], [8] define context-aware applications to be applications that dynamically change or adapt their behavior based on the context of the application and the user. [10] says a system is context-aware if it uses context to provide relevant information and /or services to the user, where relevancy depends on the users task.

2.2.3. Context Aware Services and Applications

The best known form of context awareness services is location awareness services, which is used to adapt services to the current location of a user [27]. [13] provide a taxonomy of Location Based Services (LBS), including emergency services (e.g., automotive assistance), navigation services (e.g., personal navigation), information services (e.g., mobile yellow pages), marketing services (e.g., mobile advertising), tracking services (e.g., vehicle tracking) and even billing services (e.g., location-sensitive billing). Some examples of location-based mobility services are: traffic and public transport information, location-based weather and search services. Another remarkably popular service is provided by real estate agents and shows houses on sale in the neighborhood of the user's current location. In-car personal navigation based on GPS location and advanced digital mapping technology is among the most popular applications. Personal navigation is moving from dedicated devices towards advanced mobile devices that are equipped with GPS functionality and sufficient storage and computational capacity. Context Aware applications fall into one of the following types that:

1. Automatically execute a service, e.g. Smart homes: turn off lights, adjust temperature, sending alerts and reminders
2. Present the information and services to a user, e.g. Tour guide, Active Badges
3. Tag the context to the information for later retrieval, e.g. Digital camera meta-data: time, location

Some existing Context Aware Applications:

- The Active Badge system [41] of the Olivetti Research Lab at the beginning of the 90's developed this system to locate Members of staff who wear badges that transmit signals providing information about their location to a centralized location service. The 'Active Badge' emits a unique code for approximately a tenth of a second every 15 seconds (a beacon). These periodic signals are picked up by a network of sensors placed around the host building. A master station also connected to the network, polls the sensors for badge 'sightings', processes the data, and then makes it available to clients that may display it in a useful visual form.
- Parc Tab [36] of Xerox Palo Alto Research Centre developed the PARCTAB Infra-red (IR) network which provides communication between stationary transceivers (base stations) and mobile systems.

Although the IR network was designed specifically for a hand held PDA called the PARCTAB (hence the name of the network), the network is used with other portable computing devices.

- In/Out board, a project of Georgia Institute of Technology (GeorgiaTech). This is the electronic equivalent of a simple in/out board that is commonly found in offices. The board is used to indicate which members of the office are currently in the building and those that are not. This is an example of a context viewing application. The system gathers information about the participants who enter and leave the building and displays the information to interested users. The context information is a participant's identity and the time at which they arrived or departed [34].
- DUMMBO (Dynamic Ubiquitous mobile meeting board), a project of Georgia Institute of Technology (GeorgiaTech). It is built to support automated capture of informal group gatherings. With contextual information about users and visualization Techniques, it can improve the user's ability to search a large repository of captured activity, infer structure, and salvage information from the repository. It supports group meetings that use a white-board. [4].
- StartleCam, developed at MIT media Lab: this is a wearable (i.e. worn by the user) video camera, computer and sensing system, which enables the camera to be controlled via both conscious and preconscious events involving the wearer. The system saves images when it detects certain events of supposed interest to the wearer. The aim is to capture events that are likely to get the user's attention and to be remembered. Attention and memory are highly correlated with what psychologist call aroused level, which is signaled by skin conductivity changes. The wearer's skin conductivity is monitored with the sensor. [16].
- Cyber guide project, developed at Georgia Tech: the Cyber guide project is a prototype of a mobile context-aware tour guide. It provides information according to the user's current location. The application uses Knowledge of the user's current location, as well as a history of past locations, are used to provide more of the kind of services that we come to expect from a real tour guide. [1].

2.3. Context Aware Business Processes

2.3.1. Relevance of Context to Business Processes

Business process context is the minimum set of variables containing all relevant information that impact the design and execution of a business process [31]. [30] emphasized on the need for adaptive business processes due to the trend towards decreasing time-to market and time-to-customer demands and an increasing frequency of product innovations combined with market changes such as globalization. Consequently, some flexibility is needed in current business processes. Business process flexibility can be seen as the capability of a process to respond to externally triggers by modifying only those aspects of a process that need to be changed and keeping other parts stable. Since many pervasive systems today support business processes, there is need to develop frameworks with appropriate adaptive mechanisms that can efficiently adapt business processes to context.

2.3.2 Practical Examples of Context Aware Business Processes

We are going to present some practical examples of context aware business processes by using some case studies. We shall start with the case studies presented by [42]

Context-sensitive pricing: the quotation that the oil distribution company sends to the customer depends on the context information (price level). The task of generating and sending quotation can be automated with the help of a web service that communicate with external data sources to retrieve the current price level. The calculation of the quotation depends on the retrieved price level.

Context-sensitive distribution: A DVD-rental company offers a web-based and an intra-shop electronic order system. Customers can place their order either directly using electronic terminals in the shop or they can order via the Internet. The system decides, based on the current location of the client, where the DVD will be delivered. Information on the current location of the customer could e.g. be provided by the cell phone company of the customer, which knows exactly where the cell phone (and the customer) is located. As it is unknown if the customer is currently in the shop or at home it is necessary to consider the context factor 'location of customer' in delivering the ordered items. In the situation where the customer is not in the shop, the items will have to be sent to him via a logistic company. The logistic company will only be contracted if the customer is not in the shop.

[25] presented two cases, the case of an insurance company and that of an airport. The context factor (weather) affects the claims process of an insurer in the short-term and its profitability in the long-term. Frequent natural disasters force the insurer to adapt its claims process to handle a large number of highly complex claims in a short period of time. The insurer responds to natural disasters by changing the set of questions asked by call centre consultants, by recruiting temporary staff, and by extending policy coverage or eligibility limits. The insurer balances customer experience, throughput, and risk exposure through a system of 'no-touch' (full automation), 'light-touch' (standard checks by clerk), and 'case-managed' (supervision by specialist) claims processes. In the case of an airport, passenger throughput at an airport depends on a wide range of external factors such as congestion in landside traffic to the airport, public holidays, transport innovation, geopolitical events (heightened security after terrorist attacks, natural disasters at destination), system failure and aircraft malfunction. These factors can turn the trickle of passengers into a torrent. The airport is able to scale operations, e.g., by staffing and deploying new security screening points. The system is able to rapidly adapt screening procedures to new threats (e.g., new security regulations in response to a terrorism threat).

2.4 Context Modeling

Context modeling of a context aware system is based on the data structure used for storing and exchanging context information. [40] provided a review of some of the most important context modeling approaches. Some of the approaches are:

- Key-Value Model: this is the most simple data structures for modeling the context information. In this approach, a key-value pairs is used to model the context by providing the value of context information (e.g. location information) to an application as an environment variable. In this case, the key is the environment variable while the value is the location information.
- Markup Scheme Model: this is based on the derivative of Standard Generic Markup Language (SGML), the super class of all markup languages such as the popular XML. Typical representatives of this kind of context modeling approach are profiles.
- Graphical Model: in this case context is models using graphical modeling languages such as Unified Modeling Language (UML)
- Object Oriented Model: in this approach, the main characteristics of object oriented modeling, such as encapsulation, inheritance and reusability is employed to solve some of the problems arising from the dynamics of the context in ubiquitous environments. The details of context processing is encapsulated on an object level and hence hidden to other components. Access to contextual information is provided through specified interfaces only.
- Logic Based Model: In a logic based context model, the context is consequently defined as facts, expressions and rules. [21] introduced contexts as abstract mathematical entities with properties useful in artificial intelligence.
- Ontology based Model: ontology formally represents knowledge as a set of concepts within a domain, and the relationships among those concepts. It can be used to reason about the entities within that domain and may be used to describe the domain. Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, knowledge representation etc. Ontologies are a very promising instrument for modeling contextual information due to their high and formal expressiveness and the possibilities for applying ontology reasoning techniques.

3. Related Work

In this research, we shall study existing context aware systems and frameworks.

3.1. Context Aware Architectures

[2] summarized the different architecture using different context model discussed in Section 2 (i.e. sub section 2.4). The various frameworks are explained in this section.

3.1.1 Context Management Framework Architecture:

[20] presented the architecture of the Context Management Framework. The entities that comprise this context framework are the context manager, the resource servers, the context recognition services, application, change detection service and security.

In this context framework, the entities, application, resource server, recognition service are the clients and they can add, subscribe to, and request context information.

- The context manager stores contexts and delivers responses and change notifications to the clients. The context manager stores context information from any source available to the terminal and serves it to clients. The context manager functions as a central server. The context manager, resource servers, and applications run on the mobile device itself, and the services are either distributed or local.
- The resource servers connect to any context data source and post context information to the context manager's blackboard.

Table 1 Sensor-based context ontology vocabulary example

Context type	Context value
Environment:Sound:Intensity	{ Silent, Moderate, Loud }
User:Activity:PeriodicMovement	{ FrequencyOfWalking, FrequencyOfRunning, NotAvailable }
Device:Activity:Stability	{ Unstable, Stable }

- Recognition service table registers plug-in context recognition services, which lets applications share the recognized higher-level contexts. The resource server and recognition service convert an unstructured raw measurement data flow into a representation defined in the context ontology, which permits serving the human-interpretable context information for the applications in an event-based manner. Table 1 gives an example of context ontology vocabulary, which is a high level representation of context values which an application can make use of. To facilitate ontology sharing and communication, the Resource Description Framework (RSF) (www.w3c.org) is used for syntax description. The change detection services facilitate alternative ways to detect context change. The security module checks the trustworthiness of incoming contexts, which is especially important for contexts received from outside the terminal.

3.1.2 The Service-Oriented Context-Aware Middleware (SOCAM)

[14] introduced architecture for building context aware mobile services. It uses a central server as well, which is called context interpreter. It gains context data through distributed context providers and offers it in processed form to the clients. The context is modeled with Web Ontology Language (OWL), which

is an ontology markup language that enables context sharing by explicitly defining contexts in a semantic way.

The context ontology is divided into the common upper ontology for the general concepts and the domain-specific ontology which apply to different sub domains. The generalized ontology captures general contexts for all domains. The generalized ontology is fixed once defined and will be shared among different domains. The domain-specific ontology is a collection of low-level ontologies which define the details of general concepts and their properties in each sub domain. Adaptation can take place by the low-level ontology for each sub-domain dynamically rebinding with the generalized ontology when the environment is changed. For example, if a user is at home, the home-domain ontology is bound with the generalized ontology and used to derive high-level contexts.

We now describe the components of the SOCAM architecture provided by [14]. The SOCAM architecture consists of the following components:

- Context providers. They abstract useful contexts from heterogeneous sources - External or Internal; and convert them to OWL representations so that contexts can be shared and reused by other service components. The context sensing is either distributed by the use of web services or by sensors in the mobile device.
- Context interpreter. It provides logic reasoning services to process context information. It is a central server which gains context data through distributed context providers
- Context database. It stores context ontologies and past contexts for a sub-domain. There is one logic context database in each domain, i.e. home domain.
- Context-aware services. They make use of different level of contexts and adapt the way they behave according to the current context.
- Service locating service. It provides a mechanism where context providers and the context interpreter can advertise their presence; it also enables users or applications to locate these services.

[15], in the *Context-Aware Service Oriented Systems* (CASOS) also proposed an architecture similar to SOCAM.

3.1.3 Context-Awareness Sub-Structure (CASS)

[12], in the Context-Awareness Sub-Structure (CASS) provided an extensible centralized middleware approach designed for context-aware mobile applications. The middleware contains an Interpreter, a ContextRetriever, a Rule Engine and a SensorListener. The SensorListener listens for updates from sensors, which are located on distributed computers called sensor nodes. Then the gathered information is stored in the database by the SensorListener. The ContextRetriever is responsible for retrieving stored context data. Both of these classes may use the services of an interpreter. The ChangeListener is a component with communication capabilities that allows a mobile computer to listen for notification of context change events. Sensor and LocationFinder classes also have built-in communications capabilities. Mobile clients connect to the server over wireless networks. To reduce the impact of intermittent connections, local caching is supported on the client side. CASS enables developers to overcome the memory and processor constraints of small mobile computer platforms while supporting a large number of low-level sensor and other context inputs.

The database contains the knowledge base. The knowledge base contains the rules used by the inference engine described above. CASS stores application knowledge bases as database tables where the records contain high-level context states and corresponding context behavior, the goal. Table 2 shows an example of a weather context rule from [12]. A tour-guide application might use such a rule to allow it to display hyperlinks to indoor activities. In this case the goal 'indoor', could be used as a key into a table containing the hyperlinks.

The CASS inference engine is used to find a matching goal or goals when a change in context is detected. Having a separate knowledge base means that changes can be made to context inferences and goals relevant to an application without necessitating changes in application code.

Table 2 Example of weather context rule

Rain	Brightness	Temperature	Goal
wet	dull	cold	Indoor

3.1.4 Context Broker Architecture (CoBrA)

Context Broker Architecture (CoBrA) [6] is an agent based architecture for supporting context-aware computing in intelligent spaces. Intelligent spaces are physical spaces (e.g., living rooms, vehicles, corporate offices and meeting rooms) that are populated with intelligent systems that provide pervasive computing services to users. A very prominent part of CoBrA is an intelligent context broker that maintains and manages a shared contextual model on behalf of a community of agents. These agents can be applications hosted by mobile devices that a user carries or wears, services that are provided by devices in a room (e.g. light controller and room temperature controller) and web services that provide a web presence for people, places and things in the physical world (e.g. services keeping track of peoples and objects.). This architecture explores the use of Semantic Web languages for defining and publishing context ontology, for sharing information about a context and for reasoning over such information. CoBrA implements security and privacy by including a policy language that allows users and devices to define rules to control the use and the sharing of their private contextual information. Using this language, the users can protect their privacy by granting or denying the system permission to use or share their contextual information. The context broker consists of four functional main components: the Context Knowledge Base, the Context Inference Engine, the Context Acquisition Module and the Privacy Management Module.

3.1.5 The Context Toolkit

The Context Toolkit ([34], [9], [10]), a context-aware framework, has the following components: widgets, interpreters, aggregators, discoverers, services. It is a peer-to-peer architecture with a centralized discoverer where distributed sensor units (called widgets), interpreters and aggregators are registered in order to be found by client applications. The toolkits object-oriented API provides a super class called BaseObject which offers generic communication abilities to ease the creation of its own components. A context widget is a software component that provides applications with access to context information from their operating environment. This framework has four additional categories of components - interpreters, aggregators, services and discoverers.

Interpreters: Interpretation refers to the process of raising the level of abstraction of a piece of context. Location for example may be expressed at a low level of abstraction, such as geographical coordinates or at higher levels such as street names. Simple inference or derivation transforms geographical coordinates into street names using for example a geographic information database.

Aggregators: The need for aggregation comes in part from the distributed nature of context information. Aggregators gather logically related information relevant for applications and make it available within a single software component.

Services: The context service is responsible for controlling or changing state information in the environment using an actuator (i.e. output). Services are components in the framework that execute actions on behalf of applications.

Discoverers: They are responsible for maintaining a registry of what capabilities exist in the framework. This includes knowing what widgets, interpreters, aggregators and services are currently available for use by applications. When any of these components are started, it notifies a discoverer of its presence and capabilities and how to contact that component. When any component fails, it is a discoverer's responsibility to determine that the component is no longer available for use.

3.1.6 Hydrogen project (Hydrogen's object-oriented approach)

In the Hydrogen project, [17] provided a framework based on a layered architecture. Its context acquisition approach is specialized for mobile devices. The Hydrogen system tries to avoid dependency on centralized components, unlike other frameworks. The architecture consists of three layers which are all located on the same device. The Adaptor layer is responsible for retrieving raw context data by querying sensors. This layer permits a sensor's concurrent use by different applications. The second layer, the Management layer, makes use of the Adaptor layer to gain sensor data and is responsible for providing and retrieving contexts. The Context server offers the stored information via synchronous and asynchronous methods to the client applications. On top of the architecture is the Application layer, where the application code is implemented to react on specific context changes reported by the context manager. To guarantee platform and language independency, all inter-layer communication is based on a XML-protocol.

It distinguishes between a remote and a local context. The remote context is information another device knows about, the local context is information one device is aware of. When the devices are in physical proximity they are able to exchange these contexts in a peer-to-peer manner via WLAN, Bluetooth, etc. This exchange of context information among client devices is called context sharing. Both local and remote context are made up of context objects. The super class ContextObject is extended by different context types, e.g., LocationContext, DeviceContext, etc. This approach allows the simple addition of new context types by specializing ContextObject. A context type has to implement the methods toXML() and fromXML() from the ContextObject class in order to convert the data from and to a XML stream.

3.2. Existing Approaches for Adaptive Process Modeling

Currently there is a proliferation of business information systems such as Enterprise Resource Planning (ERP) systems, the current generation of which is known under the label of process-aware information systems [11]. To dynamically adapt a process to a new context is not a trivial problem. Also to formally represent this context in a process model and also represent the adaptation is also a serious research field. Several approaches have emerged for adaptive and/or flexible process designs that are able to cope with changes that may occur during the lifetime of a business process.

[39] suggested an approach to support process flexibility through the use of web services. When designing a schema representation, the aspects of the business process have to be identified and separated. Five basic aspects of business processes are identified: the functional, control, informational, organizational and operational aspect. These "core-aspects" are orthogonal dimensions of business processes. Aspect elements are no further dividable, atomic parts of business processes which contain only functionality of one aspect. If aspect elements are applied as granularity for a schema representation, a so-called aspect element oriented schema representation is created. Web Services provide the universal and transparent access to asynchronously evolving services in heterogeneous environments by using near ubiquitous internet technologies such as HTTP and XML. The use of web services as implementation base helps to fulfill the requirement of service extensibility and integration because they allow for the transparent integration of services across heterogeneous platforms. Furthermore a discovery mechanism such as UDDI allows for the integration of unknown services and extends the set of available services. Different versions of a web service can be differentiated by different name spaces. Therefore web services fulfill also the requirement of asynchronous evolution of services. The adaptation of web services to an individual composite application is done by applying specialization information to the services: the web service is parameterized and connected to other web services depending on the requirements of the composite application. Therefore, the specialization information contains both parameterization and connection information. The parameterization information adapts the web service to the individual needs of the composite application. For different composite applications there is different specialization information.

[30] developed a process reference modeling technique that supports adaptability by extending traditional techniques with variation points. They proposed an extension of EPC (Event-driven Process Chain) modeling notation provided by [35],

called C-EPC (Configurable Event driven Process Chain), through the representation of variation points called "configurable nodes" to which variants or alternatives can be associated. Besides, restrictions (called "configurations requirements" and guidelines) may be applied in order to guide or restrict possible adaptations. One remarkable observation in this approach is that the reference model is not able to undergo dynamic autonomous reconfiguration.

[26] in PESOA (Process Family Engineering in Service-Oriented Applications) project introduced the Variability mechanisms which allows for the derivation of artifact variants from generic artifacts. While the derived artifact variant is typically specific for a concrete member of the product line, the generic artifact has features, which are common for more than one member of the product line. The PESOA project establishes the concept of variant-rich process model that represents an extended process model with the use of annotations and stereotypes so as to introduce variability in the model. Stereotypes are used to represent, for example, in the variation points, what can vary (<<VarPoint>>), what is optional (<<Optional>>) or abstract (<<abstract>>). [3] proposed the use of projection models, where it starts with a reference model that covers all possibilities and from which it is possible to create a projection for a specific scenario by eliminating the paths that are not relevant.

Another way to represent flexibility in processes may be through business rules because: (i) they can increase dynamicity in processes by providing a better support for flexibility in process adaptation at different levels of granularity and for various process elements (not only the flow activities) [22] (ii) it is independent of a specific process modeling notation and, given a semantically rich representation language, the translation to process modeling notations (mainly for viewing) represents a small effort; (iii) the representation of business rules through formal languages supports process reasoning and the verification of correctness [22]. Other research has proposed to extend traditional process modeling approaches with some contextual information. [29], for instance, showed how risk modeling can be integrated with event-driven process chains, and [28] extended the scope of process models to include regulatory perspectives by means of use and misuse cases.

4. Hybrid Architecture

4.1. Architectural Components

This system addresses the issue of business process adaptation to context situations. The principal advantage of our approach is the independence of context data from business one and the independence of context-aware components from business ones. By the use of this development concept we provide reuse, interoperability and adaptability to context-aware applications. Another advantage of this architecture is that we adapt existing well established n-tier enterprise solutions for our context aware system, and it allows for interoperability with existing business enterprise solutions and since most developers are more familiar with these more established business solutions, there is a relative ease in the development of this context aware system compared with other context aware systems. In this system, behavior changes can be realized by a new process start, reconfiguration of some parts of a process, a new service invocation or a new interface creation.

This system comprises of centralized components: process manager, context reasoning machine, model, application server; distributed SOA sensors; and the client application. Most of context-aware applications blend context information and context-aware activities with business logic, but in this architecture there is separation of concerns. Context models are defined independently of business logic models. A preliminary version of the architecture with its subsystems, modules, and databases is illustrated in Figure 3: Context model: context is handled in simple attribute-value-tuples, which are encoded using XML for transmission. The components of this architecture are:

- SOA Sensors (Distributed web services): web services are responsible for communication with distributed sensors. We believe that standardized technologies and protocols, such as Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP), could help to build more interoperable context aware services. By using web services, we take advantage of the inherent interoperability, scalability and low coupled characteristics provided by web services. This allows for simple integration among different applications and distributed software components. SOA sensors are responsible for:
    a. Capture of context information from distributed sensors: Sensing is done by web services, and the system does not know how the sensing is done. Application is separated from context acquisition concerns. Resource discovery is done as is done in discovering of web services in service oriented architecture (SOA). This is done with well established Universal Description, Discovery and Integration (UDDI) registry. The complexity of sensing is hidden and sensing web services are reused by many applications and process managers. Resource discovery mechanisms are rarely used in the frameworks discussed so far. Resource discovery mechanisms are important in pervasive environments; because sensors change rapidly (new ones are added or removed and some sensors may also malfunction and thus making it not to be available). The systems presented so far assume that the used context sources are stable and permanently available, which is not the case in real world applications. With sensing web services, when a sensor is not available, the sensing web service can connect with other available sensors.
    b. Interpretation of context information: context interpretation refers to the transformation of context data including special knowledge. For instance, raw temperature data is interpreted as Celsius of Fahrenheit.

c. Aggregation of context information: context aggregation refers to the composition of context atoms either to collect all context data concerning a specific entity or to build higher-level context objects. These forms of context data abstraction ease the application designer's work tremendously.

- Context Database: this is a centralized database. Sometimes it might be necessary to have access to historical context data. Such context histories may be used to establish trends and predict future context values. As most data sources constantly provide context data, the maintenance of a context history is mainly a memory concern, so a centralized high-resource storage component is needed. This context database can support many client machines at the same time.
- Context Reasoning Machine: This component has the facility to query historical context data. This component implements a context learning algorithm that makes use of historic data in the database to deduce new context situation. The process manager makes use of this deduced context by adapting business process to the deduced context.
- Model: this is a pool of domain objects and each object represents an atomic level activity of a business process model. These activity objects are no further dividable and they are atomic parts of business processes which contain only functionality of one aspect. [39] designed something similar to this as web services, but in our own case, the activity elements of the model are not implemented directly as web services. The model is designed as a central component and the activity elements can then access web service and distributed databases needed for transactions. The order in which these activities are executed is not fixed. The order in which the activities are executed depends on the schema representation provided by the process manager. When the schema representation in the process manager is reconfigured based on the context, the order of execution of the activities is also reconfigured. The activity elements are parameterized and they have inputs and output.
- Process Manager: Most of the systems did not adequately address the issue of business process adaptation to context situations. The process manager is responsible for business process adaptation and it is very important to this architecture. Its duty is to reconfigure a business process base on the prevailing context. The prevailing context is dependent on the context data supplied by the sensing web services or the context reasoning machine. We used the idea of schema representation proposed by [39] and adapted it to our on purpose. The process manager contains schema representation of business processes. The elements of the schema representation must be connected in a reconfigurable manner, so that extensions and changes to the schema representation can be implemented dynamically. The schema representation enacts the activity elements (business process) elements in the model. To improve flexibility of schemas, service extensibility and integration, the partitioning of the schema representation is done. The partitioning is done at the variation points of the schema where a change in a part of a business process occurs due to change in context. A variant schema partition is provided to be executed in a specified context situation. The schema representation has the information on how activity elements will be connected to each other based on the context information. For a different schema, there is different specialization information. The specialization information contains both parameterization and connection information. The process manager is event-based, so that there is no requirement on applications to poll for changes in context.
- Application Server: existing application servers used for enterprise solutions is adopted. This is the server for server side scripts like Java Servlets/JSP, ASP/ASP.NET, PHP and also web servers for html and xml.
- Client Application: this comprises java application and web view.

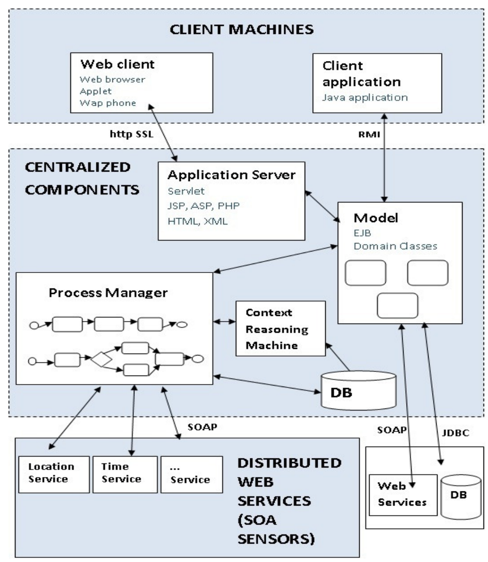

Figure 1 Hybrid Architecture

5. Future Work and Conclusion

More work will be done on better ways of achieving process adaptability in response to context changes, e.g. the investigation on the use of aspect oriented programming in developing process adaptive mechanisms. Research will also be done on the application of semantic web ontologies in modeling contexts and the application of knowledge mining techniques in inferring knowledge from historical context data. Further research will be done on the use of ontologies for building a context model as is demonstrated in existing frameworks to build more sophisticated algorithms, which derive new contextual knowledge or patterns to proactively aggregate new context-aware services.

6. References

[1] Abowd G. D., Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper and Mike Pinkerton; 'Cyberguide: A Mobile Context-Aware Tour Guide'. 1997

[2] Baldauf M. and Schahram Dustdar and Florian Rosenberg; 'A survey on contextaware system', Int. J. AdHoc and Ubiquitous Computing, Vol. 2, No. 4. 2007

[3] Becker J., P. Delfmann and R. Knackstedt; 'Adaptive Reference Modeling: Integrating Congurative and Generic Adaptation Techniques for Information Models', Proceedings of the Reference Modeling Conference, pp. 27-58, 2007.

[4] Brotherton J. A., Abowd G. D., Truong K. N.; 'Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings', Paper submitted to CHI'99. 1999

[5] Brown P. J., Bovey, J. D. and Chen, X.; 'Context-aware applications: From the laboratory to the marketplace'. IEEE Personal Communications, 4(5), 58-64, 1997

[6] Chen, H.; 'An Intelligent Broker Architecture for Pervasive Context-Aware Systems', PhD Thesis, University of Maryland, Baltimore County, 2004

[7] Chen, H., Finin, T. and Joshi, A.; 'An ontology for context-aware pervasive computing environments', Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Vol. 18, No. 3, pp.197-207, 2004

[8] Davies N., Mitchell, K., Cheverst, K. and Blair, G.; 'Developing a context-sensitive tour guide'. Proceedings of the 1st Workshop on Human Computer Interaction for Mobile Devices, Glasgow, Scotland, 1998

[9] Dey Anind K. and Abowd Gregory D.; 'Towards a Better Understanding of Context and Context-Awareness'. In the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands, April 3, 2000. Also GVU Technical Report GIT-GVU- 99-22. Submitted to the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999.

[10] Dey, A.K. and Abowd, G.D.; 'A conceptual framework and a toolkit for supporting rapid prototyping of context-aware applications', Human-Computer Interactions (HCI) Journal, Vol. 16, Nos. 2-4, pp.7-166. 2001

[11] Dumas, M., van der Aalst, W. M. P., and ter Hofstede, A. H. M. (Eds.).; 'Process Aware Information Systems: Bridging People and Software Through Process Technology'. Hoboken, New Jersey: John Wiley and Sons. 2005

[12] Fahy, P. and Clarke, S.; 'CASS - a middleware for mobile context-aware applications', Workshop on Context Awareness, MobiSys 2004.

[13] Giaglis G. M., Zeimpekis V. And Lekakos G.; 'A Taxonomy of Indoor and Outdoor Positioning Techniques for Mobile Location Services', 2002

[14] Gu, T., Pung, H.K. and Zhang, D.Q.; 'A service-oriented middleware for building context-aware services', Journal of Network and Computer Applications 28 (2005) 1-18, 2004

[15] Hatim Hafiddi, Hicham Baidouri, Mahmoud Nassar and Abdelaziz Kriouile; 'Context-Awareness for Service Oriented Systems', IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012 ISSN (Online): 1694-0814 www.IJCSI.org

[16] Healey J. and Picard R. W.; 'StartleCam: A Cybernetic Wearable Camera', ISWC proceeding of 2nd IEEE International Symposium on Wearable Computers, page 42 ,1998

[17] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G. and Altmann, J.; 'Context-awareness on mobile devices the hydrogen approach', Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pp.292-302. 2002

[18] Hull, R., Neaves, P. and Bedford-Roberts, J.; 'Towards situated computing', Proceedings of the First International Symposium on Wearable Computers (ISWC '97), p.146. 1997

[19] Johansson, Henry J.; 'Business Process Reengineering: BreakPoint Strategies for Market Dominance'. John Wiley and Sons, 1993

[20] Panu Korpipää, Jani Mäntyjärvi, Juha Kela, Heikki Kernen, and Esko-Juhani Malm; 'Managing Context Information in Mobile Devices'; Pervasive Computing; Published by the IEEE CS and IEEE ComSoc, 2003

[21] McCarthy, J. and Buvac, S.; 'Formalizing context (expanded notes)', in Buvac, S. and Iwahska, L. (Eds.): Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language, California. American Association for Artificial Intelligence, Menlo Park, pp.99-135. 1997

[22] Mejia Bernal J. F., P. Falcarin, M. Morisio and J. Dia; 'Dynamic context-aware business process: A rule-based approach supported by pattern identification'. In: Proceedings of the ACM Symposium on Applied Computing, pp. 470 - 474, Sierre, Switzerland, 2010.

[23] Pascoe J., Ryan, N. S. and Morse, D. R.; 'Human-Computer-Giraffe Interaction - HCI in the field'. Proceedings of the Workshop on Human Computer Interaction with Mobile Devices, Glasgow, Scotland. 1998

[24] Prekop, P. and Burnett, M.; 'Activities, context and ubiquitous computing', Special Issue on Ubiquitous Computing Computer Communications, Vol. 26, No. 11, pp.1168-1176. 2003

[25] Ploesser Karsten, Recker Jan, Rosemann Michael; 'Challenges in the Context-Aware Management of Business Process: A Multiple Case Study'; ECIS 2011 Proceedings. Paper 6

[26] Puhlmann F., Schnieders A., Weiland J. D. and Weske M.; 'Variability mechanisms for process models' PESOA-Report No.17/2005, Hasso Plattner Inst. Acessed in: frapu.de/pdf/PESOA TR 17-2005.pdf, 2005.

[27] Rapera J., Gartnerb G., Karimic H. and Rizos C.; 'Applications of location-based services: a selected review', Journal of Location Based Services, Vol. 1, No. 2, June 2007, 89-111

[28] Regev G., Bider I. and Wegmann A.; 'De_ning business process flexibility with the help of invariants', Software Process: Improvement and Practice, v. 12, n. 1, pp. 65-79, 2007.

[29] Rosemann Michael and zur Muehlen; 'Michael Integrating Risks in Business Process Models', 16th Australasian

Conference on Information Systems, 29 Nov - 2 Dec 2005, Sydney

[30] Rosemann, M., Recker, J. and Flender, C.; 'Contextualization of Business Processes', Int. J. Business Process Integration and Management, Vol. 1, Nos. 1/2/3, pp. 111. 2007

[31] Rosemann Michael, Recker Jan; 'Context-aware Process Design: Exploring the Extrinsic Drivers for Process Flexibility'; In Latour, Thibaud & Petit, Michael (Eds.) *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium*, Namur University Press, pp. 149-158. 2006

[32] Rosemann, M., van der Aalst, W. M. P. ; 'A Configurable Reference Modelling Language'. Information Systems. 2005

[33] Ryan, N., Pascoe, J., Morse, D.; 'Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant'. Gaffney,V., van Leusen, M., Exxon, S. (eds.) Computer Applications in Archaeology (1997)

[34] Salber, D., Dey, A.K. and Abowd, G.D.; 'The context toolkit: aiding the development of context-aware applications', Proceedings of the ACM CHI, Pittsburgh, PA, pp.434441. 1999

[35] Scheer A.; 'ARIS - business process modeling', Springer, 2000.

[36] Schilit B. N. , Adams N. , Gold R. , Michael M. Tso , Roy Want; 'The PARCTABMobile Computing System', Palo Alto Research Center, Xerox Corporation. 1993

[37] Schilit B., Adams, N. and Want, R.; 'Context-aware computing applications'. Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, 85-90. Los Alamitos, CA: IEEE. 1994

[38] Schilit, B. and Theimer, M.; 'Disseminating active map information to mobile hosts', IEEE Network, Vol. 8, No. 5, pp.22-32. 1994

[39] Schmidt, R.; 'Flexible Support of Inter-Organizational Business Processes Using Web Services'. In: Castro, J., Teniente, E. (eds.): Proceedings of the CAiSE'05 Workshops. Vol. 1. FEUP, Porto, Portugal (2005) 51-58

[40] Strang, T. and Linnhoff-Popien, C.; 'A Context Modeling Survey', First International Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp. 2004

[41] Want, R., Hopper, A., Falcao, V. and Gibbons, J.; 'The active badge location system', ACM Transactions on Information Systems, Vol. 10, No. 1, pp.91-102. 1992

[42] Daniel Wagner, Otto K. Ferstl; ‘Enhancing Flexibility and Precision of Enterprise Architectures by Modeling Context-Aware Business Processes’; Bayerischer Forschungsverbund forFLEX – Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse. 2009

[43]Wikipedia, http://en.wikipedia.org/wiki/Business process

Henry Eleonu received a bachelor’s degree in computer science from Michael Okpara University of Agriculture, Umudike (MOUAU) in 2005. He received his MSc in Computer Science from African University of Science and Technology, Abuja in 2010. He is a Systems Analyst at Abia State Planning Commission, Umuahia. His research interests are Context-Aware Systems, Service-Oriented Architecture, Software Architectures and Self-Adaptive Systems.

Jane Oruh received a bachelor’s degree in computer science from Michael Okpara University of Agriculture, Umudike (MOUAU) in 2005. She just completed her M.Sc in computer science from Ebonyi State University, Abakaliki. She is presently a teaching staff with the Computer Science Department of Michael Okpara University of Agriculture, Umudike, Nigeria. Her research interests are Database Security, biometric authentication systems and context aware systems.