MONITORING SERVICE SYSTEMS FROM A LANGUAGE-ACTION PERSPECTIVE

ABSTRACT

The Exponential growth in the global economy is being supported by service systems, realized by recasting mission-critical application services accessed across organizational boundaries. Language-Action Perspective (LAP) is based upon the notion as proposed that "expert behavior requires an exquisite sensitivity to context and that such sensitivity is more in the realm of the human than in that of the artificial.

Business processes are increasingly distributed and open, making them prone to failure. Monitoring is, therefore, an important concern not only for the processes themselves but also for the services that comprise these processes. We present a framework for multilevel monitoring of these service systems. It formalizes interaction protocols, policies, and commitments that account for standard and extended effects following the language-action perspective, and allows specification of goals and monitors at varied abstraction levels. We demonstrate how the framework can be implemented and evaluate it with multiple scenarios like between merchant and customer transaction that include specifying and monitoring open-service policy commitments.

EXISTING SYSTEM:

While the existing system was most of our large software systems are now constructed as group of interoperating systems made to interoperate through various forms of interfaces.

Instead, the open and distributed process includes tasks that are performed by services that are not centrally controlled, and hence, can be unpredictable. As a result, service outcomes themselves tend to be uncertain.

Disadvantage:

- ❖ Although, we can conceive these large service systems, we have trouble building them.
- ❖ Anecdotal accounts of systems integration efforts points to failures and underscore the need to address the underlying cause:
- ❖ Rapidly Changing Environmental forces.

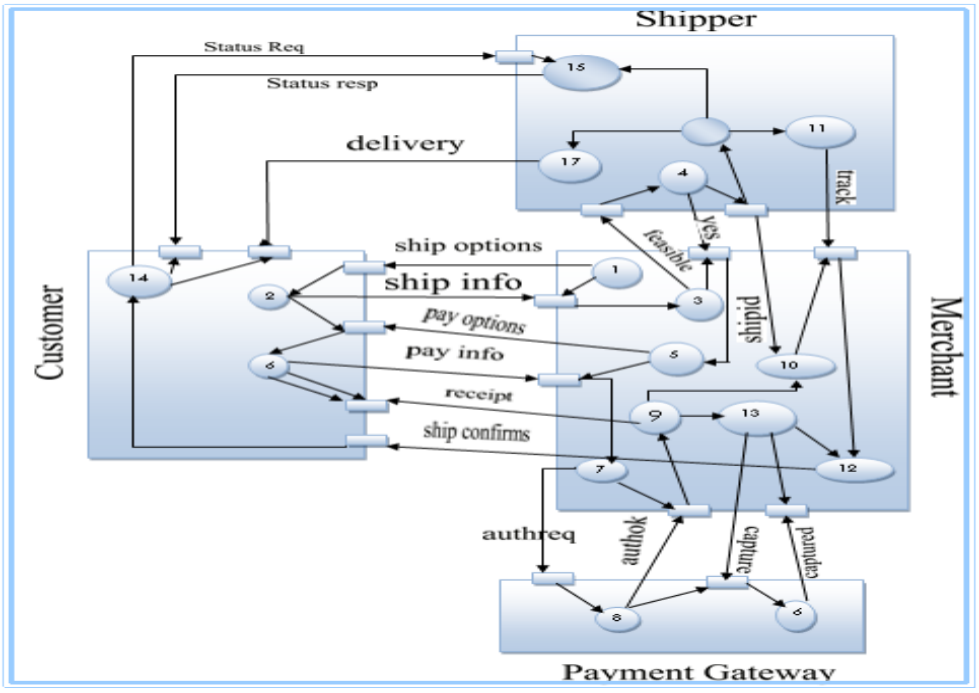
PROPOSED SYSTEM:

- ✓ The goal of this research is to develop a framework for monitoring such service systems. To establishes its feasibility, and evaluates it with scenarios and comparisons against existing proposals.
- ✓ It monitoring the software systems is closely tied to the monitoring of requirements and how they are realized in software.
- ✓ If observes and analyzes the behavior of another (target) system, that determining qualities of interest, such as satisfaction of the target system's requirements.

ADVANTAGE:

- ✓ Where well-specified business processes with service to perform tasks are advanced changes such as stability, feasibility, predictability.
- ✓ We develop the contribution as an extension to prior work in monitoring, establish its feasibility and demonstrate it with multiple scenarios.
- ✓ Service systems build upon the basic idea of software as a service that allows legacy and new applications to declare their capabilities in a machine-readable format.

ARCHITECTURE:



MODULES:

- Customer Details

- Merchant Details
- Loan Approved
- Shipping Process
- Payment Gateway

Customer Details:

In this module the details about various Customers are maintained and each has a unique id. And includes about various Merchants are maintained and each has a unique id. The Customer select best merchant and then send quote to that merchant Mail account. And also send his particulars and conditions.

Merchant Details:

As same as, in this module the details about various Merchants are maintained and each has a unique id. The merchants receives item ID, consults its policy for quoting and sends the quote price itemprice, and receives either accept-quote or reject-quote.

Loan Transaction:

In this module the authentication of the Customer's to be checked their quotation and items are monitored. Then, Loan Approver Confirm his account and quotation to be authorized/unauthorized. If, the customer to be authorized person the loan to be sanction to that specific customer. When unauthorized customer loan to be rejected.

Shipping Process:

The Shipper who transports the physical items to the Customer from Merchant within the given date. And they know the conditions and commitments of both customer and merchants. He gets the orders from merchants and sends to that particular customer.

Payment Gateway:

Merchants who authorize the Customer Payments for his Purchase order. The customer who pays the money to his account. After payer sends payment information, eventually payer receives receipt. After payer commits to order, payer pays for order.

Algorithm:

LINEAR TEMPORAL LOGIC (LTL) ALGORITHM:-

It can be used to prove properties when the set of states reachable from an initial state in a system module is finite.

We can distinguish two levels of specification:

- A *system specification* level, provided by the rewrite theory specified by that system module which defines the behavior of the system, and
- A *property specification* level, given by some property (or properties) φ that we want to state and prove about our module.

An interaction protocol is defined as the "Rules of Engagement" among interacting participants. Therefore, include:

1) Specificity
2) Semantic
3) Composability

In Specificity represent interactions among services. That is domain independent. (e.g. for shipment , may be domain-dependent.)
In Semantic concern is semantic content. i.e., the nature of the interactions. (Request – Reply mechanism). (e.g., destination, payment info) to capture content. The message itself may capture the semantics to distinguish actions, such as direct, inform, cancel, and others.
Finally, composability is concerned with how a designer might specify the process by combining protocols.

System Requirements:

Hardware Requirements:

- System : Pentium Iv 2.4 GHz
- Hard Disk : 80 GB
- Monitor : 15 VGA Color
- Mouse : Logitech
- Ram : 512 Mb

Software Requirements:

- Operating System : Windows XP
- Coding Language : Asp.Net With C#
- Data Base : Sql Server 2005

REFERENCE:

William N. Robinson, and Sandeep Purao, "Monitoring Service Systems from a language-action perspective", **IEEE Transactions on service computing, Vol.4, No.1, January-March 2011.**