INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION AND MANAGEMENT

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing, Human Resource Management, General Management, Banking, Insurance, Corporate Governance and emerging paradigms in allied subjects like Accounting Education; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Monetary Policy; Portfolio & Security Analysis; Public Policy Economics; Real Estate; Regional Economics; Tax Accounting; Advertising & Promotion Management; Business Education; Business Information Systems (MIS); Business Law, Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labor Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; Public Administration; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism, Hospitality & Leisure; Transportation/Physical Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Digital Logic; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Multimedia; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic and Web Design. The above mentioned tracks are only indicative, and not exhaustive.

Anybody can submit the soft copy of his/her manuscript **anytime** in M.S. Word format after preparing the same as per our submission guidelines duly available on our website under the heading guidelines for submission, at the email addresses: **1** or info@ijrcm.org.in.

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR

IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF ____.**

(e.g. Computer/IT/Engineering/Finance/Marketing/HRM/General Management/other, please specify).

DEAR SIR/MADAM

Please find my submission of manuscript titled '____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication anywhere.

I affirm that all author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:

Designation:

Affiliation with full address, contact numbers & Pin Code:

Residential address with Pin Code:

Mobile Number (s):

Landline Number (s):

E-mail Address:

Alternate E-mail Address:

2. **MANUSCRIPT TITLE**: The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.
3. **AUTHOR NAME (S) & AFFILIATIONS**: The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.
4. **ABSTRACT**: Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.
5. **KEYWORDS**: Abstract must be followed by list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT**: Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of the every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS**: All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS**: All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT**: The main text should follow the following sequence:

   INTRODUCTION

   REVIEW OF LITERATURE

   NEED/IMPORTANCE OF THE STUDY

   STATEMENT OF THE PROBLEM

   OBJECTIVES

   HYPOTHESES

   RESEARCH METHODOLOGY

   RESULTS & DISCUSSION

   FINDINGS

   RECOMMENDATIONS/SUGGESTIONS

   CONCLUSIONS

   SCOPE FOR FURTHER RESEARCH

   ACKNOWLEDGMENTS

   REFERENCES

   APPENDIX/ANNEXURE

   It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed 5000 words.

10. **FIGURES &TABLES**: These should be simple, centered, separately numbered & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS**: These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES**: The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use **(ed.)** for one editor, and **(ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio," Ohio State University.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITE

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on December 17, 2011 http://epw.in/user/viewabstract.jsp

AN OPTIMAL BROKER-BASED ARCHITECTURE FOR TRANSACTIONAL AND QUALITY DRIVEN WEB SERVICES COMPOSITION

KAVYA JOHNY
M. TECH. STUDENT
DEPARTMENT OF COMPUTER SCIENCE
VISWAJYOTHI COLLEGE OF ENGINEERING & TECHNOLOGY
VAZHAKULAM

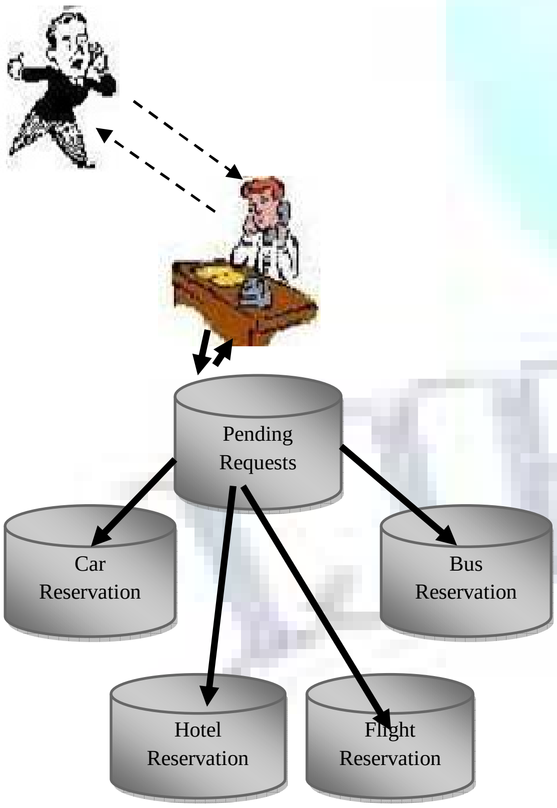
ABSTRACT
Web service composition consists of combining web services that supports business-to-business or enterprise application integration to offer more complex services. While performing web service composition, the selection of appropriate web service for each activity in the workflow from the discovered services that satisfy a given requirement has become an important problem. However, all the existing broker based architectures consider only QoS factors for web service composition. They do not consider the transactional constraints during the composition process. This paper proposes and implements a broker-based framework for web service composition not only according to their QoS characteristics but also to their transactional properties and thereby facilitates dynamic integration of atomic web services. This is verified for various inputs and results shows that the broker provides a better web service composition.

KEYWORDS
Web services, quality of service, transactional web service, broker based architectures, workflow, web service composition.

INTRODUCTION
Web services provide ready-to-use functionalities through fixed interfaces for other applications by hiding the implementation details and they are commonly used in real world applications. There exist a huge number of web services that can handle particular requests. With the increasing number of available web services, maintaining these services and searching for the ones that satisfy a given requirement has become an important problem. In order to handle a complex request, a combination of more than one service is required. This process of combining web services to achieve complex tasks is called Web Service Composition (WSC).

Consider the example of travel agency as shown in Figure. 1. A customer calls a travel agent who notes down the customer requests. He generates a trip request that consists of many hotel and flight reservations etc. The travel agent performs all bookings by sending the generated requests to specialized services which work independently of each other. If both the services have been completed successfully, the confirmation process put all the documents together and informs the customer. This is in the case of closed world scenario. In the open world scenario, there will be a number of processes trying to achieve the same goal. The Web service composition is done to integrate all the required services together to provide successful execution of the request from the customer.

FIGURE 1.TRAVEL AGENCY



The Web service composition mainly includes three steps: 1) Web service discovery, 2) selection of the component Web services, and 3) execution of the composite Web services.

A. Web Service discovery

There are several thousand services, which can be used for composing required applications. However, for composing services for the application, these services must first be discovered. Based upon the description given by the customer, the broker based architectures discovers available web services for each activity in the work flow. The UDDI of Web services provides a more powerful approach to model and search Web services.

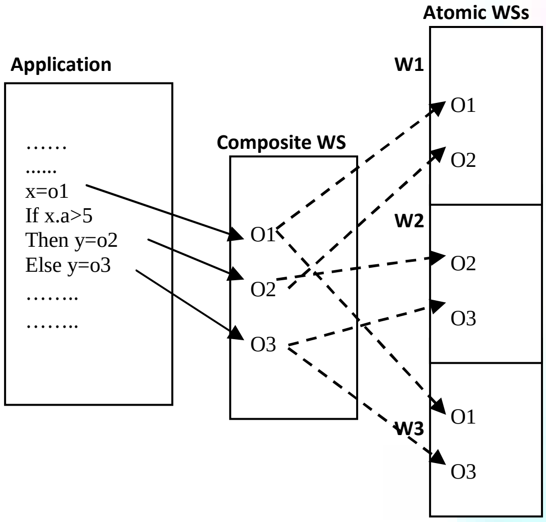
B. Web Service selection

The service selection is the step of deciding which service has to use to finally perform the task. For instance there may be services performing similar goals and users can select the one that is closest to the customer intention. However, different services that perform the same goal, may offer different objects, at different costs, quality levels, etc. The Service providers provide web services with the QoS specification. The consumers specify the service that is to be served with their own QoS requirements. So the need to select the appropriate web services becomes more important during the second step of web services composition.

Generally, when a client application invokes operations in a composite WS, the composite WS executes these operations by serving as a broker that delegates the operations to some atomic WSs. The operation invocations between the application, the composite WS, and the atomic WSs are shown in Figure. 2. When the application invokes operations in composite WS[1], for instance let it be O1, then O1 can be executed either using O1 in atomic WS W1 or W3. The set of atomic WSs to be delegated is called the WS community.

Web service consumers require tools to search for suitable services. This poses challenges not only in discovery mechanisms and guaranteeing high quality services. The web services composition performs the service selection, that can be done based on quality factors and transactional properties. Due to the dynamic nature of Web services, several system issues must be considered when integrating Web services. This paper proposes a broker-based architecture for the dynamic integration of Web services with QoS constraints and transactional properties. The main motivation of the proposed framework is to delegate brokers to make intelligent service selection for business process requesters.

FIGURE 2: INVOKING RELATIONSHIPS BETWEEN ATOMIC WSS, THE COMPOSITE WS, AND THE APPLICATION



RELATED WORKS

Many researchers have studied on the adaptive and dynamic service composition problem. Many industry standard XML-based languages have been developed recently, to specify web services interactions. Previously, a framework for quality-driven web service composition [2] was proposed that selects the web services based upon the QoS requirements of the requestors. S.Majitha et al proposed a framework for reputation-based semantic service discovery [3]. Diego and Maria [4] proposed an extended Web service architecture to support QoS management. There are many web service discovery models that contain UDDI to accommodate the QoS information and a management system to build and maintain service reputations and a discovery agent to facilitate service discovery. Different approaches for various optimal web service selection problems have been proposed in the previous years. However, none of these approaches takes into account the transactional behavior of the composite WS.

A set of activities having a transactional behavior are co-ordinate by workflow systems, by transactional protocols, and by Advanced Transactional Models (ATMs).The workflow systems can be managed by using exception handlers[5] and also by ensuring atomicity of each activity within the workflow. Bhiri et al. [6] extend workflows patterns in order to consider the transactional behavior in case of failures and recovery. BTP (Business Transaction protocol), WS-TXM etc are transaction protocols. BTP uses XML messages for managing the workflow between business partners. ATMs (Advanced Transaction Models) can be nested transactions, split and join transactions, flexible transactions etc. Several approaches use ATM to implement transactional behavior for WSs. This paper considers flexible transactions for web service composition.

When considering end-to-end constraints, selection algorithms are of two types: Combinatorial approach and graph approach [7]. The combinatorial approach models the selection problem as a Multiple Choice Knapsack Problem (MCKP) and gives an efficient algorithm to solve it. However, this approach can only treat one execution path at a time. So for an execution plan with more than one execution path, we need to run the algorithm more than once, and compare the results to find the optimal solution. The graph approach models the problem as a Directed Acyclic Graph (DAG) and converts the original problem to the problem of finding the highest utility path in DAG with an end-to-end constraint. It can handle the whole execution plan at the same time. Both the approaches consider end-to-end QoS constraints.

Genetic Algorithms [8] are also used for web service selection. A hybrid genetic algorithm uses a local optimizer to improve the individuals in the population and utilizes a crossover operator and mutation. The individuals in the initial population are randomly generated at first, and at the end of each generation the local optimizer is used to improve the individuals in the population. The local optimizer maximizes the overall QoS value and also minimizes the number of constraint violations of an individual.

A new TQoS selection algorithm is proposed [9]. It consists of a Web service selection approach supporting transactional and quality-driven WS composition. The new broker based architecture focuses on the selection of WSs based on their QoS and transactional behavior using the TQoS selection algorithm.

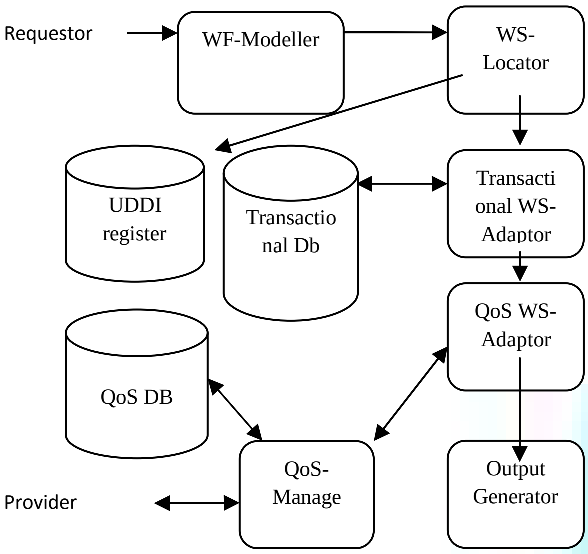
THE TQoS BROKER ARCHITECTURE

The new TQoS broker based architecture performs with an objective of selecting the best web service that satisfies transactional properties as well as the requester's QoS constraints and preferences. The Broker [10] contains six main components such as WF-Modeller, WS-Locator, Transactional WS-Adaptor, QoS WS-Adaptor, Output-Generator, and QoS-Manager as shown in Figure. 3.

A. WF-Modeller

The First step is done by the WF-Modeller. The requestor inputs a service description to the Broker referring to the required final Web service [11]. The WF-Modeller returns a workflow as a result. It returns a set of activities where each activity is complemented by a set of semantic annotations, to describe its functionalities and capabilities.

FIGURE 3: QoS BROKER ARCHITECTURE



B. WS-Locator

Based upon the workflow that has been generated, the WS-Locator will identify one or more Web services for each workflow activity. It searches the UDDI registry to find similar web services for each workflow activity. It exploits both UDDI search functionality, and semantics annotations to perform the assignment.

C. Transactional WS-Adapter

Based upon the required transactional property of the activity, the appropriate web services are selected from the located ones. The required transactional property depends upon that of the immediate previous activity.

D. QoS WS-Adaptor

The requestor specifies QoS requirements along with the request. From the set of services for each workflow activity obtained from the Transactional WS-Adaptor it will select the best web services satisfying requestor's end-to-end QoS constraints and preference for every task to take part in the composition.

E. QoS Manager

The QoS property values provided by service providers are finally verified by the QoS-Broker

F. WSBPEL generator.

This is the last step. The results returned by the QoS Adaptor are translated into a WSBPEL document.

Step 1: User submits the request to the Broker.

Step2: The WF-Modeller converts the request to a flow model.ie; it generates a workflow based upon the user's request. The workflow is given to the WS-Locator.

Step3: The WS-Locator discovers the available web services for each activity in the workflow. It searches the services in the UDDI registry.

Step4: The Transactional Adaptor selects the appropriate services that satisfying the transactional property from the available services for each activity given by the WS-Locator.

Step5: The requester's demand may include not only functional aspect of web service but also non-functional aspects like Quality of service (QoS). The services those satisfy transactional property for each activity is given as input to the QoS WS-Adaptor. From that it will select the service that satisfies user's end-to-end QoS constraints.

Step6: If a service problem occurs during process execution, it is reported to QoS-Manager and affects on some QoS property values. The QoS property values obtained from the service providers are verified and certified by the verifier and certifier component before registering them into the QoS Database.

Step7: Finally Output generator generates a document based upon the composition and give back to the customer.

TRANSACTIONAL COMPOSITION

A. Atomic Transactional Web Service

An atomic transactional Web Service (*AT*) is a component service of a composite web service. These component services are Web services operation (*op*), i.e. AT= {*op*1, *op*2, ... *opi, ... opn*}, where *opi* denotes service operation which is the component service of ATS, *opi*∈*ATS*, *i*=1, 2,..., *n*.

The transactional properties of an atomic web service may reveal the 3 features [12]:

Definition 1(Pivot WS): A pivot WS is one which is neither compensatable nor retriable. On one hand, there is no guarantee that this type of web service can be executed successfully. On the other hand, a committed pivot web service cannot be rolled back.

TABLE 1: TRANSACTION PROPERTY OF A SEQUENTIAL AND A CONCURRENT EXECUTION OF TWO ELEMENTARY WSS

ws1	ws2	ws1seqws2	ws1splitws2
p	p	a	a
p	c	a	a
p	pr	a	a
p	cr	a	a
pr	p	a	a
pr	c	a	a
pr	cr	ar	ar
pr	pr	ar	ar
c	p	a	a
c	c	c	c
c	pr	a	a
c	cr	c	c
cr	p	a	a
cr	c	c	c
cr	pr	ar	ar
cr	cr	cr	cr

Definition 2(Compensatable WS): A WS is compensatable if it is able to offer compensation policies to semantically undo the original activity.
Definition 3(Retriable WS): A web service is retriable if it is able to offer forward recovery. In other words, activities with this property can guarantee a successfully termination after a finite number of invocations.
For example, consider a simple service for buying flight tickets. Consider that such service has three operations: Reservation, Cancel reservation and Purchase. The Reservation is for making a flight reservation, Cancel reservation for canceling a reservation, and Purchase, for buying a ticket. The reservation operation that reserves a seat in an airline reservation system is compensatable since there is an operation for canceling reservations. The cancel reservation operation is retriable since it eventually succeeds after a finite set of retries. On the other hand, suppose if it is a non refundable ticket purchases then the purchase operation is pivot because it cannot be undone.

TABLE 2: TRANSACTION PROPERTY OF A SEQUENTIAL AND A CONCURRENT EXECUTION OF A CWS WITH AN ELEMENTARY WS

cws	ws	cwsseqws	cwssplitws
a	p	a	a
a	c	a	a
a	pr	a	a
a	cr	a	a
ar	p	a	a
ar	c	a	a
ar	pr	ar	ar
ar	cr	ar	ar
c	p	a	a
c	c	c	c
c	pr	a	a
c	cr	c	c
cr	p	a	a
cr	c	c	c
cr	pr	ar	ar
cr	cr	cr	cr

B. Composite Transactional Service

A composite transactional Service (*CT*) is a composite Web service, i.e. *CT*= {*tws*1, *tws*2, ..., *twsi*, ..., *twsn*}, where tws*i* denotes atomic transactional Web service, *twsi* ∈*CT*, *i*=1, 2, ..., *n*.

The transactional properties of a CWS highly depend on the transactional properties of its component WSs and on the structure of the workflow. We have the following definitions:
Definition 4(Atomic CWS): A CWS is atomic if once its entire component WSs complete successfully. Their effect remains forever and cannot be semantically undone. On the other hand, if one component WS does not complete successfully, then all successfully executed component WSs have to be compensated.
Definition 5(Compensatable CWS): A CWS is compensatable if its entire component WSs is compensatable.
Definition 6(Retriable CWS): An atomic or a compensatable CWS is retriable if all its components are retriable.
Definition 7(Transactional CWS): A Transactional Composite Web Service (TCWS) is a CWS whose transactional behavioral property is in { $\vec{a}, \vec{ar}, c, cr$ }

QOS COMPOSITION

QoS issues are considered as an important factor in web service selection. QoS can be used to differentiate web services. The new broker architecture uses a global QoS optimization selection algorithm whereas the existing TQoS algorithm considers only local optimization. The global QoS optimization selection algorithm is as follows: a set of transactional WSs is selected for each activity based upon their transactional properties. From those set of services, an end-to end QoS driven service selection is done.
For the selection of a WS for each activity, the system uses the classical Multiple Criteria Decision Making (MCDM) approach [13]. This selection is based on the weights assigned to each quality factors. The local selection is not suitable for QoS based web service composition when consider the constraint like maximum total price since such global constraints cannot be verified locally. The global selection solves the problem by considering all possible service combinations. The aggregated QoS value of each service combination is computed and the one with maximum aggregated utility value is selected.

The Quality assessment of web service is used for obtaining high-quality results [14]. Web service QoS requirements affects the performance of web services. Often, unresolved QoS issues [15][16] cause critical transactional applications to suffer from unacceptable levels of performance degradation.
The Quality of service (QoS) is a combination of several qualities or properties of a service, specified in Table 3.

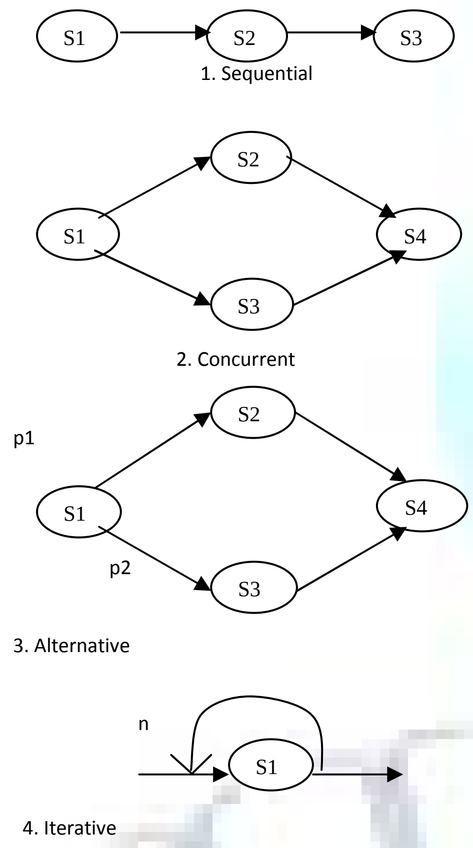
TABLE 3: OUTLINES SOME POSSIBLE QWS METRICS THAT CONSIDER WHEN DISCOVERING RELEVANT SERVICES

	Parameter	Description
1	Response Time	Time taken to send a request and receive a response
2	Availability	Number of successful invocations/total invocations
3	Throughput	Total number of invocations for a given period of time
4	Likelihood of success	Number of response/number of request messages
5	Reliability	Ratio of the number of error messages to total messages
6	Compliance	The extent to which a Web service follows the Web Services Interoperability (WS-I)Basic Profile
7	Latency	Time the server takes to process a given request

COMPOSITION FLOW MODELS

1) *Sequential Composite Services*: The activities within a sequential composite service [17] are executed with a sequential dependency between them. As a result, the transactional property of an activity will be affected by the transactional property of the immediate preceding activities.
2) *Concurrent Composite Services*: In concurrent composite services, different paths are allowed to execute simultaneously. There are no dependent restrictions between these paths. The system can schedule these activities independently.
3) *Alternative Composite Services*: Only one path of alternative composite services will be executed.
4) *Iterative Composite Services*: The activities within an iterative composite service are executed with repetition. Figure.4 illustrates composition flow models.

FIGURE 4: COMPOSITION FLOW MODELS



TQOS SELECTION ALGORITHM

In the TQoS-driven selection algorithm, a workflow is given as the input and a TCWS is the output. When a WS is assigned to an activity of the workflow, its transactional property influences the selection of the WS for the next activities.
TQoS define two notions of execution risk in a transactional system:
Risk 0: The system guarantees that if the execution is successful, the obtained results can be compensated by the user. The user can choose another application that can be used to undo the previous effect.
Risk 1: The system does not guarantee that the result can be semantically undone by the user in case of successful execution.
TQoS algorithm considers both the sequential pattern and parallel pattern while assigning web services to the activities in the work flow.

A. Sequential Pattern Assignation

Proposition 1: In a sequential pattern, if the web service assigned to the first activity of the pattern is pivot (p), pivot retriable (pr), atomic $(\tilde{a})$, or atomic retriable $(\widetilde{ar})$, then the WS assigned to the second activity should be pivot retriable (pr), atomic retriable $(\widetilde{ar})$, or compensatable Retriable (cr) in order to obtain a TCWS. If all components of a CWS are Retriable, then the Transactional Property (TP) of the resulting TCWS is atomic $(\tilde{a})$ and is moreover atomic retriable $(\widetilde{ar})$.

Proposition 2: In a sequential pattern, if the WS assigned to the first activity of the pattern is compensatable (c) or Compensatable retriable (cr), then WS of any transaction property can be assigned to the second activity to get the resulting CWS as always transactional (TCWS).

If the WS assigned to the second activity is either pivot (p), pivot retriable (pr), atomic $(\tilde{a})$, or atomic Retriable $(\widetilde{ar})$, then the TP of the resulting TCWS is atomic $(\tilde{a})$. The TP of TCWS is compensatable (c) if the WS assigned to the second activity is either compensatable (c) or compensatable retriable (cr). When both component WSs are retriable, the resulting TCWS is retriable. Most of the existing broker-based architectures consider only sequential pattern.

B. Parallel Pattern Assignation

Proposition 3: If a pivot (p) or an atomic $(\tilde{a})$ WS is assigned to one activity of a parallel pattern, to obtain a TCWS, and then the WS assigned to the other activity should be compensatable retriable (cr). The transactional property of the resulting TCWS is atomic $(\tilde{a})$.

Proposition 4: If a pivot retriable (pr) or an atomic retriable $(\widetilde{ar})$ WS is assigned to one activity of a parallel pattern, to obtain a TCWS, the WS assigned to the other activity should be pivot retriable (pr), atomic retriable $(\widetilde{ar})$, or compensatable retriable (cr). The transactional property of the resulting TCWS is atomic retriable $(\widetilde{ar})$.

Proposition 5: If a compensatable (c) WS is assigned to one activity of a parallel pattern, to obtain a TCWS, the WS assigned to the other activity should be compensatable (c) or compensatable Retriable (cr). The transactional property of the resulting TCWS is compensatable (c).

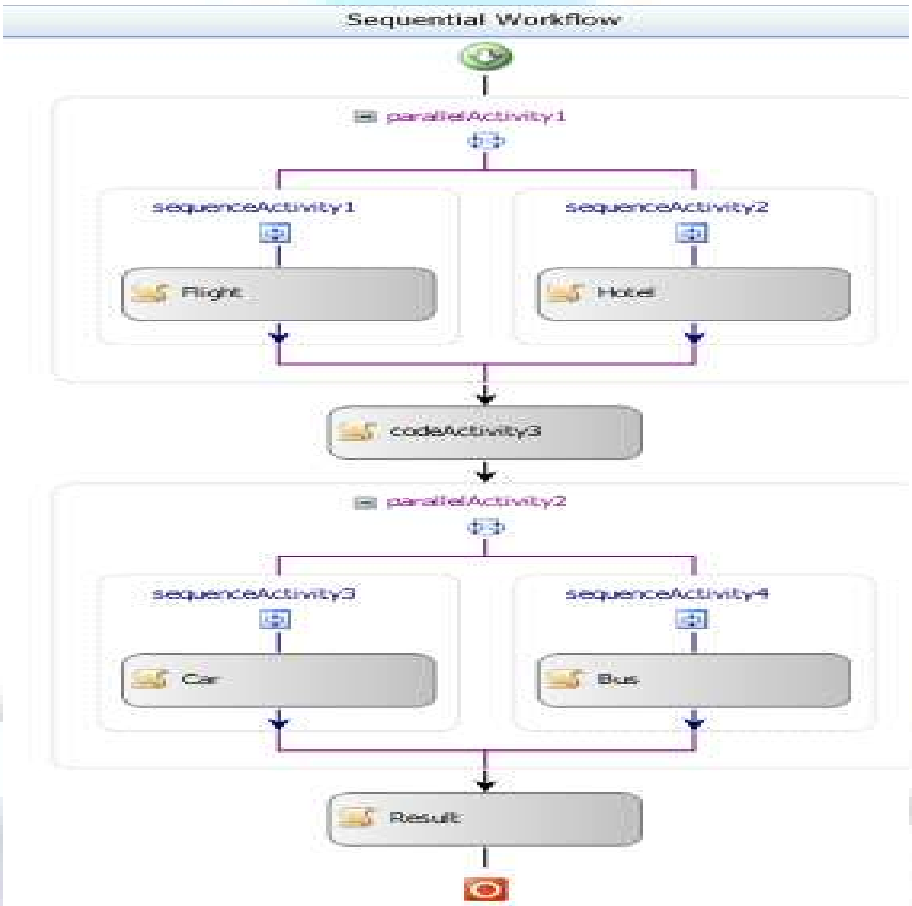
Proposition 6: If a WS is assigned to one activity of a parallel pattern is compensatable retriable (cr) then the resulting CWS is independent of the WS transactional property assigned to the other activity. If the WS assigned to the other activity is pivot/atomic $(p/\tilde{a})$, compensatable (c), pivot/ atomic retriable $(pr/\widetilde{ar})$, or compensatable retriable (cr), then the transactional property of the resulting TCWS is atomic $(\tilde{a})$, compensatable (c), atomic retriable $(\widetilde{ar})$, or Compensatable retriable (cr) respectively.

Previously coordination of a set of activities having a transactional behavior has been tackled by workflow systems, by transactional protocols, and by Advanced Transactional Models (ATMs).Transaction protocols define a model for coordinating the transaction execution of Web services based on a predefined set of transaction messages.

IMPLEMENTATION AND RESULTS

The web service composition with transactional and quality driven properties is implemented in Visual Studio 2008.It uses the workflow template. The programming is done in C# and platform used is Windows XP. It uses the SQL database 2000 as the backend server.

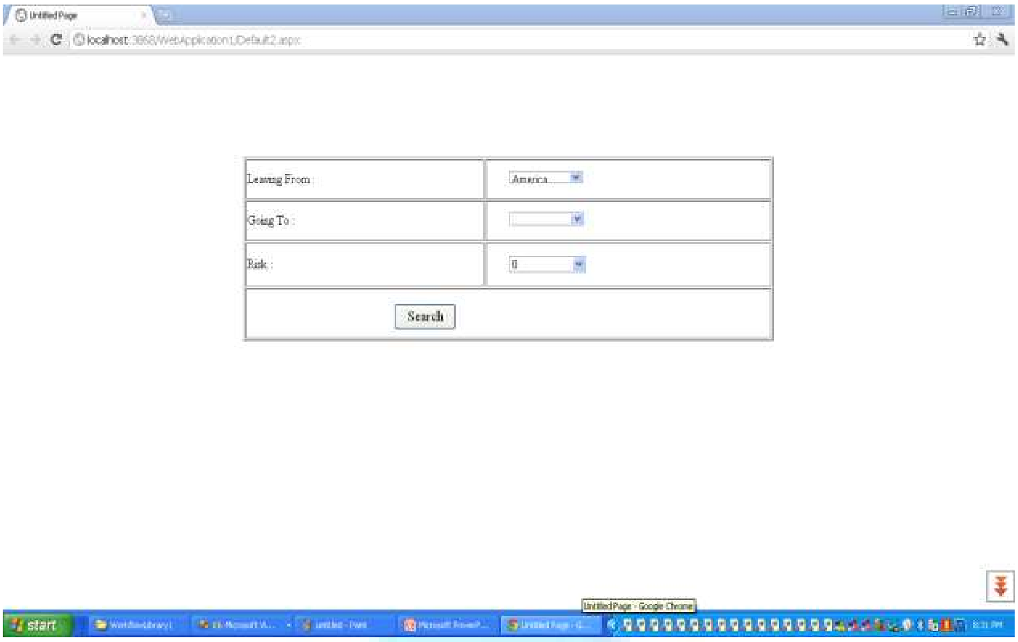
FIGURE 6: THE WORKFLOW USED



A Broker with 15 web services is implemented. The workflow generated as shown in figure 6. It considers both the sequential and parallel execution. In sequential the transactional property of an activity depends on its immediate preceding activity. Therefore it selects a web service refering to the transactional properties such that it provides efficient commit transactions. For parallel implementation both the actions are executed at the same time. The transactional property of the output of parallel activities depends on the transactional property of both the actions.

The customer gives the requirements to the input webpage is shown in fig.7. Based on the requirements our broker selects the best web services from the 15 available web services. For this it consider the transactional property of the web services based on the workflow. The execution begins and finally it give the best results as shown in fig.8 and customer can book them based on the number of travelers and the details are stored to the database. The program executes and gives the best results. The results are tested for various inputs and it has been proved that it provides better efficient results considering both the quality and transactional properties for both the sequential and parallel workflow.

FIGURE 7: THE INPUT WEB PAGE USED



For instance, based on the input requirement the broker selects best web services for flight, hotel, car or bus services. The web services associated with the activities are shown in Table4. The flight and hotel web service selections are performed using AND split and the car or bus service selection is performed using XOR split. The AND and XOR split are executed sequentially. Thus sequential, parallel and alternative executions are performed together in a workflow. The results are verified for random inputs and it provides a better service selection as shown in Table 5.

FIGURE 8: THE OUTPUT FORM THAT GIVES THE BEST SERVICES

TABLE 4: WEB SERVICE ASSOCIATED WITH THE ACTIVITIES

Activity	Candidate web services
Flight	F1,F2,F3
Hotel	H1,H2,H3,H4
Bus	B1,B2,B3,B4
Car	C1,C2,C3,C4

TABLE 5: RESULT OF THE BROKER

From	To	Flight	Hotel	Bus	Car
A	B	F2	H3	B4	X
S	A	F3	H4	X	C3
A	B	F3	H1	B1	X
D	B	F1	H3	X	C2

CONCLUSIONS

With the emerging role of web services in business processes, the requirement of composing and executing them has begun to draw high attention, and today the need to find the optimal web services composition for the business processes is a challenging issue. This paper proposes a new broker based architecture that performs an optimal web service composition according to the transactional properties and user QoS requirements which is an added advantage than existing brokers. This will dynamically generate a workflow based on user's request and make a better web service selection for the composition. As the new broker based architecture makes use of the modified TQoS algorithm, it considers both sequential and parallel patterns and provides global optimization.

REFERENCES

[1] San-Yih Hwang, Chien-Hsiang Lee, and Cheng-Hung Chen, "Dynamic Web Service Selection for Reliable Web Service Composition", IEEE Transactions on Services Computing, vol. 1, no. 2, April-June 2008.

[2] T. Yu and K. Lin, "A Broker-Based Framework for QoS-Aware Web Service Composition", Proc. of the Intl. Conf. on e-Technology, e-Commerce and e-Service, 2005.

[3] S. Majitha, A. Shaikhali, O Rana, and D. Walker, "Reputation based semantic service Discovery," *In Proc. Of the 13 th IEEE Intl Workshops on Enabling Technologies Infrastructures for collaborative Enterprises (WETICE)*, pp.297-302, Modena, Italy, 2004.

[4] D. Garcia and M. Toledo, "A web service Architecture providing QoS Management", *Web Congress, LA- Web '06. Fourth Latin American*, pp.189-198, 2006.

[5] Hagen and G. Alonso, "Exception Handling in Workflow Management Systems," *IEEE Trans. Software Eng*., vol. 26, no. 10, pp. 943-958, Oct. 2000.

[6] S. Bhiri, O. Perrin, and C. Godart, "Extending Workflow Patterns with Transactional Dependencies to Define Reliable Composite Web Services," *Proc. Advanced Int'l Conf. Telecomm. and Int'l Conf. Internet and Web Applications and Services (AICTICIW)*, p. 145, 2006.

[7] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Trans. on Web*, vol. 1,no. 1, p. 6, 2007.

[8] Maolin Tang and Lifeng Ai, "A Hybrid Genetic Algorithm for the Optimal Constrained Web Service Selection Problem in Web Service Composition", *Evolutionary Computation (CEC), 2010 IEEE Congress*, July 2010.

[9] Joyce El Haddad, Maude Manouvrier, and Marta Rukoz," TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition", *IEEE Transactions On Services Computing*, Vol. 3, No. 1, January-March 2010.

[10] S.M. Babamir, *S.* karimi and M.R. Shishehchi, "A Broker-Based Architecture for Quality-Driven Web Services Composition," IEEE *computational Intelligence and software engineering (CISE), International conference*, December 2010.

[11] B. Carminati, E. Ferrari, P.C. K. Hung, "Web Service Composition: A Security Perspective", *International Workshop on Challenges in Web Information Retrieval and Integration*, Tokyo, Japan April 08-April 09.

[12] S. Mehrotra, R. Rastogi, H. Korth, and A. Silberschatz, "A Transaction Model for Multidatabase Systems" , *Proc. Int'l Conf. Distributed Computing Systems,* pp. 56-63, June 1992.

[13] M. Zeleny, *Multiple Criteria Decision Making.* McGraw- Hill, 1982.

[14] Eyhab Al-Masri and Qusay H. Mahmoud,"Toward Quality Driven Web Service Discovery," *IEEE Computer Society,* IT Pro May/June 2008.

[15] Daniel A. Menasce, "QoS Issues in Web services," *IEEE Internet Computing,* December 2002.

[16] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, May 2004.

[17] L. Li, C. Liu, and J. Wang, "Deriving Transactional Properties of Composite Web Services*," Proc. IEEE Int'l Conf. Web Services (ICWS '07)*, pp. 631-638, July 2007.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application and Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mails i.e. **infoijrcm@gmail.com** or **info@ijrcm.org.in** for further improvements in the interest of research.

If you have any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-

Co-ordinator