Workbook Approach to Algorithm Design and Service Accounting in a Component Orientated Environment

Jonathan Brazil
jbrazil@tssg.org

Eamonn de Leastar
edeleastar@tssg.org

Conor Ryan
cryan@tssg.org

Mícheál Ó Foghlú
mofoghlu@tssg.org

Telecommunications Software & Systems Group (TSSG)
Waterford Institute of Technology,
Cork Road, Waterford, IRELAND
http://www.tssg.org

Project Funded by Enterprise Ireland

***Abstract* - As the shift towards the next generation of services runs its course, the emphasis within accounting for such services has moved towards robust and flexible components. An accounting architecture with the ability to scale, based on the foundation of a solid core of rating components, incorporating transaction management, object pooling and queuing is presented. Using the workbook approach to algorithm modelling is a logical option that allows for the decoupling of the algorithm from the architecture of the rating component. Decoupling the algorithm, allows for the design of core functionality that will rate for any service based on the concept of "pluggable" algorithms for each respective service.**

I. Introduction

Rating[1] is a time-consuming operation; requiring significant resources to cope with the volume of transactions involved in a typical (inter)national service. The next generation of services has just started to emerge into the market place and a suitable rating mechanism is needed in order to meet the demands of multiple services, which are becoming increasingly diverse.

Traditional solutions involved the use of relational database tables [11] to support the large volume of records generated by a typical telephone exchange. These tables were exhaustively queried and cross-referenced in order to obtain results relating to subscriber charges. Once calculated this information is sent to the billing department for processing. The residue left behind is a huge collection of database rows representing the original Charge Detail Records (CDR). This information is not self-descriptive and provides very little insight into the usage of the service in its raw form. The Internet Protocol Detail Record (IPDR) [2] offers a more descriptive representation mechanism, enabling the integration of varied service delivery models.

A suitable rating engine that combines efficient high volume processing with the flexibility to cope with multiple types of service is paramount to the continual evolution of customer charging. For such a rating engine to succeed it must comprise a number of features. Firstly, it should have a solid, refined rating core that is primarily concerned with the evaluation of algorithms against usage data to produce charges. Secondly, it must be scalable in order to cope with the high volumes of traffic. Thirdly, it must present a way of decoupling the charging algorithms from the system so that as services or charging structures change over time, the rating engine will not require recoding to reflect the new ideas.

This paper describes the Rating Bureau Service (RBS) project currently at Waterford Institute of Technology. This project is applying this abstraction approach to the design (and prototype development) of a flexible, distributed, component orientated rating architecture. RBS extends some of the ideas implemented in the Bandwidth 2000 project [1].

At a conceptual level, the RBS is built using the concept of broker domain management via a web-based user interface. When a broker, who can be considered to be an agent for a service provider, registers with the system an account is set up where they can modify account details and their algorithm portfolio. Charge records are uploaded, in IPDR [2] format, by the broker to the system via the UI. The broker can trigger a rating of these records, can modify the algorithm at any stage, and rate again using the updated algorithm without any need for administrative interaction. Also the updated IPDRs that remain after the rating process has finished contain meaningful data which, is in standard XML mark-up [10] for added clarity.

For practical and pragmatic reasons the project has adopted the use of standard spreadsheet-based workbooks as a serialization medium for algorithms and lookup tables. This maximises the system's potential to allow the general operator to maintain the system, as

[1] Rating is the process of applying an algorithm to a set of usage data to produce a charge for a particular service

manipulation of Excel workbooks [4] is a common skill which allows simple tweaking of relatively complex charging algorithms. In order to implement this, the Rating Engine is constructed using a standard COM-based [8][9] spreadsheet evaluation engine component (essentially a computational component that adheres to the Microsoft Excel workbook format [4][5]). Worksheets provide a natural design canvas for algorithms. The spreadsheet model is a tried and tested model in the field of conventional accountancy and now it is being applied to rating for e-services.

II. Broker Interaction Model

It will always be necessary to have some user interaction in order for a rating engine to operate successfully. The algorithms must be maintained, and usage data pushed into the system for processing. However, new thinking on the provision of services and the role of the service provider has opened up a new avenue in the design of a rating engine.

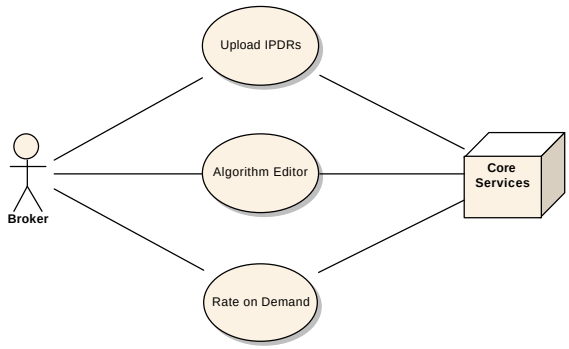

Figure 1: Broker Interaction

A rating engine might typically operate within an organization, processing usage data from some record source into a specific charge for the associated service. This is no longer a hard and fast rule. Service providers are outsourcing customer management to brokers with increasing frequency. For a service to succeed it must offer value for money and Quality of Service to meet the needs of the customer [3]. The concept of a service broker provides the means to offer customers value for money. A composed service can be distributed across multiple service providers to provide the cheapest solution. In the case of atomic services the broker can perform the labour intensive work of shopping around for the best deal therefore saving the customer time and money.

This new model of service provision to customers dictates that a new control path must be added to the existing service architecture. A Rating Engine can become a service used by a broker to deliver a flexible charging regime to its service providers, and customers. The broker must be able to manage their respective aspect of a generic rating engine service. Brokers need to maintain personal profiles and portfolios (collections of algorithms for different services). The broker needs to be able to view the impact of changes within the algorithm on their profits or potentially, their customer base. Allowing brokers to remotely manage and update relevant information delegates the entire user profile administration end of the system to the broker.

Thus candidate components within the RBS can be identified: User Interface, Broker Manager, Portfolio Manager and Rating component emerge as clear subsystems from the above discussion. However, this architecture is still missing one component in order to be complete. For rating to take place information must be passed into the system in a standardised format. In the RBS system the medium for the transfer of data records is the Internet Protocol Detail Record (IPDR) [2], an XML-based file with a well-defined XML Schema.

Within the RBS the IPDRServices component is responsible for accepting IPDRs from a requesting source and passing these IPDRs into the system so that they can be rated. There is standardised approach to IPDR management, covered in more detail in other publications beyond the scope of this paper.

```
<?xml version="1.0"?>
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
      xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" docid="01234"
      seqNum="1" version="2.5" startTime="2002-05-05T18:50:00Z">

      <IPDRRec info="ipdr-rec.wit.ie" startTime="2002-05-05T18:50:00Z" id="1"/>
      <IPDR seqNum="0" time="2002-05-05T18:50:00Z">
            <SS service="FormulaOne_FormulaOne" id="OCS_01">
                  <SC xsi:type="SC-FixedLine-Type">
                        <userID>JBrazil</userID>
                        <contextID>RBS1</contextID>
                        <subscriberID>WIT0002</subscriberID>
                        <userApplicationHost>193.1.184.11</userApplicationHost>
                  </SC>
                  <SE xsi:type="SE-FixedLine-Type">
                        <serviceProviderID>WIT0003</serviceProviderID>
                        <serviceTariffID>CALL_01</serviceTariffID>
                        <serviceProviderHost>193.1.184.13</serviceProviderHost>
                  </SE>
            </SS>
            <UE xsi:type="UE-FixedLine-Type">
                  <ANumber>050679390</ANumber>
                  <BNumber>1891110110</BNumber>
                  <date>05/05/02</date>
                  <time>18:30:45</time>
                  <duration>100</duration>
            </UE>
            <CE>
                  <CustomerCharge xlref="charge"/>
            </CE>
      </IPDR>
</IPDRDoc>
```

Figure 2: IPDR Document

III. Workbook Representation

The idea of using a spreadsheet workbook to manipulate or process accounting type data is universally accepted. The practice is considered the norm in conventional accounting. The workings of a spreadsheet are quite simple. Data is inserted into either index addressable or named cells. This information can then be modified or injected into some calculation by the use of

standard formulae. Every workbook has this typical layout and structure. Microsoft Excel [4] is the de-facto standard for spreadsheet modelling in the workplace, adopted in the financial world as a standard document format.

Conventional rating engines extract usage information from a Charge Detail Record (CDR) and insert this information into programmatic variables for calculation in a hard-coded formula. Representing the same information in a workbook makes a considerable difference to this unyielding design. Cells in a workbook can be named so as to create an intuitive alias rather than a cell reference. The IPDR is a standard XML document with named sections and data elements. Each element in this IPDR document can be mapped to a named cell in a workbook. This simple mechanism can serve as the foundation for a sophisticated rating process.

The composition of a workbook is a hierarchical tree structure and can easily be reverse engineered. Figure 3 contains a graphical view of a workbook schema for Excel XP and OpenOffice Spreadsheet. A workbook comprises zero or more worksheets each containing a table of rows. Within each row are a number of cells, each of which may contain data and an optional name reference for the cell. This name reference is the foundation to the simplicity of the data mapping. If a new service is provisioned for use, a charging algorithm is needed to rate for that service. Assuming that IPDR is the standard transmission medium for the usage data, it is possible to design a suitable workbook in Excel and name specific cells to match the names of the necessary data elements in the IPDR. Thus when an IPDR is received each element in the IPDR is placed into the workbook using the name of the element as the cell reference. A pre-loaded algorithm within the workbook operates on the newly populated cells and produces a result in a specified location. This result is then extracted from the workbook and inserted into a charge element field within the IPDR, which is then persisted as a rated IPDR.

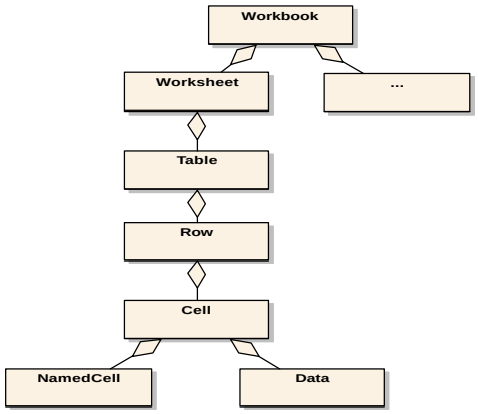

Figure 3: Workbook Schema

Workbook formulae are a simple correlation of basic mathematical functions, used to operate on the data contained within cells in a worksheet. These formulae can be constructed by anyone with a reasonable working knowledge of spreadsheets. The native interface of Excel lends itself to the non-taxing creation of arbitrarily complex charging algorithms. It makes sense that in a multi-service environment the brokers should be the authors of the algorithms and not the overall system administrator.

IV. Rating for Multiple Services

In traditional rating environments it is normal for the variety of services being offered to fall under the same fundamental service umbrella. Such an example would be a standard telecommunications service provider, offering telephone and Internet services. Both of these services use a phone call. The only difference is one carries voice whereas the other carries data. It is most likely that each call is rated the same way initially and then a post rate process decides what special rates, if any, to apply in the event of the number called resolving to an Internet call or a voice call.

The next generation of services is far more diverse than this model. The connection to the Internet will only be the first step on the ladder of a composite service. Once connected, users will be able to Video Conference, make VoIP calls, and use a host of other online services. However, with provisioning of services over GPRS which is *always on*[2] the initial connection to the Internet no longer becomes applicable and this must be rated in accordance with the customer's usage of the GPRS service. Either way each of these services must be accounted for in a specific manner. For example, it would not be possible to rate a file upload the same way that one would rate a VoIP call. The type of transactions involved and guarantee of QoS level is completely different for each service. From this arises the need for each broker of a service to have a dedicated algorithm for each service that they are offering. A portfolio of these services is maintained and consulted at the time of rating to match an algorithm with the type of service being accounted for. Information relating to the broker and type of service is contained in the IPDR and is easily extracted and mapped to a suitable algorithm stored within the respective broker's portfolio.

This allocation of algorithms at rate-time allows for the loosely coupled system captured within RBS (Fig 4). Should a broker wish to change the way the VoIP service

[2] *always on* refers to the concept of a mobile device never having to connect to the network. Once powered on the device is automatically connected and the duration of the connection does not incur any cost, just like listening to a terrestrial radio service.

is rated then it can be done rapidly with minimum effort to reflect these changes in the system. The broker logs into the system, modifies the desired workbook and saves the changes. Upon saving this new workbook the broker can then choose to rate new IPDRs or re-rate existing IPDRs to see the effect of the modifications. Compare this to the traditional environment where an algorithm would have to be manually changed by the development team and then the entire system would have to be brought down and reinitialised with the new code. This approach enables the system to remain live while changes are being made to the way usage data is rated. The system adheres to a term to be identified as Test Once Rate Again (TORA), where the broker is the highest authority regarding the maintenance of the algorithm.

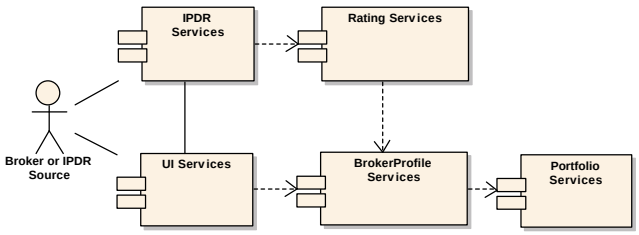

Figure 4: RBS Architecture

V. Workbook Based Algorithm Design

The creation of an algorithm in a workbook is essentially composing formulae within the context of an IPDR aware spreadsheet. This formula takes data from other areas of the workbook and uses it to calculate a result, in this case a charge for the service being rated. The native interface of Excel or OpenOffice Spreadsheet provides a familiar working environment. Brokers can design algorithms offline and then when tested, can be uploaded to the system. The ability to address individual cells with named aliases is paramount to the success of this concept. By removing the need to design the workbook to a rigid standard the algorithm designer needs only to insert the named fields that correspond to the fields of the IPDR being rated. The named cells allow the designer to position cells anywhere that is desired for easy interpretation of the workbook and it will not make any difference to the working of the system. They can also use colours and other formatting methods as long as the basic underlying structure is present.

Clearly workbook-based algorithm design can deliver the flexibility and ease of use required to compose algorithms for next generation services. However, in the context of the RBS, if workbook-based algorithms are to be used then an engine to interpret these workbooks must become the core of the rating component. Tests have proven that Excel, operated programmatically using COM Automation technology, is capable of rating small amounts of IPDRs in a reasonably efficient manner. It has not yet been determined if Excel could handle the volume of IPDRs being generated by a live system rating for multiple services. Commercially available components exist that are tuned towards higher throughput processing. The RBS has chosen to engage the Formula One [6] ActiveX control from Tidestone Technologies, Inc. This control is a spreadsheet engine that understands the file format of Microsoft Excel workbooks. The Formula One package also has its own workbook editor but Microsoft Excel is a more fully featured interface, and thus continues to be used for design of algorithms.

An algorithm designer is under construction for inclusion in the RBS suite. A broker wishing to create a new algorithm will open this application. Using schemas defining the structure of the IPDRs to be rated, each of the elements in the IPDR become available to the user as named cells in a workbook that can be used in custom formulae. This type of interface ensures a valid algorithm will be composed, by leveraging the precise formatting and structural rules embodied in the Schema. It must be stressed however that sophisticated low level detail with respect to algorithm construction may never be achievable with wizards, as the complexity involved would be beyond the scope of a generic application. Instead it is viewed that the algorithm wizard in this context will provide outline algorithm templates using the usage element fields from the IDPR schema to create named cells in the workbook. These cells can then be used in the creation of sophisticated formulae combined with large tables of rates or policies.

VI. Conclusion

Workbook-based algorithms are a logical choice when it comes to rating systems. The decoupling of the algorithm format and composition from the core rating system has obvious and positive effects. Utilising the IPDR as a transmission and storage medium for records removes the old style CSV files and database tables. Instead research is being conducted into XML storage spaces and simple query based retrieval services, such as XML-Tuples and XML-Spaces [7]. Native XML databases are also being researched, such as Ipedo's XML Database [12]. A basic implementation of this technology has been trialled for IPDR storage and retrieval in the context of the RBS project.

Component-based architecture enables the incorporation of scalability features such as object pooling, object activation strategies, and security mechanisms into the application. The COM+ environment suits the needs of this type of system. An event driven core of pooled components allows for multiple rate engines to run concurrently. Although this

will not increase overall throughput it will dramatically improve response times to brokers rating small amounts of IPDRs. There is no longer a need to wait for a large batch job to terminate. Components can be distributed across multiple machines for load balancing and security reasons. For example, a high-spec dedicated machine could be assigned solely to the task of rating and nothing else. Whereas a lesser spec machine could be assigned the task of dealing with UI requests and IPDR traffic. In addition this system gateway machine could be placed outside a firewall for external access, while the other components remained inside the firewall for security reasons.

The big question that remains to be answered is whether or not the large rating organisations such as telecommunications providers will adopt a radically new architecture in favour of the old systems. Most operators have an existing rating framework that has been tested and proven for reliability over many years. Reluctance to change from a working and reliable system will be a huge factor in gaining acceptance from this type of operator. However, in favour of systems like the RBS is that the flexibility which, is much sought after within rating systems is offered up front to subscribers of the system. Full reporting features can very easily be incorporated into the system, allowing brokers to analyse service usage under a customisable criteria base. Communications with such operators have confirmed this thinking. Existing systems offer reliability but are restricted in the services that they can rate for and are usually customised for a particular type of service. Reporting features are rather rigid and the ability to take a subset of records and rate them in different ways with different algorithms is currently unavailable for most systems.

The component architecture described in this paper is entirely demand driven and can meet these needs. The architecture lends itself to flexible and pluggable charging schemes that do not involve an overhaul of the system as new services are introduced. The architecture described in this paper outlines the component orientated framework needed for successful, flexible rating in the future.

VII. Acknowledgements

The RBS project is conducted through funding from Enterprise Ireland (EI) [14]. Acknowledgements must be made towards the groundwork performed in both the Bandwidth 2000 [1] and FORM [13] projects. Both of these projects have helped to identify key requirements for Next Generation Service accounting.

References

[1] Bandwidth 2000, “An Information Infrastructure for Bandwidth Trading”, Ten Telecom http://www.ten-telecom.org/

[2] IPDR Organization, Network Data Management – Usage (NDM-U) v3.1, IPDR.org http://www.ipdr.org, May 2002

[3] Ryan et al., “Value-Based Billing in a 3G IP Services Environment”, Systemics Cybernetics and Informatics (Florida), July 2002.

[4] Microsoft, Microsoft Excel XP, http://www.microsoft.com/

[5] OpenOffice, Spreadsheet, http://www.openoffice.org

[6] Tidestone, Formula One, http://www.tidestone.com

[7] David Moffat, “XML-Tuples & XML-Spaces”, ATN ITD, UNC-CH, Mar 1999.

[8] Sara Williams et al., “The Component Object Model: A Technical Overview”, Developer Relations Group, Microsoft Corporation, October 1994

[9] Steve Robinson and Alex Krassel, “COMponents”, Panther Software, August 1997

[10] W3, XML, http://www.w3.org/XML

[11] Amdocs, Message Processing System (MPS), http://www.amdocs.com

[12] Ipedo, IPEDO XML Database, http://www.ipedo.com

[13] FORM-IST, FORM Project, http://www.ist-form.org

[14] Enterprise Ireland, Research Innovation Funding, http://www.enterprise-ireland.com