Business Intelligence for N=1 Analytics using Hybrid Intelligent System Approach

Rajendra M Sonar

Abstract—The future of business intelligence (BI) is to integrate intelligence into operational systems that works in real-time analyzing small chunks of data based on requirements on continuous basis. This is moving away from traditional approach of doing analysis on ad-hoc basis or sporadically in passive and off-line mode analyzing huge amount data. Various AI techniques such as expert systems, case-based reasoning, neural-networks play important role in building business intelligent systems. Since BI involves various tasks and models various types of problems, hybrid intelligent techniques can be better choice. Intelligent systems accessible through web services make it easier to integrate them into existing operational systems to add intelligence in every business processes. These can be built to be invoked in modular and distributed way to work in real time. Functionality of such systems can be extended to get external inputs compatible with formats like RSS. In this paper, we describe a framework that use effective combinations of these techniques, accessible through web services and work in real-time. We have successfully developed various prototype systems and done few commercial deployments in the area of personalization and recommendation on mobile and websites.

Keywords—Business Intelligence, Customer Relationship Management, Hybrid Intelligent Systems, Personalization and Recommendation (P&R), Recommender Systems.

I. INTRODUCTION

Many organizations increasingly using BI tools such as data mining to build analytics especially domains like customer relationship management (CRM) to identify, attract, understand, serve and retain the customers [1]-[5]. The new competitive landscape requires continuous analysis of data for insight, only episodic and ad-hoc or periodic will not suffice. Traditional analytics approaches are often asynchronous with business changes. Delays in recognizing, interpreting and acting on trends are critical emerging impediments to competitiveness [6]. Traditional analytics approach based on data mining used in applications like CRM classifies large number of retail customers into few predefined groups like good v/s poor or clusters them into groups by mining huge amount of data [2]. This approach is more of top down and focuses on grouping customers rather than treating them as individuals. However, in worst-case scenario each customer can exhibit a unique pattern based on demographic profile of individual and interactions that individual does with the firm. This means it may not be sufficient to apply the broader group rules to the customer or try to fit the customer into group.

Even a firm is dealing with millions of consumers, the firm must focus on one consumer experience at a time called N=1 [6]. We use the term N=1 analytics to represent analytics that is focused on analyzing one consumer at a time and providing each consumer personalized experiences. N=1 analytics should work in real-time, analysis is done on continuous basis and is integrated into operational systems. The term consumer is used to denote a customer, user etc. [7]

Fig. 1 Possible inputs for N=1 analytics

Products and services or even various offers by the firms are built and configured for a particular kind of target group of consumers. Telecom firms offer various telephone plans keeping in mind various classes of consumers e.g. business class, students etc. or may be based on types of usage patterns e.g. a plan more suitable for more incoming calls rather than outgoing calls. They are not tailor made or dynamically configured to each one’s profile, usage pattern and preferences. In order to provide such unique personalized experiences to the consumer, the inputs must come from various sources. Figure 1 shows kind of possible inputs required to offer personalized experiences to consumers. Most of the information like user profiles, user transactions, product and services is already available in the organization in databases generated through operational systems like CRM, ERP (Enterprise Resource Planning), supply-chain management (SCM) etc. Market information is required to understand how consumers in general accept products and services, review and rate them, and what kind of products they are or would be interested in. While offering personalized experiences to individuals, the firm has to follow compliance and policy rules set by the firm, industry and regulatory bodies under which the firm operates e.g. in India, telecom firms are...
regulated by TRAI (Telecom Regulatory Authority of India) [8] while banking and financial institutions by RBI (Reserve Bank of India) [9].

On the basis of interactions and transactions, if the buying pattern of individuals or risk profiles of individuals can be determined in real-time, it can help firm a) to target the customer with unique, differential and customized product or services offerings, b) to determine which campaign, when and how customer would respond, c) to estimate risk of customer continuous basis and adjust various parameters such as credit limit accordingly, etc. Focusing on analyzing one customer at a time, needs analysis of transactions and information specific to that customer only. This reduces computation overhead and these tasks can be distributed and triggered based on some event or as and when required.

Functional areas like analytical CRM require combination of various modeling techniques: association, classification, clustering and prediction to identify, attract, retain and develop customers [2]. Hybrid intelligent systems [12]-[14] that combine and integrate various intelligent techniques at grass root level are well suited to build BI systems requiring such modeling techniques.

Web services architecture makes it easier to exchange information in plain text XML (Extensible Markup Language) [15] across diverse platforms, technologies and systems in open and standard way. The existence of both synchronous and asynchronous versions of web services enables both real-time and non-real time versions of application integration. Figure 3 shows approach to business intelligence using hybrid intelligent system approach where intelligent systems are loosely (using web services/HTTP: Hypertext Transport Protocol calls) or tightly (using APIs: Application Programming Interface) integrated into operational systems. Once intelligent systems are modeled and configured for specific problem, they are automated to learn on their own continuously based on some event or at specific time intervals based on set criteria. E.g. a) intelligent system modeled to compute associated items to a given item using collaborative filtering technique can be configured to learn new associations whenever user clicks on the item, b) intelligent system modeled to compute personalized items to be shown to the logged in active user can be configured to automatically re-compute best personalized items every fifteen minutes fulfilling some criteria like at least one download etc. Such systems can take required information from other websites using RSS: really simple syndication feeds [16] to reflect changes happening in market. Intelligent systems such as knowledge-based recommendation (advisory), can be invoked into operational system to have live interactions with user. One of the advantages of such approach, if intelligent systems are loosely coupled using web services, they can be managed well. They can be distributed running on different machines on network. This can help in better manageability and scalability. In this paper we describe such a framework for business intelligence using integrated approach as shown in figure 3 based on hybrid intelligent systems.

The rest of paper is divided into four sections, section II reviews the base hybrid development framework, in section III
we describe P&R framework to develop P&R systems on top of hybrid framework, section IV describe some case studies and section V concludes the work.

II. REVIEW OF HYBRID DEVELOPMENT FRAMEWORK

Figure 4 shows the components and architecture of hybrid framework [17]. This is an integrated development environment to develop and deploy intelligent systems backed by hybrid of ESs, CBRs and ANNs. It is being commercialized as product called iKen Studio [18] subsequently referred as the hybrid framework. The hybrid framework has all the components required to build intelligent systems backed by rule-based ES, neural network and CBR technology. It supports both structural as well as conversational CBRs [19][20]. Depending upon the broad functionalities, the framework is divided into five layers as shown in figure 4, each layer having components and interfaces specific to functionality of that layer. The components implement core functionality while interfaces provide interactive environment to develop, manage and configure the models. For example, inference engine is implemented as a component: Rule-based ES Engine, while the interface: Rule and UDF Manager provides interactive interface to retrieve, build, modify and store rule-based expert systems and UDFs (User Defined Functions). The framework allows knowledge engineers to add their own functions, called UDFs and invoke them in rule-base. A function implements well defined functionality that do not need complex set of rules but procedural logic. The code of UDF can be easily modified without change in inference engine like expert system rule-base. The details of this framework and its working can be found in [17]. Our purpose in this paper is to discuss how such frameworks can be used to build N=1 analytics that work in real-time and can be integrated into operational systems. Access layer has been added to access various services of the hybrid framework and integrate with other external systems. The services can be accessed through web services, APIs, HTTP calls or embedding interactive sessions inside web applications. The components and interfaces of access layer are connected to other four layers although it is not explicitly shown in figure 4.

The hybrid framework works in two modes: development and run-time environment. The intelligent systems can be developed and tested in development mode and can be deployed and executed through run-time environment. Figure 5 shows a screen shot of development environment. Based on functionality, the development interfaces are grouped. Once intelligent systems are developed and tested, they can be accessed and executed by using various web services, through HTTP calls or by using APIs. The web services supported by the hybrid framework are divided into groups based on functionality of web services.

Figure 6 shows the partial list of web services supported by the hybrid framework. Functionalities of all the components of hybrid framework are divided into groups based on functionality of web services. The hybrid framework works in two modes: development and run-time environment. The intelligent systems can be developed and tested in development mode and can be deployed and executed through run-time environment. Figure 5 shows a screen shot of development environment. Based on functionality, the development interfaces are grouped. Once intelligent systems are developed and tested, they can be accessed and executed by using various web services, through HTTP calls or by using APIs. The web services supported by the hybrid framework are divided into groups based on functionality of web services.
the hybrid framework can be accessed using web services. Some of the common web services like starting with DB\_ help to access data from databases while web services starting with Session\_ help to manage common pool of session data which is shared by multiple intelligent systems.

Some prototype systems have been developed to demonstrate the capabilities of the hybrid framework. Some of them are in knowledge-based recommendation systems. Most of them use combination of expert system and case-based reasoning technology. One of the prototype applications is transaction monitoring which is discussed briefly. The purpose of such applications is how such applications are built and invoked. This application monitors individual transactions. It has been modeled defining generic transaction characteristics like transaction Id, amount (cost/value), date, time, merchant name, merchant location (city/area/place), merchant type, mode of payment and items purchased. We derive other attributes from base attributes. E.g. day type: week-day, week-end, holiday derived from date and time.

The application scans and matches the current transaction with past transactions (of that user) and user’s criteria. The user can specify what kind of generic transactions he or she would like to do. Each generic type of transaction becomes a case. E.g. a user may specify preference ‘he would normally buy grocery items of amount m around location l preferably on evenings on week-ends’. This user preference is stored as a generic transaction. The user can specify multiple generic transactions as criteria. The application uses combination of rules, basic statistics and CBRs. Rules are used to do qualitative analysis as well as quantitative using various functions. These rules also control and call CBRs. Variety of functions are been added to the hybrid framework rule engine like STAT to do basic analysis of transaction of data. It returns statistics of past transactions like average, standard deviation, and maximum amount transacted. Qualitative analysis is done using set of rules. For example, one of the rules is:

IF Transaction.Merchant Type IS Grocery Store AND Transaction.Merchant Location City IS Mumbai AND Transaction.Time >= 2300 AND Transaction.Time <= 0700 THEN APPEND (ScanReport,"Purchasing at Odd Hours")

Figure 5 shows one of the rules loaded of this application in Rule Manager. Figure 7 shows screen-shots of interactive mode and results of the application. Figure 8 shows explicit execution of the application using web service: ExpertSystem\_RunWithData with goal: Transaction.Scan. The outcome of the application is shown in figure 7 has three distinct components: a) how much current transaction is matching with past ones using CBR, b) how much current transaction is matching with user set criteria using CBR, and c) quantitative and qualitative assessment using basic statistics and set of rules. Intelligent systems that monitor transactions as discussed above can be integrated into operational systems such as bank automation systems through web services. This can help banking firms to continuously monitor transactions especially like credit card transactions on continuous basis to reduce possible frauds. These intelligent systems can be triggered and executed based on certain criteria (e.g. it exceeds certain amount) before transaction gets authorized.
Figure 9 shows list of prototype applications being developed backed by intelligent techniques to have 1-to-1 services from customer’s perspective. These kinds of applications can play important role in customer attraction, retention and development. Figure 10 shows in brief how such 1-1 services can be backed by intelligent systems. Once such systems are developed and tested, these systems can be distributed, integrated and accessed/invoked in real-time into operational systems.

Figure 11 shows an application developed in banking domain to advise users on mortgage loans based on banking or financial institution selected. It is invoked into financial website by embedding the expert sessions with the user whenever user clicks on tab conversion.

BI tools such as data mining tools play major role in building recommender systems. Recommender systems [21] have become wide-spread and many websites have these systems now for better user and consumer personalized experiences and to discover items (products, contents or services) by mining huge transactional data gathered through customer interactions. Firms like Amazon, Netflix provide unique experiences to their users using recommender systems. Based on understanding and analysis of individual user’s browsing and buying patterns, the websites serve the personalized web pages. This helps such firms to attract and retain (customer intimacy: deeper understanding of individual customer preferences) the customers, understand buying patterns and manage resources well.

There are four basic types of recommendation approaches cited in literature [21]-[23] a. collaborative filtering: items are recommended based on liking of similar users, b. content-based filtering: items are recommended based on user’s past
likes and dislikes, c. demographic filtering: recommendations based on a demographic profile of the user and, d. knowledge-based recommendations (like advisor systems): items are recommended based on deep knowledge about the item-domain exactly fitting to customer needs, objectives and interests. All these approaches have limitations and strengths which bring in scope for hybrid approach that combine content-based, collaborative, demographic filtering and knowledge-based recommendations [24]. There are many hybrid systems that combine content and collaborative filtering [25], while some systems implement only knowledge-based recommender [26]-[28].

<table>
<thead>
<tr>
<th>Inputs to recommendation system</th>
<th>Outputs/functionality of recommender System</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interactions &amp; Transactions</td>
<td>Information about products &amp; services</td>
</tr>
<tr>
<td></td>
<td>Consumer Profile &amp; Preferences</td>
</tr>
<tr>
<td>User Interactions &amp; Transactions</td>
<td>Compliance, Policy &amp; Business Rules</td>
</tr>
<tr>
<td></td>
<td>Market Information (RSS Feeds)</td>
</tr>
<tr>
<td>Related Keywords</td>
<td>Related Items</td>
</tr>
<tr>
<td>Related Keywords</td>
<td>People also liked keywords</td>
</tr>
<tr>
<td>Related Keywords</td>
<td>Help Me</td>
</tr>
</tbody>
</table>

**Fig. 12 P&R system Input and Output**

### III. P&R FRAMEWORK

In this section we describe a P&R framework called Mooga.NET. Throughout the paper we will be referring to it as P&R Framework. This framework is developed with specific extensions required for a typical recommender system on top of the hybrid framework. Figure 12 shows set of possible inputs recommender system takes and functionalities it providing using hybrid of various recommendation approaches. Outputs will be displayed in respective widgets on web pages. Some of the widgets will be personalized items. Since P&R framework works on top of hybrid framework it has all advantages of such kind of frameworks [17]. P&R framework uses hybrid of ES and CBR techniques to develop recommender systems that can have all four recommendation approaches implemented. It is domain independent and can work for many types structured items including financial ones.

We describe the working of P&R framework at higher architecture level rather than going into finer details. The purpose of describing this framework is more from utility, usability and ease of model development point of view rather than comparing it with other frameworks, tools and algorithms in recommendation space. We see contributions of the P&R framework from the perspective: a) it uses hybrid AI approach to P&R: combination of deep domain knowledge using ES + machine learning using CBR, b) it is a comprehensive integrated development framework to build, test, deploy, customize and configure recommender systems using hybrid approach quickly, c) it is configured to work in real-time and can be integrated into operational systems using web services or HTTP calls, d) it includes most of the functionalities required for true P&R (collaborative, content-based, demographic as well as knowledge-based).

Some of applications developed using the hybrid framework includes knowledge-based recommender system like for mobile selection [29], car selection [18] and software selection [30]. Most of these applications use combination of ES and CBR technologies. ES technology has been used in host mode and CBR technology in assistant mode. Various parameters of CBR can be controlled and set at run-time. In product recommendation applications, ES assists and guides the user in selecting and specifying right and only relevant inputs based on user’s constraints, interests and objectives, while CBR searches and recommends the most matching products based on relevance to user’s preferences and constraints [29]. The hybrid framework supports large number of matching functions and various types of features to model problems. Rules/s can be written to cluster the products based on features to recommend the products when a product is selected by the user.

CBR engine of hybrid framework has in-built component to build associations (similarities) between feature values based on transaction history (e.g. past downloads, browse, ratings etc.). This reduces the job of explicitly defining similarities between feature values e.g. if one of the feature in book selection is *author*, similarity between author say *x* and other authors can be done in two ways a. by considering click-streams (people who clicked on *x* also clicked on author *y*), the strength of association depends upon number instances of clicks and number of users and recentness of clicks b. using some context like category of the book e.g. if the books written by author *x* are already classified under category say *Management/Marketing*, then similarities between author *x* and other authors who write books in *Management/Marketing* category would be set. First approach is like using collaborative filtering on authors while second one is clustering authors based on context of selected author.

Similarities between features values are calculated from result data-set returned in response to database query defined for that purpose. The query can be configured to have the fields (attributes) that represent context while clustering values using context or that restrict to build associations amongst values (e.g. associating authors within language of books). In above example, along with category *Management/Marketing*, other attributes like *language of book, publisher of book, type of book* etc. can be considered if they are present in database. A weight for each type of context (category, language, selected feature, etc.) is set to calculate similarity based on weighted sum method. The weights of features themselves can be store in a database table to do computation at database server. This is illustrated as follow using mathematical form.
We have tested such associations in Indian movie database (we developed a prototype for one of the movie rental website in India). To associate actors, actresses and directors, context like movie name, genre, sub-genre (like family/kids), tags given by experts (like romantic, happy, relationship) and language (like Hindi, English, Gujarati, etc.) were used. We found interesting clusters (associations) of actors who have acted in many movies of specific genre, sub-genre and language together.

We understood various requirements (including inputs from RSS feeds [17]) of a typical recommender solutions and added extensions (functionalities) to develop recommender systems quickly and integrate them into existing websites/portals. Following four broad requirements are considered to develop the P&R framework. a. It should support almost all kinds of structured items. It should support and extract contents stored in RSS formats b. P&R solution should include all functionalities like user-profiling (based on user’s static as well as dynamic derived out of user’s transactions), recommendation using hybrid of collaborative, content-based, demographic and knowledge-based approach. Recommendation should not only work for items but also for keywords, tags and selected features c. P&R solutions should be developed quickly, they should be customizable, configurable, and models should be easier to understand. The entire framework should be self-learning dynamic reflecting changing interests of users. d. It should work as a BI layer or added as a feature which can be easily plugged with fewer modifications in existing systems. Keeping above requirements, hybrid framework is extended to build P&R framework. This P&R framework supports all four recommendation approaches as described in table I and functionality as shown in table II. Functionalities are implemented in modular way using web services that are specific to P&R framework.

![Diagram of P&R framework](Image)

**Table I: Recommendation Approaches**

<table>
<thead>
<tr>
<th>Approach</th>
<th>How it is implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative Filtering</td>
<td>This has been addressed using association algorithm in-built into CBR engine as discussed earlier. This algorithm directly takes dataset (through predefined queries) from database to build associations.</td>
</tr>
<tr>
<td>Content-based Filtering</td>
<td>Using set of user defined functions and CBR approach. Functions implement logic to create dynamic user profiles (reflecting user’s changing likes/dislikes: captured feature-wise) based on click-streams and ratings. CBR is used to show most matching items whose features match with individual profiles. It can take into consideration user’s preferences set in the form of feature values e.g. if one of the feature is item language then user can set preference for language.</td>
</tr>
<tr>
<td>Demographic</td>
<td>Profiles of most similar users are searched that match demographic profile of selected user using CBR. These profiles are used to show the most matching items.</td>
</tr>
<tr>
<td>Knowledge-based</td>
<td>a. Combination of expert system and CBR approach. Expert system guides the user in selecting relevant inputs based on user’s requirement and CBR searches most relevant items matching requirements. b. CBR is used to search most similar items to given item based on feature similarities (e.g. Show Similar/Related kind of functionality)</td>
</tr>
</tbody>
</table>

**A. Major Components and Architecture**

As shown in figure 13, there are three distinct entities in overall recommendation solution a. Client system, b. P&R

**Equation:**

If $f_v$ is value of feature $F$ then

$$\text{Similar}(f_v) = \sum_{i=0}^{n} W_i * \text{Sim}(C_{fv}, C_{dv})$$

where $W_i$ is weight given to context $C_{fv}$, $C_{dv}$ value of $F$ context for database value $dv$. 

$\text{Sim}(C_{fv}, C_{dv}) = 1$ if $C_{fv} = C_{dv}$ else 0.

We have tested such associations in Indian movie database (we developed a prototype for one of the movie rental website in India). To associate actors, actresses and directors, context like movie name, genre, sub-genre (like family/kids), tags given by experts (like romantic, happy, relationship) and language (like Hindi, English, Gujarati, etc.) were used. We found interesting clusters (associations) of actors who have acted in many movies of specific genre, sub-genre and language together.
framework, and c. External systems. Client system represents website, web application or portal that needs P&R solution. P&R framework consists of all components and interfaces required to build, test, integrate, customize and configure recommendation solutions (as a part of base hybrid framework). External systems are required when specific information about the items is to be retrieved from external websites such as details of features, user ratings and reviews.

Client system is either loosely coupled to P&R framework through web services, HTTP calls or by embedding interactive sessions or tightly connected by API calls. Client CMS (Content Management System) database or web application database (henceforth called client database) is accessed in P&R framework in non-intrusive mode. Access to client database is required to synchronize data transfer from client system to P&R framework on periodic basis or based on events.

Since web services are widely used to interconnect different types of systems loosely, we restrict our discussion to only coupling using web services. P&R framework can be coupled to any client system that supports web services.

1) Database Schema

We have modeled P&R framework on three basic types of information entities: meta-content, user and transaction. Database schema and models are divided into two sets a. generic and b. specific to domain. Domain schema (table III) depends upon type of products supported. The generic schema is designed in such a way that it can support multiple applications in the same database instance. Domain specific schema contains information required for domain modeling e.g. for mobile device item attributes can be screen size, multimedia support, internet connectivity options, operating system, etc. Similarly user and transaction specific information can be added.

Generic schema is common to most of the type of items (like RSS formats support multiple types of contents: movies, music, news etc.). In generic schema, we define various generic database tables and customizable views to retrieve datasets required for modeling at development and run-time.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Major Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items you may like</td>
<td>User ID</td>
<td>List of items that user may like. This is based on discovery (using combinations of collaborative and content filtering)</td>
</tr>
<tr>
<td>Top picks</td>
<td>User ID</td>
<td>List of items based on user likes and dislikes using context-based filtering technique.</td>
</tr>
<tr>
<td>Related Items</td>
<td>Item ID</td>
<td>List of items which are logically related to given item using CBR based matching</td>
</tr>
<tr>
<td>Help Me</td>
<td>User ID</td>
<td>Knowledge-based recommendation system to help in selecting the item</td>
</tr>
</tbody>
</table>

TABLE III
INFORMATION ABOUT GENERIC DATABASE SCHEMA

<table>
<thead>
<tr>
<th>Information item</th>
<th>What information is stored</th>
</tr>
</thead>
</table>
| Meta-content     | • Basic attributes (features) of item like title, type, language, format, creation date, publication date etc. each item has unique ID that we maintain same as that of client system follows.  
• Taxonomy, Keywords, Predefined Tags by experts 
• Relationships between items, taxonomy, keywords (with roles and weights) and tags |
| User             | • Static profile: demographic information captured initially through questionnaire or account opening form: it can be personal (like age, gender, area code, education, employed) as well as user’s interests.  
• Dynamic profile (based on user’s transactions). Dynamic profile is of two types. Global profile and Category-wise profile. Global profile captures and stores what user likes in general. While category-wise profile captures what user likes in a particular category. Each subcategory-wise profile contains what items, keywords, tags, language user chooses or likes along with time abstraction and location specific profile. |
| Transaction      | • User interactions with client system (like click-streams, downloads etc.), interactions can be codified based on events like click, browse, search, download, rent, reminder, add to favourites, add to wish-list, add rating, add tag etc. |

Some of the item features we put under broad feature called keywords with role and weight assigned to it e.g. authors in book item will be treated as keywords with role: author similarly publisher will be given role: publisher. Weight indicates relevance of that keyword in that item, helps in content filtering e.g. if a book is written by two authors, weight (relevance to that book item) of each author name can be equal. Especially in movie databases, the actor names are stored in sequence, with first actor playing major role vis-à-vis the last actor, in such a scenario, first actor has more relevance hence more weight etc.

2) Models and Configurations

Similar to database schema, models are also divided into two types, generic ones and specific to domain. The hybrid framework has various interfaces to manage models and configurations. Following sub-sections describe some of the important models and configurations required for recommendation solution. Knowledge-based recommendation systems are not covered in this paper, it is discussed at length in [29].

a) Domain Vocabulary

Set of all objects (variables, domain values etc.) that are
required to model P&R using ESs and CBRs. The domain vocabulary has variables for meta-contents, users and transactions. The variable names are prefixed appropriately to discriminate logically as shown in figure 14 (most of the figures are screen shots of application EPG: Electronic Programme Guide for TV application, screen-shots are either cropped, extended to proper and readable views).

We call them lookup tables, they are indexed by some index key like C_ID_Index in figure 18. Whenever query is executed, based on values of index keys in rows, query results are appended with corresponding static information.

Dynamic queries are predefined database queries that are used by the framework to retrieve the datasets at run time in various components. This reduces efforts of writing database queries explicitly in rule-base or in functions. These queries are not hard-coded and can be added, modified and removed easily based on modeling requirements. The queries are formulated based on requirements in different components. Figure 15 shows different purposes of dynamic queries and figure 16 shows dynamic query that is used to calculate recommended keywords using collaborative filtering algorithm. Normally queries access data from database views, these views can be easily customized at database side. Figure 17 shows a generic database view called AG_Keyword and its output. This view takes last 90 days transactions and restricts associations of keywords within their roles only. That means relationship between two keywords would be built only if their roles are same. This gives greater degree flexibility in customization and helps in better modeling of filtering at website, e.g. if user clicks on keyword (say author of book), then recommended keywords would show only authors not other keywords like publishers etc. Similarly dynamic queries for collaborative filtering (or content filtering based on context) for items, tags or any other feature can be modeled.

Dynamic queries can contain variables whose values are populated at run-time.

Other type of dynamic queries includes predefined queries (accessed by IDs) for defining CBR search space, getting user profiles, item details from database and do on. Figure 18 shows dynamic query used for CBR search whenever items matching with user profile are searched. If a new feature is added in CBR search, corresponding field can be added in this dynamic query. The P&R framework stores static information about items (which is not used for indexing purpose) like language, actors, directors in memory so that they don’t have to be fetched from database every time CBR search is made.

c) Modeling using UDFs, Expert Systems and CBR

Various generic ESs, UDFs and CBR schemas are created to take care of various functionalities in recommendation solution. UDFs are useful when only single rule is required, however if there are set of rules required to build complex logic, rule-base is written. Figure 19 shows some of the UDFs
coded in P&R framework. Figure 20 shows UDF CreateClusterForContent coded to create and return cluster of items around given item. Similar to rule-base, UDFs can also be modified depending upon the functionality required. This helps to implement dynamic business rules and logic using functions. Modifications can be done even client system is live e.g. weights of feature EPG.Content.Language can be changed from 25% to 30%, the P&R framework reflects such changes immediately.

P&R framework has predefined generic CBRs modeled and configured using generic features as listed in figure 21 to address various functionalities given in table II. Word CF in feature name represents that feature similarity values are generated using collaborative filtering algorithm. Features named EPG.Content.L1, EPG.Content.L2... represent taxonomy at first level, second level respectively.

Based on modeling requirements various features and their parameters are set, figure 22 shows some of the parameters used. Models using hybrid recommendation approaches can be implemented in various ways. For example, if content-based filtering is to combined with collaborative filtering in weighted hybrid mode [24] then feature (e.g. EPG.Content_CF_CID in figure 21) whose similarity values are calculated using collaborative filtering algorithm is included in CBR (which is configured and used for content-based filtering) by setting appropriate weight. Feature can be removed or its weight is set to 0 if it is not to be combined.

Based on business logic, feature parameters such weight, similarity function, range of values are changed. For example, if language of item is one of the important features, and business policy is to show the user items in the language that user likes, then weight of language feature can be set to high compared to other features. There are lot of database and CBR functions provided to implement complex business logic using...
UDFs and rule-based ESs. For example UDF GenerateNextPushForUser, looks into each of the subcategory-wise user profile and based on relative interests of user in that subcategory it searches the items in that category using combination of filtering techniques. Hybrid framework provides testing and debugging environment along with various development interfaces. Debugging and testing can be done interactively, UDFs can be included in rule-based and inputs can be asked at run time.

d) Tag Mapping

Meta-contents are created from client database or even from RSS feeds. Since hybrid framework converts all database calls into XML, field names are converted as XML tags. Mapping is defined between source field names (which are be used to import data) and meta-contents. Figure 23 shows sample tag mapping from RSS schema to P&R generic schema. Once this mapping is defined, data can be extracted or imported from client database or RSS feeds by calling appropriate web services.

e) Reports and forms

Functionality of Form Designer and Report Designer of the hybrid framework has been extended to generate forms and reports automatically for interactive knowledge-based recommendation sessions based on features selected in CBR configuration and domain vocabulary. This saves explicit efforts of creating forms and reports. During recommendation session, appropriate forms are invoked to ask and get relevant inputs from the user while formatted reports are used to show recommended products, explanations and related information.

f) Scheduled tasks:

P&R framework has scheduler to carry out various activities especially data transfer from client database to P&R framework on periodic basis. It carries these activities using web services. List of such activities can be entered into scheduled task setup configuration file.

3) Web Services

Once development, analysis, customizations and testing of recommendation solution are done, it can be loosely integrated into operational systems through web services. There are two types of web services P&R framework uses while integrating with client systems. P&R framework web services and hybrid framework web services (shown in figure 6). Figure 24 shows partial list of P&R framework web services. The names of web services are prefixed with broad functionality they provide. Services starting with MetaContent_ are used to manage meta-contents e.g. MetaContent_ImportFromRSS is used to import data from RSS feeds. While PnR_services are used for P&R. Services starting with Trans_ are used notifying transactions (user interactions) to P&R system. In PnR services word related indicates clustering based on CBR matching algorithm while recommended indicates collaborative filtering. Some of the web services wrap functionality using ESs and UDFs so that business logic can be changed easily. For example web service PnR_GetRelatedContents internally calls UDF CreateClusterForContent which returns cluster of items around given item. If the logic for creating cluster is changed (e.g. giving more weight to feature language), it will be automatically reflected when PnR_GetRelatedContent called next time. This gives greater degree of flexibility in managing business logic based on client’s requirements.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaContent_GetCountryList</td>
<td>Category AddChild</td>
</tr>
<tr>
<td>MetaContent_GetFormData</td>
<td>Category GetCategories</td>
</tr>
<tr>
<td>MetaContent_GetLanguageList</td>
<td>Category GetChildren</td>
</tr>
<tr>
<td>MetaContent_GetTypeList</td>
<td>Category GetID</td>
</tr>
<tr>
<td>MetaContent_GetUserTypeList</td>
<td>Category GetAllCategories</td>
</tr>
<tr>
<td>MetaContent_ImportFromRSS</td>
<td>Search CreateAllCategories</td>
</tr>
<tr>
<td>MetaContent_R四季Contents</td>
<td>Search CreateCategories</td>
</tr>
<tr>
<td>MetaContent_RemoveKeyword</td>
<td>Search CreateAndDeleteCategories</td>
</tr>
<tr>
<td>MetaContent_ReleasePersonalDetails</td>
<td>Search CreateAndDeleteKeywords</td>
</tr>
<tr>
<td>PnR_DiscoverContents</td>
<td>Search SearchForCategories</td>
</tr>
<tr>
<td>PnR_GenericFiltering</td>
<td>Search SearchForKeywords</td>
</tr>
<tr>
<td>PnR_GetContentsForUser</td>
<td>Trans AddTag</td>
</tr>
<tr>
<td>PnR_GetContentsForUserAndCategories</td>
<td>Trans AddTag</td>
</tr>
<tr>
<td>PnR_GetPersonalizedContentsUsingFilter</td>
<td>Trans NotifyTransaction</td>
</tr>
<tr>
<td>PnR_GetRecommendedContents</td>
<td>User AddUser</td>
</tr>
<tr>
<td>PnR_GetRelevantContents</td>
<td>User CreateUserFromMasterUsers</td>
</tr>
<tr>
<td>PnR_GetRelevantKeywords</td>
<td>User GetUserStatus</td>
</tr>
</tbody>
</table>

Fig. 24 Partial list of P&R web services

Most of the PnR_ and Search_ services cache results in database (along with date and time) instead of recalculating again and again. These include parameter called p_Minutes, to indicate how long results should be cached in database before they are recalculated again. Based on frequency of updates of items, number of users active at a time, available hardware resources, these parameters can be set for different web services. For example, item clustering need not be done again and again unless new items are added or existing removed etc. while web services for collaborative filtering e.g. PnR_GetRecommendedContents and personalization e.g. PnR_GetPersonalizedContents can be run every fifteen
minutes or half-an-hour to reflect changing users interests.

Figure 25 shows invoking of web service PnR_GetRelatedContents. Figure 26 shows output of web service call. Most of the web services return only item IDs (we call as Content_ID and in database field C_ID, while importing item data from client database, IDs are maintained as they are in client database). This makes it easier to implement and manage at front-end and items can be displayed in various widgets based on website design policy. Figures 27 and 28 show how items are displayed in various widgets in EPG website (to respect confidentiality we have removed other details).

We will not discuss web services but one of the important services is PnR_GetPersonalizedContents. It takes three major inputs user id, top level category and criteria. It returns items in selected top category or genre (like film, sports, fashion, etc.) that user would like based on user’s static as well as dynamic profile. Criteria indicate when user profile and personalized contents should be recomputed for the active user e.g. only after at least one download in last 15 minutes (see figure 29 for major steps involved in web service PnR_GetPersonalizedContents).

Various prototype systems have successfully developed to demonstrate and test P&R framework functionality and results. Right now P&R solution is being implemented for one of the big content providers in India for their e-Commerce B2C web site selling video and audio albums/tracks online. The first case study is on P&R for mobile VAS (value-added services) application for SonyBMG and other case study includes recommendation on website for EPG for TV.
programs (TVPs).

Because of confidentiality issues we would not discuss case studies in details. The screen-shots are included in figures 27 and 28 for EPG application. One of the peculiarities of P&R in TVPs websites, recommendations have to be in real time and forward looking (forthcoming TVPs unless repeats), there is no point in recommending the TVPs which are already telecasted and not going to be repeated again on any TV channel.

Figure 30 shows steps followed while developing P&R solutions using P&R framework. TVPs in eight top level categories: Film, Sports, Children, Documentary, TV Show, News, Business and Finance and Music have been modeled and implemented. The data in client database was already structured with TVP attributes like subgenre, episode title, episode no, brief synopsis, casts, sub-casts, directors, guest-casts, keywords, tags and so on. Other information included schedules, which TVP is scheduled on what television channel, external TVRs (television ratings by third parties) etc. When data is extracted, all required information like meta-contents, user and transactions is available in client database. What kind of functionality needs to be implemented, events to be included, study of existing website/portal, widgets available etc.

Creating Generic P&R Database and Models: this is done by creating shallow copies of database schema and generic models from master setups. Defining tag mappings to extract data from client database, etc.

Creating Master Database: Create meta-contents, users, initial transactions by importing data from client database using various web services.

Modifying database, extract additional data and customize models and configurations

Are customizations required?

Yes

Testing various functionalities, UDFs, expert system results, etc. manually or through web services.

No

Integrating P&R solution into client system through web services, web services can be wrapped by writing APIs at client system to take care of additional security, calls with fewer parameters etc, testing the results after integration.

On-going improvement, fine tuning the models to get better user experience and desired results.

Fig. 30 Major steps in developing recommender solutions using P&R Framework

A screen-shot depicts the SonyBMG’s portal in Argentina, Chile and being launched in other Latin America countries. Item and context information (like genre, song title, artist name etc.) were extracted from their database into the P&R framework.

One of the challenges in personalization is to show right information on limited screen size to the right user at right time. The personalization algorithm (figure 29), in P&R framework is configured to show only most relevant items based on users interest and device screen size.

System went live in March 2008, figure 32 shows analysis of data transactional data collected from August 2008 to May 6, 2009 to validate effectiveness of P&R framework in mobile space [31].

Figure 31 shows sample screen-shots of SonyBMG WAP front-end.
V. CONCLUSIONS

Analytics has become integral part of every business. However, traditional approach of building analytics is more of off-line, ad-hoc or periodic and not integrated into operational systems. In this paper, we represented a framework for BI that uses intelligent systems such as expert system, case-based reasoning in modular and integrated way, and accessed through web-services. One of the advantages of such frameworks, they can be used to build N=1 analytics applications focused on analyzing one customer at a time. Similar to dealing with customer one at a time, N=1 analytics can be integrated into organization’s enterprise systems to deal with internal stake-holders like employees on as well as manage relationships with external stake-holders like suppliers on one-to-one basis.

Various approaches and techniques are in use to address P&R problems and develop solutions. We have given a complete different thought to P&R problem by conceptualizing and developing a unique comprehensive framework which includes various components required for P&R. We effectively used combination of rule-based expert system and CBR technologies to P&R. The benefits of using such technologies are automatically available to P&R framework like: models are easier to understand and customize, fine tune proper blend of various approaches and manage business policies and rules. Various algorithms that are implemented to provide functionalities required for P&R do not need to crunch huge amount of data like required for algorithms based on data mining techniques. They are developed in modular way based on required functionality and focus on only one entity at a time: user, item, keyword, tag or any other feature. Based on availability computational resources and number of active users, computation schedules can be modified. At run-time intelligent systems can be distributed to enhance scalability. Couple of industry deployments show utility and practicability of N=1 analytics approach in P&R space.

REFERENCES


