



Vendor Insight

Customer analytics vendor profile: Oracle Real-Time Decisions

Helena Schwenk

Premium Advisory Report
November 2011

Oracle Real-Time Decisions (*RTD*) combines data and business rules with self-learning predictive models to help organisations automate and optimise high volume operational decisions. In contrast to other popular approaches to implementing predictive analytics, *RTD* focuses on learning from each interaction and using these insights to adjust what is presented, offered or displayed to a customer. Likewise its capabilities for optimising decisions within the context of specific business goals and a report-driven framework for assessing the performance of models and decisions make it a strong contender for organisations that want to continuously improve decision making as part of a customer experience marketing, e-commerce optimisation and operational process efficiency initiative.

This premium report is published as part of one of MWD's advisory services. You can find out more about these services at <http://services.mwdadvisors.com>.

Review this research in context

For further insight around the best practices highlighted in this research and to discuss this in the context of your own organisation, you can schedule a private advisory session with our expert analysts by emailing clientservices@mwdadvisors.com or call on +44 (0)20 8099 8297.

Summary

Oracle Real-Time Decisions (RTD) provides a framework for building and deploying decision services used to analytically enhance business processes and customer interactions. At the core of *RTD* lies a learning engine that combines business rules and adaptive predictive models to deliver recommendations to operational systems while simultaneously learning from the experience.

Behind the scenes it adapts to interactions or processes using both historical and contextual information (such as the time of day or how long a customer has been on the phone) to learn what actions have worked or not worked in the past. Through this closed-loop learning process decisions are continually refined and optimised – typically in real time – based on what *RTD* knows about the interaction or process and how similar situations have played out in the past. In turn this enables *RTD* to become more prescriptive about how it targets future customers or recommendations.

Like other decision management offerings, *RTD* focuses on classic customer interaction use cases including offer management for cross sell, up sell and customer retention. However, its ability to understand and continuously learn at a granular level what attributes and factors are likely to result in more successful decisions means it is also being applied to other areas. E-commerce is one such scenario where *RTD* is being used to not only recommend offers to customers or prospects but to recommend the most optimal content, assets or layout (such as a web banner ad or a web page configuration) most likely to drive successful business outcomes. For example, its self learning recommendation engine can be used to select the best display for a website visitor, the best agent for a call, the best email composition or understand what's driving shopping cart abandonment.

While closed-loop recommendation engines are becoming more prevalent across certain industries such as retail, financial services and hospitality, there are a number of features that distinguish *RTD*'s approach. Firstly, it makes its decisions in the context of business objectives, such as maximising customer revenue or reducing service costs, by choosing the action most likely to achieve these business performance goals. Likewise *RTD* can arbitrate between goals where they compete or where there is more than one. One of the key advantages of this approach is it forces organisations to think about the measures of success by which to benchmark *RTD* performance – something that organisations often find hard to do.

Secondly *RTD* is heavily report driven. As part of the closed-loop self-learning process *RTD* gathers and reports on a variety of statistics about decision and model performance. It can help business users understand the level of acceptance across a range of offers for comparison purposes, for example, or provide an indication of analytic model quality and give a detailed breakdown of what attributes are influential as predictive drivers for particular offers. This reporting-centric approach is very useful in furthering the understanding of how decisioning technologies, specifically predictive models, work as well as providing evidence of its success or possibly failure in making those decisions. Similarly it can become a useful tool for getting business user buy-in to the decisioning process, especially when it can positively verify the approach taken compared to a previous offering. That said we are a little surprised to see that *RTD* doesn't fully leverage Oracle's existing reporting capabilities in particular *BI Publisher* to support these aims, especially given its integration with *OBIEE*.

Finally, *RTD*'s support for operational integration offers organisations some flexibility in how they implement the offering. It doesn't, for instance, prescribe to a centralised architectural design where all elements of the decision service are owned by *RTD*. Instead it can leverage its service-oriented architecture to support a more federated approach, where the content, offers and promotions, for example, are managed and owned by other source systems such as a CRM, CMS or campaign management application. In this scenario *RTD* calls components of the decision service such as business rules or third-party predictive models independently as part of execution without having to own, replicate or re-define them. Supporting a more pragmatic approach to implementation is a good thing, especially considering the levels of complexity and inter-dependent nature of decisioning applications.

Overview of offering

Oracle RTD (RTD) offers a framework for developing, maintaining and managing decision services that optimise recurring high volume operational decisions. It works by mixing together business rules and automated predictive modelling as part of a closed-loop self-learning environment where feedback is used to continuously enhance and optimise recommendations.

RTD evolved from a company called Sigma Dynamics, where its decision engine was OEM'ed by Siebel and included as an option for their CRM line of products. In 2005 Oracle acquired Siebel and then in 2006, acquired Sigma Dynamics. Although traditionally its focus centred on CRM, under Oracle's ownership this positioning has widened to incorporate a more general purpose decisioning framework – one that isn't necessarily aligned to a particular problem or issue such as next best offer, campaign management, or customer interaction management but instead one that focuses on the wider context of customer experience management across both inbound and outbound channels. As such *RTD* is proving itself applicable to a wide range of use cases including next best action and AB testing, but also analytically-driven recommendations for selecting the best product, website content, advertisement or even the best agent to handle a call. Given this focus *RTD* often finds success in organisations with large-scale customer service, point of sale, contact centre or e-commerce footprints such as those found in the telecommunications, financial services, insurance, retail and hospitality industries.

Despite its relatively low profile within the *Oracle Fusion Middleware* portfolio – especially when compared with its BI counterparts – *RTD* has a varied and noteworthy customer list including many Fortune 1000 companies. Many of these organisations initially implement *RTD* in one area of the business (such as a customer retention process) and after realising early benefits move on to expand usage to other areas of the organisation (such as all inbound customer interactions). Those that have found success tend to measure return in terms of improvements in transaction volume, customer value, satisfaction and retention.

Today *RTD* is sold as a stand-alone offering, but it is also an integral part of the Oracle BI technology foundation stack (given its support for high volume operational decision making and reporting). Likewise as part of *Oracle Fusion Middleware*, *RTD* is built on a service-oriented architecture and can integrate with key operational systems and processes through industry standards such as web services, XML SOAP and JDBC as well as share common administration, security and identity management services. Although *RTD* is tightly aligned with the Oracle BI stack, it can also be viewed as an extension to Business Process Management (BPM) where its decision services are used to continuously adapt and analytically enhance business processes in real time. At the same time this approach enables *RTD* to remain process agnostic by guiding decision making without altering the process itself; providing organisations with a degree of flexibility especially when making structural changes to business processes can be hard.

Although we see merit in positioning *RTD* as part of *Oracle Fusion Middleware*, especially in terms of its operational integration and linkages to BPM and BI, we believe it could run the risk of sidelining *RTD* as something that purely fixes the IT 'plumbing' problem of decisioning, rather than one that focuses on enhancing or optimising the customer experience. Oracle's move towards infusing *RTD* as part of its *Oracle Fusion Applications* suite will help in some way to overcome this perception and in turn give *RTD* the visibility it deserves.

Key capabilities and products

RTD's functionality is split across two core capabilities:

- **Decision Optimisation** – *RTD*'s learning engine can be programmed to listen, analyse and learn from transactions and interactions. By updating its self-learning predictive models after every interaction *RTD* is able to constantly improve its ability to predict and optimise outcomes or recommendations based on what is known about the interaction and how similar interactions have been processed in the past. Decision services takes into account historic information, but can also give information from the most recent interactions more weight in the decision-making process, as its learning engine is time aware.

Likewise all decisions are made in the context of business performance goals as a way of helping to ensure organisational alignment around decisions. An example performance goal may be to increase the number of products for cross sell or to reduce the time agents spend per call. In this scenario *RTD* is programmed to consider and arbitrate between both goals to ensure the most optimal decision is made for the customer and for the business.

- **Decision Management** – *RTD* also provides capabilities for business users to collaborate and manage enterprise-wide decisions. Although it originally started out as a learning and decision optimisation engine for a more IT literate audience, the scope of *RTD* has since expanded to focus on business users in recognition of the need to engage and enable more of them in the decision-making process. As part of this extended focus, *RTD* now supports business users in managing decision logic and eligibility rules, modelling relationships between different elements of a decision service as well as providing support for collaborating around decision strategies.

Key Products

Several UIs are used to support the decision optimisation and decision management capabilities listed above, the key ones being *Decision Manager*, *Decision Centre* and *Decision Studio*.

- **Decision Manager** – a relatively new addition to *RTD* supports the ‘management’ of decisions enabling business users to control and refine decisions throughout their lifecycle from planning, definition, execution, and optimisation. Its role based interface can be used by business users for authoring and managing decision logic and eligibility rules, while providing access and version control, search, collaboration and workflow integration.
- **Decision Center** – a Web-based application that allows IT and power analytical users to monitor and optimise deployed decision services. From *Decision Center*, users can view statistics gathered from the models and fine tune decision logic as well as adjust how decisions are made. Similarly *Decision Centre* provides a large range of analytic reports and discovery capabilities based around the performance of models and decisions over time.
- **Decision Studio** – a graphical development tool for configuring decision services, the services that allow *RTD* applications to monitor activity, gather statistics, and make recommendations. The tool is based on the Eclipse integrated development environment (IDE) and is therefore suited towards IT developers that have a basic knowledge of Java and a general understanding of application development and lifecycle issues.

In addition, the *Real-Time Decision Server* is used as the deployment and processing environment for decision services.

RTD Decision Services

There are five key *RTD* components that make up a decision service:

- **Choices** – a broad term used to describe what is presented at the end of the decisioning process. Choices are more than just offers, however, representing, for example, the selection of the best content or the best product for a given customer in a marketing context. Likewise it can represent the best provider for a given customer or the selection of the best action for reducing process defects in other contexts. Choices can also be grouped together in choice groups.
- **Rules** - are used to decide whether a choice is eligible for a recipient, for targeting segments of the population, and for scoring and ordering choices where choices need to be prioritised.
- **Predictive models** – *RTD* predictive modelling capabilities focus on two key competencies. Firstly, they are used to rank the order of choices to optimise specific business goals. Secondly, the tool’s self-learning predictive modelling capabilities learn from each interaction over time and optimise recommendations based on closed loop information.
- **Performance Goals** – define the business objectives for *RTD*, such as increasing response rates, maximising profit, minimising handling times, or decreasing customer service costs.

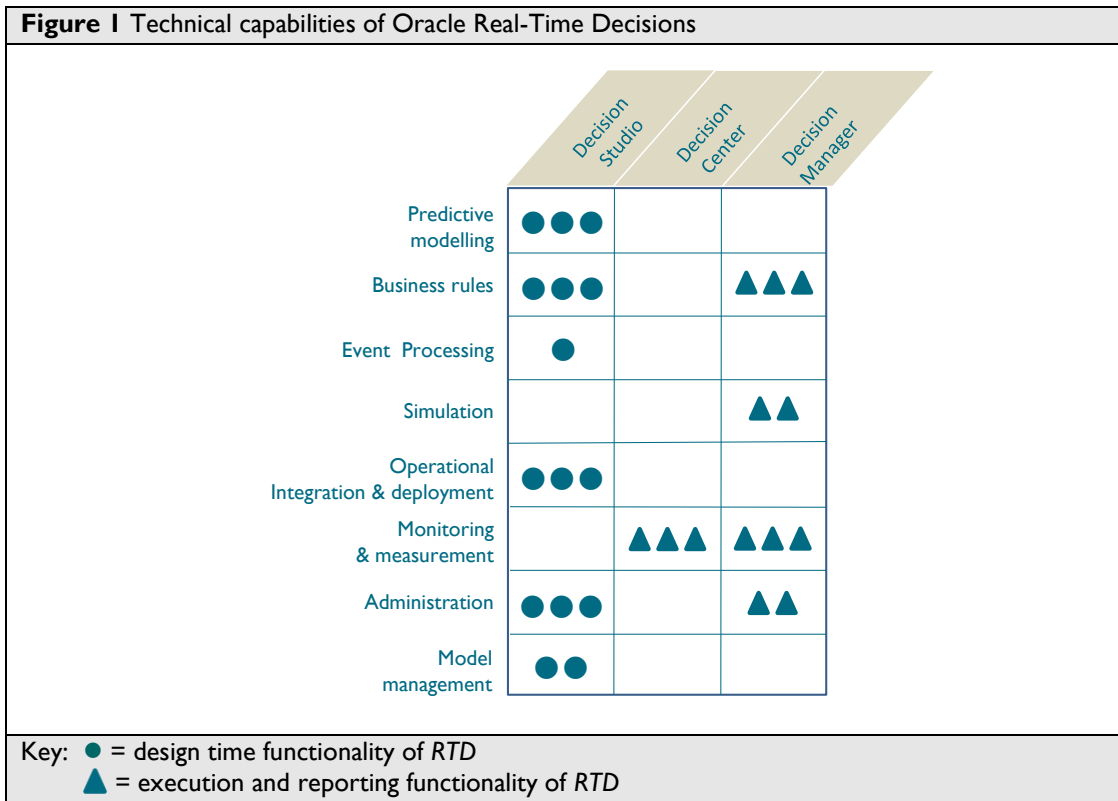
- **Decisions** – the resulting ranked list of choices that are used to optimise specific business goals. The ranked list is derived by combining rules and predictive model for achieving the most appropriate decision given each business objective.

These components are combined within a “decision service” – *RTD* terminology for a deployable application that monitors and advises business processes or customer interactions. *RTD* can create decision services using a combination of both models and rules, but can also support “rule only”-driven decisions and model-driven decisions. Rule-based decisions are typically created by business users and use explicit business rules such as “credit limit must be at least \$5000”.

In contrast, model-based decisions are driven from analysing historical data stored in Real-Time Decision predictive models. The ability to mix both approaches provides some flexibility in building decision services as it allows users to start with business rules and incorporate self-learning models as they gain more confidence in making analytically driven decisions.

Technology capabilities

The core technology capabilities offered by the different offerings are summarised below.



Predictive Modelling

There are two ways in which *RTD* supports predictive modelling. Firstly, it can integrate with non-*RTD* predictive models – such as those created by *Oracle Data Mining* or third-party models from suppliers such as *SAS Enterprise Miner* – by either reading a table which holds the output of those models or, in the case of *Oracle Data Mining*, triggering a formula for the predictive model in real time.

Secondly, *RTD* has its own self-learning predictive modelling capabilities courtesy of its data mining engine. This engine supports two key capabilities. First, it can establish correlations and causality between different data elements to predict the likelihood of a particular behaviour happening (such as the acceptance of an offer). Secondly, it can learn over time by tracking which offers are most likely to be accepted by a certain type of customer.

Assuming *RTD* is tracking the right data attributes, the self-learning process will get more accurate over time as it adjusts the offers in line with previously successful offers either in real time (such as when the customer is on the phone or clicking on a website link) or staged over a later time period (such as when they open an account, or start using their new credit card).

To accomplish this *RTD* does not employ a conventional predictive modelling work bench but instead uses *Decision Studio*. This tool is used to define predictive models in terms of what events are tracked, what data and time windows are used for prediction and what choices (such as an interest in or acceptance of an offer) need to be modelled and where they reside in the enterprise. The learning engine uses this information to help predict what influencing factors will increase the likelihood of a positive outcome happening based on what it knows about a customer or interaction and how similar interactions have worked in the past. In addition, *RTD*'s predictive models are time aware and use contextual information, allowing it to place a greater emphasis on recent behaviours or events and factor this into the model and prediction process.

Business Rules

RTD leverages one rule editor across *Decision Center*, *Decision Manager* and *Decision Studio*. The former is used to view existing business rules, whereas the latter two tools are used for creating, editing and viewing rules. What tool is used is dependent on the type of user, the context of the rule creation and the type of editing required. For example, *Decision Manager* is used by power users to author, manage and view business rules, ranging from simple to complex rule expressions. The tool is also configurable so certain users are able to control specific elements of business rule logic, such as what type of predictive model can be used for a decision service. In particular it supports two types of rules:

- **Eligibility rules** – decide whether choices or a choice group (such as an offer, promotion, display of a website banner) should be eligible for a particular decision.
- **Filtering rules** – are used as another form of eligibility. A filtering rule can be used to provide a more sophisticated way of segmenting the user population and in turn can be used to prioritise performance goals associated to a decision.

Similarly *Decision Studio* is used by IT developers to create and edit business rules – using logical and Boolean operators – as part of the *RTD* application development process. In addition to eligibility and filtering rules, *Decision Studio* can create scoring and predictive rules. The former is used to set the score for a given performance goal or to set a score for a choice, whereas the latter incorporates predictive model scores as part of the business rule logic.

Both tools can be used to view business rule logic, as can *Decision Center*, which in particular has a range of rule definition overview reports for different elements of a decision service.

And finally, *RTD* can also call business rules from third-party applications such as a campaign management tool or a product catalogue system, as well as rules authored within a business rules management system such as *Oracle Business Rules*. In this scenario *RTD* links into the third-party application and uses their rules to determine the eligibility for an offer within a campaign. As rules are created dynamically at runtime there is no scientific way of resolving conflicting or competing rules within the decision logic itself. To circumnavigate any issues, organisations may choose to monitor or go live with a *Real-Time Decision* application in learning-only mode; this will then allow them to assess the effectiveness and accuracy of rules using a more pragmatic approach.

Business event processing

RTD does not currently support any pre-canned integration with event processing tools such as *Oracle CEP*, although it is possible as part of an *Oracle RTD* services engagement. Providing this level of integration would enable *RTD* to automate the decisioning process in response to detected event patterns happening in high velocity data environments. Oracle has stated integration with *CEP* is on the product roadmap for *Oracle Business Process Management (BPM)*.

Simulation

Decision services can be tested and debugged within *Decision Manager* after first being deployed to the *Oracle Real-Time Decision Server*. Likewise, the results of test runs can be traced and viewed within *Decision Center* reports. Support for simulation and performance benchmarking is also provided through a load generator that has two functions. Firstly, it is used to evaluate the impact of potential changes to decision components such as a business rule or performance goal, but secondly, it can be used to evaluate the performance of the system under a production load. The results from the load generator can also be viewed in *Decision Center*.

Operational integration

There are three aspects to operational integration within *RTD*. Firstly, it integrates with external systems via web services, offering a lightweight and loosely coupled point of integration; in fact Oracle states that each integration point requires minimal effort and represents around 20 lines of additional code.

Secondly, *RTD* supports the integration of data for each interaction or event used as part of the decisioning process. Here *RTD* queries the source data, for example from an operational source system, in-memory cache or a data warehouse, and inserts it into the business rules or predictive models for decision processing.

Finally, *RTD* is also required to integrate data about choices. In this scenario *RTD* uses its federated architecture to support both local and centralised control of choice data. It can, for example, support dynamic access to third-party systems such as Siebel's offer management application as well as providing support for integrating choice data from a 'local' *RTD* repository. This is the most complex form of integration required because it involves multiple and dynamic API calls between both *RTD* and third-party systems.

Deployment

Decision services are created and deployed within *Decision Studio* where they are executed as Java code in the *RTD Server* from XML metadata generated within the tool. As part of this workflow a management service on the server receives the metadata and compiled decision service files and stores them in the *RTD* repository, before loading the decision service into memory ready for processing. It is also possible to state what type of deployment is required within *Decision Manager*, such as whether it's for development, QA, or production.

Monitoring and measurement

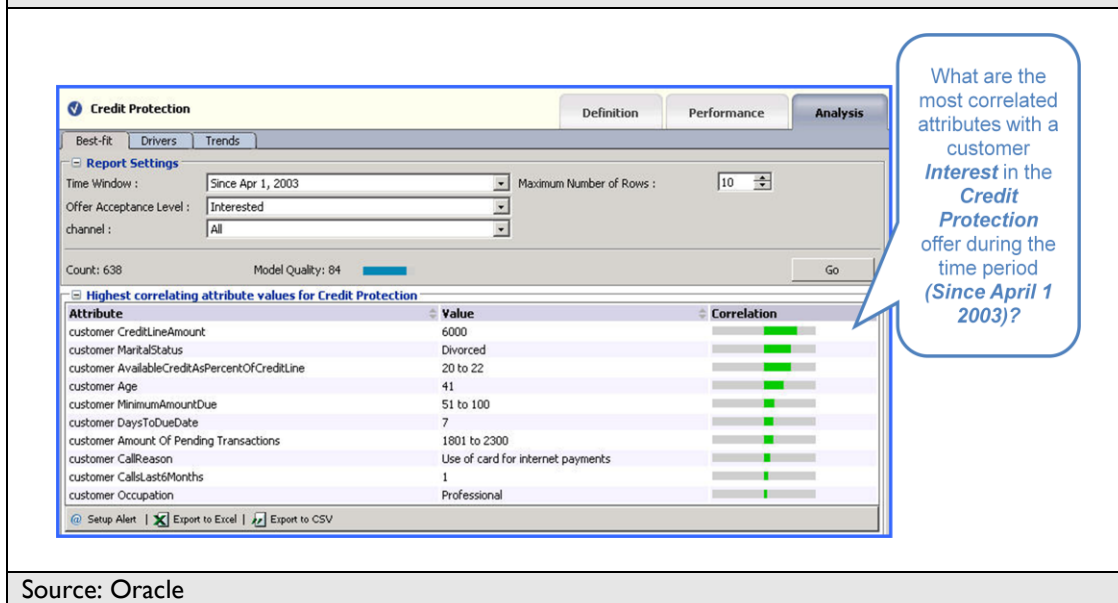
One of the design philosophies behind *RTD* centres on using reporting, visualisation and discovery capabilities to demonstrate what the *RTD* engine has learned and what lift in performance has been realised by using it. It supports this aim by gathering, monitoring and analysing the statistics and attributes of deployed decision services and uses this information to report and benchmark – at varying degrees of detail – decision service performance.

The statistics and results from deployed decision services can be viewed in a vast number of *Decision Manager* and *Decision Center* reports and alerts. Some of the questions these reports help to answer include:

- How is each choice performing? The *RTD* report is able to show for each choice (such as a gold card, platinum card or miles card) how many customers were interested or accepted the offer as a percentage of all customers who were presented with the offer.
- What attributes are influential predictors of a choice? This visualisation shows the highest correlating attributes that show the likelihood of an interest in a choice. For example, for a choice such as a reduced interest rate, the most likely correlating attributes may be where the customer has called to do a balance transfer and the percentage of available credit is less than 5%. Likewise users can drill into the detail of these predictors and compare them against the total population and over time.

- How are the predictive models performing? The reports can show the predictive lift generated by the self-learning models, which indicate how much better those models are at identifying the “best” choices versus a random selector.

Figure 2 A report illustrating the attributes that are influential as drivers of predictiveness for an interest in Credit Protection



Source: Oracle

As demonstrated in figure two, these reports can be very useful in helping organisations pinpoint and analyse sub-optimal aspects of process and decision performance. One such example comes from a large insurance company that used *RTD* predictive driver information to discern that humorous subject headings in an email were less likely to be opened over more casual subject headings for a particular young audience segment. This in turn led the company to revise and redesign their marketing materials based on this analytic insight.

All decisions defined within *RTD* can also use a control group. This enables users to compare results and show the performance gains (i.e. the lift) achieved by *RTD*. For example, in a cross-selling application for a call centre, the control group may return a random offer, whereas *RTD* returns a targeted offer based on its predictive models; the results of both groups can then be compared. Similarly it is possible for a user to define the control strategy for each *RTD* decision where the assignment of a control group and the creation of decision logic is outside of the *RTD* environment.

Although *RTD* doesn't directly leverage *Oracle Business Intelligence Enterprise Edition (OBIEE)* – and in particular its *Publisher* reporting engine – for reporting on choice effectiveness and predictive drivers, users are able to export the statistics associated with *RTD* self-learning models into a relational format whereby they can be analysed and sliced and diced within the context of a BI environment. Reports created from this data can be used for example to identify which offers are performing well compared to those that aren't and how choices are trending over time.

Model Management

RTD's data mining engine supports automated model building and maintenance as part of its operation, so some of the manual tasks associated with model management aren't applicable to its predictive models. That said, it's very easy to create many, possibly thousands, of analytic models within *RTD*, so support is provided to check and report on the performance of these models. For example, *RTD* includes the ability to validate the quality of the models, rebuild and recalibrate models when new target attributes occur, and validate the accuracy of models when there are new sources of data or new outcomes.

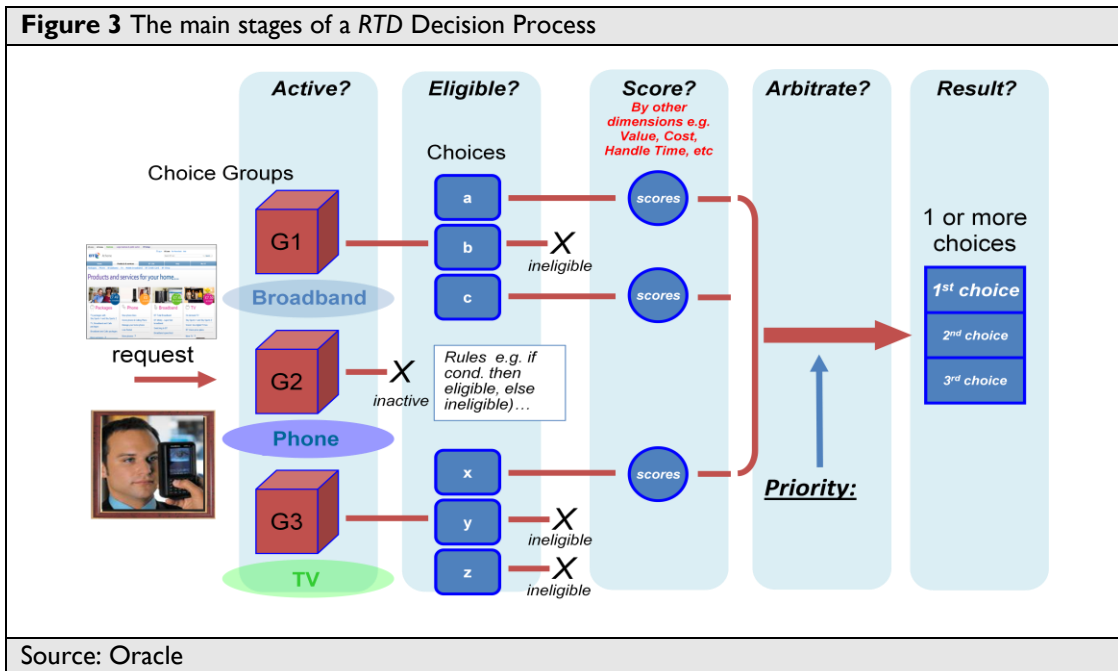
Administration

Decision Manager is the main administration tool for business users who want to create, update and manage decisions within the *RTD* repository. In particular the tool supports role-based access to the repository so only relevant stakeholders such as a marketing manager, content team or channel manager can view the necessary objects and data related to a particular choice. Likewise *Decision Manager* uses the notion of projects to describe all the objects associated with a piece of work. Projects are displayed within a tree-like structure in the *Decision Manager* interface where users are able to view, collaborate, test, version and commit changes for that project to the *RTD* repository. Audit trails of active and committed changes are also maintained.

As part of the *Oracle Fusion Middleware platform*, *RTD* can also integrate and share some of its common administration features including those for security, performance monitoring (such as viewing *RTD* log files) and starting and stopping *RTD*. Similarly *RTD* administrators can leverage the *Oracle WebLogic Server Administration* console to set up users, groups, data sources and clusters.

Decision workflow

The *RTD* decision process is typically kicked off by an external event. In our example this will be an inbound Interactive Voice Response (IVR) call to a contact centre. As part of this interaction the IVR call is transferred to a contact centre agent where a CRM application requests a decision and connects to *RTD* to establish a session. At this time as much contextual information as possible about the customer and the call is passed to *RTD*.



In this *RTD* decision workflow example the first part of the decisioning process involves assessing choices. A choice represents the offers that will be presented to a recipient as a result of a decision process and where at least one choice group is included from the choices selected. In our example a choice group determines a grouping of offers from a telecommunications company into categories such broadband, phone or TV. Next, eligibility rules are associated at a choice group level, so if for example a customer is an inactive phone user, this choice group can be ruled out at this stage.

The next stage of the process involves running through a set of eligibility rules at the choice level to determine which choices pass on to the next stage of the decision process. In our example some of these choices have been marked ineligible if, for instance, the customer already has a particular broadband package or if the customer has already rejected an offer for the same product in another channel within the last few weeks.

After deciding eligibility, the analytics engine predicts the customer's likelihood of accepting each eligible choice. This stage involves leveraging *RTD*'s predictive models, but it's also feasible to incorporate scores from third-party propensity models as well. In determining both a customer's eligibility and likelihood, *RTD* uses business rules, predictive models and a combination of the customer profile (such as demographics and behavioural measures) and real-time interaction information (such as call reason, type of service request, or Web click stream events).

Choices are then scored and prioritised using a scoring formula defined by users within *RTD*. The scoring method itself can involve using scoring rules, involve scoring using a mathematical formula (for example the distance between two points) or one that can score using a predictive model. These scores are calculated within the context of performance goals, the key performance indicators with which the organisation has chosen to measure success. Common performance goals include increasing response rates, reducing call handle times, increasing customer satisfaction and the number of products per customer. Where there are several performance goals for a choice and decision, *RTD* can arbitrate by scoring each choice against each performance goal or weighted combination of goals. For instance, in our example the decision to present an offer could be made in the context of a customer retention performance goal, where it is heavily weighted for a particular segment of the customer population. The end result of this arbitration stage is a numeric score that is produced for each choice.

The final stage involves comparing each of the choices and ordering them by decreasing total score to identify the top recommendations to be returned to the calling operational or third-party fulfilment system.

Channel and partner strategy

Oracle has a well-established partner programme that encompasses consulting providers and systems integrators, resellers and technology partners. The company has a worldwide network of more than 20,000 partners and more than 9,000 ISVs. They also have over 10 million developers – one of the largest developer communities worldwide.

Services and support

Oracle offers a range of consulting, education and support services specific to *Real-Time Decision*. In particular Oracle Consulting's *Rapid Start for RTD* is designed to kick-start an initial implementation by providing a framework for designing, building and deploying a *RTD* application. Normally a typical *RTD* project focuses on addressing a key business challenge, such as increasing cross sell, up sell, and customer retention, optimising A/B testing, improving the analytically-driven personalisation of customer interactions and product recommendations. Where an initial deployment has proved successful many customers go onto expand their deployments over time to include additional business functions, channels, customers and users. For example, if the initial deployment has focused on retention, subsequent phases may focus on cross selling as well as expanding and increasing the use of real-time analysis and self-adapting models across multiple areas of the business.

From a support perspective Oracle premier support customers have access to product updates and enhancements, and can file service requires through the *My Oracle Support* portal. The portal provides round-the-clock online customer support enabling users to log and track issues, interact with community peers and search the knowledge base across a range of content including podcasts, webcasts and company blogs. In addition the software configuration manager capabilities of *My Oracle Support* allows for automated system health checks and patch advice.

Advanced customer support services are available for *Oracle Fusion Middleware* – including *RTD* – for an additional cost

Standards

The design philosophy of *Oracle Fusion Middleware* is based around four tenets – ‘complete’, best in class’, ‘integrated’, and open’ – and this influence also extends to *RTD*. As such the offering is built on a service-oriented architecture using industry-standard J2EE application servers (such as Oracle WebLogic Server and IBM WebSphere) as well as being able to integrate with wide range of web-based, client-server, and mainframe-based architectures through industry standards such as XML, SOAP, and JDBC.

That said, PMML is currently not supported by *RTD* predictive models and nor does *RTD* support business rules standards such as Rule Interchange Format and Production Rule representation.