

DECISION
MANAGEMENT
SOLUTIONS

James Taylor
CEO

Real-Time Responses with Big Data

A White Paper

Organizations are moving to real-time responses and making optimal, real-time business decisions by detecting rapidly changing situations from massive volumes and various sources of data.

Organizations make decisions that range from one-off strategic decisions to monthly and weekly tactical decisions to high-volume, high-speed front-line operational decisions. These front-line decisions must increasingly be made in real time. Mobile devices, cloud computing, social media, sensors, and cameras are now ubiquitous and produce massive amounts of changing data. To extract the maximum value from dynamic and perishable data, organizations need to process data much faster and take timely action. Whether marketing to customers and prospects, offering proactive support, detecting and preventing fraud, or managing the Internet of Things, real-time decision-making is essential.

Responding in real time requires systems to make operational decisions automatically. Agile, analytic, and adaptive Decision Management Systems combine business rules and predictive analytics to render tailored recommendations. Event processing adds correlation and pattern detection on a scale of millions of events and data streams in motion at microsecond speeds.

Event-based Decision Management Systems enrich event-based data with traditional and big data sources and determine when and why a real-time response might be required. By leveraging decision engines based on business rules and analytics, Decision Management Systems can determine what is the best, most effective business response for every interaction. A combination of Oracle products allows organizations to respond effectively in real time. This solution brief discusses Oracle Event Processing and Oracle Real-Time Decisions, which are at the heart of a real-time business decision solution.

For more on Decision Management best practices and technologies, please contact us at info@decisionmanagementsolutions.com

CONTENTS

Systems for Real-Time Response

Oracle's Solution for Responding in Real Time

Conclusion

Appendix: Oracle Event Processing Features

Appendix: Oracle Real-Time Decisions Features

Sponsored By

ORACLE

Systems for Real-Time Response

For an organization to respond in real time, it needs to acquire or develop Decision Management Systems to capture, filter, and analyze data, and make decisions in real-time. Such systems need to be able to rapidly determine that a response is required and also intelligently determine what the relevant and appropriate response should be—they need to decide when and how to act. In an era of big data, organizations increasingly need to ensure that a response is delivered in real time so more event-centric Decision Management Systems are required. This section explores the power of Decision Management Systems, the impact of big data, the capabilities of event-based Decision Management Systems, and some common use cases for these kinds of real-time response systems.

The Power of Decision Management Systems

Decision Management Systems improve profitability (better results, less fraud, lower costs), increase compliance (with policies and regulations), and deliver more precise and effective risk management. They offer organizations the ability to deliver on predictive real-time business opportunities with specific time and location content in context for 1:1 customization. These systems maximize the value of assets while allowing staff to be reallocated to higher-value decisions and deliver unprecedented business agility with greater automation at the heart of their operations. Building a Decision Management System involves:

- ▶ Managing decision logic for transparency and agility
- ▶ Embedding predictive analytics for analytical decision making
- ▶ Optimizing results given real-world trade-offs and simulating results
- ▶ Monitoring and improving decision-making over time

Decision Management Systems are described in more detail in *Decision Management Systems* (Taylor, 2012) but there are some critical characteristics:

Agile, Analytic, Adaptive

Decision Management Systems are agile, analytic, and adaptive.

- ▶ They are agile so they can be rapidly changed to cope with new regulations or business conditions.
- ▶ They are analytic, putting an organization's data to work improving the quality and effectiveness of decisions.
- ▶ They are adaptive, learning from what works and what does not work to continuously improve over time.

Business Rules, Predictive Analytics, Optimization

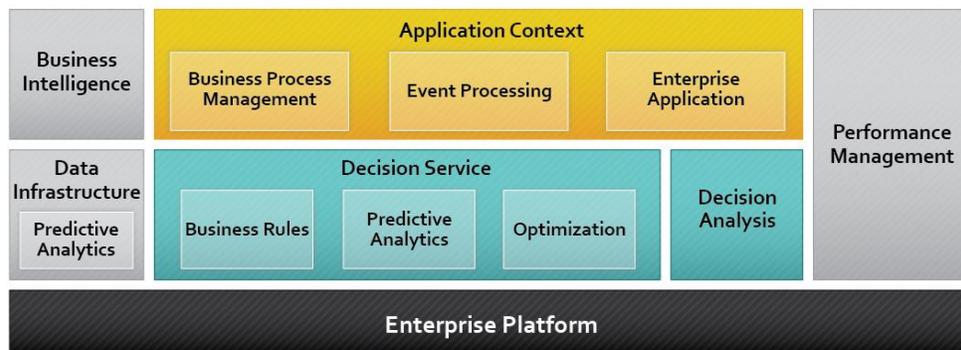
Building agile, analytic, and adaptive systems requires a focus on decisions and the effective use of proven technologies—primarily business rules, predictive analytics, and optimization:

- ▶ Business rules harness best practices, expertise, policies, and regulations. They deliver agility by making it easy for business owners to make changes when they need to.
- ▶ Predictive analytics explain propensity, association, segmentation, and risk empirically to allow analytics to be embedded in systems.
- ▶ Optimization handles trade-offs and constraints, and when combined with ongoing monitoring and review of decision-making, allows these systems to be adaptive to change.

Fit in Enterprise Architecture as Decision Services

A Decision Management System pulls together these technologies and deploys them as a decision service that can be consumed by business processes, business events, and applications. These decision services fit at the heart of an enterprise architecture as shown in Figure 1.

Figure 1: Decision Services



Source: Taylor, *Decision Management Systems Platform Technologies Report, 2012*

The Impact of Big Data

Big data has made the value of Decision Management Systems increasingly clear. Big data means that more data, of more types, is arriving more quickly.

- ▶ The need to handle larger data volumes can reveal the limitation of human interpretation and overwhelm decision-makers. Decision Management Systems, in contrast, automate decision-making and use more advanced analytics so they scale to larger data volumes.
- ▶ The increasing variety of data means that organizations handle more types of data. This data comes from both inside and outside an organization and can

include more unstructured data types such as video, audio, or text. Decision Management Systems pull together whatever data will improve a specific decision and so leverage data stored in a variety of formats.

- ▶ As this data arrives more quickly, organizations have to make decisions more quickly and have to deal with data “in motion”—streaming data—not just data at rest. Automating decisions in response to this fast-moving data is essential.

The increasing velocity of data adds to the general pressure on organizations to respond in real time. This is leading to an increasing focus on event-based Decision Management Systems.

Event-Based Decision Management Systems

To respond in real time, organizations need to filter, correlate, and process large amounts of fast-moving data in multiple formats. Event-based Decision Management Systems deploy decision agents, not just decision services, to deliver real-time responsiveness from high-velocity data. They combine pattern detection with business decisions and recognize that business decisions must be shared between event-centric and process-centric solutions.

From Decision Services to Decision Agents

The essence of a decision service is that it makes a decision, or answers a business question, for a business process or other service. Separating out the decision-making in this way allows for decision-making to evolve independently and to be reused in multiple contexts. An event-based system needs a similar capability to determine an appropriate action. Generally, however, this decision needs to be made as part of the overall response to some event and so needs to be more tightly coupled to the detection of that response. In addition it is often more granular, with less coupling to other elements of the system. To describe this, it can be helpful to talk about decision agents rather than decision services.

From Data at Rest to Data in Motion

In a decision service it is generally true that all the data needed for the decision is either passed in as parameters or available for access from the service. This implies that the data required to make the decision is mostly “at rest” in that it is already stored somewhere it can be accessed. In an event-based Decision Management System some of the data may be available in this way, perhaps as an unstructured or NoSQL format, but some is not. This “in motion” data is data that must be processed or acted on before it is saved to a data infrastructure. Data associated with event streams is critical in these systems and so decision agents need to be able to interact with data at rest *and* data in motion.

Pattern Detection and Business Decisions

A decision service is generally invoked as part of executing a business process. As the process works its way through the various steps involved, it invokes the needed decision services. In an event-centric, more real-time scenario, however, it may not be obvious what the right next step is or in what order steps should be executed. Event-based Decision Management Systems detect patterns in events, in the data that is in motion, to determine what the next step might need to be. In addition, these patterns become data that must be consumed in the decision-making because the specifics of that pattern are often relevant to the action selected.

Decisions Shared Across Processes and Events

For many organizations the reality is that they need a mix of process-based and event-based systems. They have standard processes that work through a set of steps in a given order. They have situations that must be detected so they can be responded to. And they have processes that generally work through a certain set of steps but that occasionally need to detect exceptions or abnormal conditions and so respond differently. In this real-world environment it is important that business decision-making—who is a good customer, what's the best offer for this customer, how suspicious is this fraud network—can be shared between process-based and event-based scenarios and is not embedded in either the process management or event correlation environments.

Use Cases for Event-Based Decision Management Systems

There are many situations where organizations must respond in real time and many use cases for event-based Decision Management Systems. There are four, however, of particular note:

- ▶ Real-time marketing
- ▶ Proactive customer support
- ▶ Fraud detection
- ▶ Internet of Things

In all four there is value to a real-time response, value to situational awareness, and a need for increasing sophistication in the response selected.

Real-Time Marketing

Organizations are trying to ensure that their marketing is relevant, situational, and targeted. As a result they are dividing customers and prospects into increasingly smaller segments using analytics and then focusing in to effectively target every prospect individually rather than relying on blanket campaigns that send the same offer to everyone. These may be up-sells, cross-sells, coupons, or retention offers.

They can include geospatial precision and temporal relevance and be delivered in every channel from mobile and text to chat and call centers.

Some organizations are evolving their marketing and support systems to a next-best action approach that coordinates all possible actions and selects the one that is most likely to move a customer conversation along and build long-term value. The list of possible actions can include an attempt to sell a complementary product, encourage use of a service a customer already has, recommend a product fix or FAQ answer, or ask customers for clarification on their circumstances.

Many organizations also are personalizing their interactions with customers and prospects using everything they know about a prospective customer, making a micro decision specific to that prospect. As customers are increasingly mobile and social, and as more of an organization is instrumented, the challenge is to integrate long-term trends and assessments of customers with more immediate, event-based information about them to maximize the efficacy of the offers being made.

Event-based Decision Management Systems can focus marketing efforts to make 1:1 marketing decisions in real time so that relevant and appropriate marketing responses, imagery, and content can be delivered in every channel.

Proactive Customer Support

Retaining, developing, and supporting customers have always been focus areas for organizations. For large organizations and for those whose clients interact with them wholly or largely electronically, servicing and developing customers is something that must be done by *systems*. The advice given to a customer, and the willingness of an organization to invest in helping each customer, varies depending on the value of that customer. For these and a myriad of other reasons, these systems must make decisions about how and when to help.

The same move to real time that has impacted marketing is also impacting customer satisfaction. Customers working online or using a mobile application late at night need support immediately. The challenge is proactively identifying that they need this support—if an organization has to wait until customers realize they are in trouble, it may well be more difficult and expensive to “fix” the problem. Detecting that a user’s behavior is suggestive of someone who needs support and correlating patterns in the stream of events generated by the behavior allow for proactive, automated support to be offered or for a technician to be dispatched. This requires event-based Decision Management Systems.

Fraud Detection

Many organizations suffer losses from fraud and abuse. These range from fraudulent claims for services that were never performed, to applications for credit for people that don’t exist, to orders that include bribes and illegal payments. In every case an

organization must decide whether to accept the transaction as valid, reject it, or investigate it for fraud. These decisions are high volume and must increasingly be made in real time. Such systems might determine that a transaction or application is fraudulent, that someone is not who they say they are (identity fraud), that a supplier or provider is fraudulent, or that a transaction involves a fraud network, with multiple people collaborating such that each piece looks valid but the overall effect is fraud.

As those committing fraud have become more sophisticated and as more fraud can be conducted entirely electronically, it has become essential for organizations to be context-aware when assessing fraud risk. When organized gangs are conducting electronic fraud, the time between the start of an attempt and a loss can be very short. This puts enormous pressure on the fraud detection system to be both real time and able to handle rapidly arriving and changing data as part of its assessment. Add in the need to assess groupings and locations to spot abnormalities, and the value of event-based Decision Management Systems for detecting fraud becomes clear.

Internet of Things

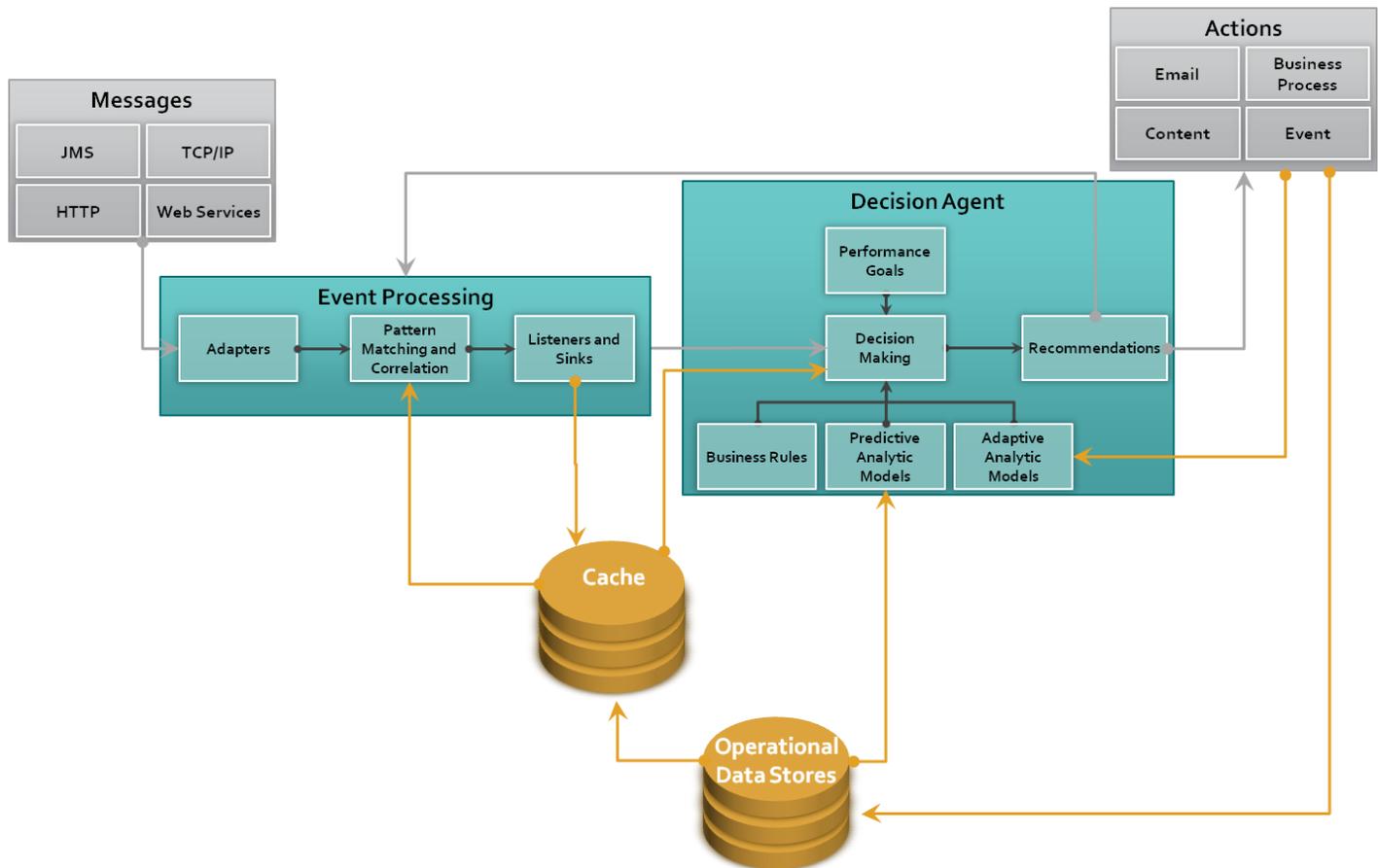
As more and more things can be identified both in the real world and in a matching virtual world, it becomes useful to connect them in an Internet-like structure. This begins by attaching RFID tags or other sensors, embedding Java on physical things, and connecting them, often wirelessly, to record where they are and what they are doing. The real world, or at least the bits with sensors on, then can be tracked and a virtual representation manipulated using computers. Actuators and controllable devices increasingly allow these things to be manipulated in turn. More of these devices now embed their own computing power and are able to make their own decisions, often while collaborating with nearby objects or instead of communicating with some central network.

When things are connected, using smart sensors on devices, they create a real-time flow of information about what is going on (or not going on) and Decision Management Systems are required to determine the appropriate actions to take. Such systems make decisions about what to do (based on inputs, locations of things, etc.) and then tell the things to take specific actions. This has begun with smart devices (such as smartphones in the hands of mobile customers) but will become increasingly common with physical equipment in factories and the outside world. The need to respond to this ecosystem of machines in real time with appropriate instructions will require a generation of event-based Decision Management Systems.

Event-Based Decision Management System Architecture

A typical event-based Decision Management System involves integrating event processing and a decision agent with a cache, a data layer, and with inbound and outbound channels as shown in Figure 2 below.

Figure 2: Solution Architecture



Source: *Decision Management Solutions*

The architecture involves:

- ▶ Identifying and integrating a series of data and message feeds.
- ▶ Processing the events on these feeds by defining patterns of interest, optionally by integrating with existing data sources and caches of those data sources.
- ▶ The ability to invoke a decision engine that selects the best available asset or choice from a pool to meet defined performance goals.
- ▶ Using predictive analytic models, adaptive analytic models, and explicit business rules to make these decisions appropriately.
- ▶ Feeding recommendations back to the event processing engine or to external systems to take action.
- ▶ Closing the loop to learn what works and what does not.

Oracle's Solution for Responding in Real Time

The combination of Oracle Real-Time Decisions and Oracle Event Processing delivers the core capabilities for building a real-time, event-based Decision Management System—correlating events, managing decision logic, embedding predictive analytics, optimizing results, and monitoring and improving decision-making. Key features of the solution include:

Table 1: Key Features

Key Capability	Value Proposition
Event correlation, business rules, and predictive analytics in combination	Some real-time response solutions focus on event correlation, on business rules, or on predictive analytics. With Oracle Event Processing and Oracle Real-Time Decisions working together, the solution balances these capabilities, maximizing the flexibility and power of the decision-making systems that can be built with it.
Scalability and flexible deployment	Oracle Event Processing offers extreme scalability and deployability thanks to its OSGi open standards-based design. With Oracle Event Processing deployed at the edge of a network and handling events as they cascade through an architecture, event detection and correlation is widely distributed and close to the event source. This improves responsiveness and contextual awareness while lowering latency. At any point Oracle Real-Time Decisions can be invoked or used to learn from the patterns detected due to its flexibility of deployment and its support for a shared Oracle Coherence cache.
Broad support for heterogeneous environments	The Oracle solution supports externally managed data (relational and NoSQL), content, business rules, and predictive analytic models. A wide range of event sources can be handled and event handling is extensible with Java. All this makes it an ideal product for organizations working in a heterogeneous environment.
Adaptability and robustness in the face of change	The overall solution is very robust in the face of ongoing change. Oracle Event Processing allows new patterns and queries to be deployed to a live instance while Oracle Real-Time Decisions allows similar changes to business rules and analytic models. Oracle Real-Time Decisions provides support for analytic models that learn while automation of the full analytic lifecycle allows customers to use hundreds of regularly updated analytic models in production.
Extensibility	The solution is extremely extensible. The OSGi and Java-based approach to event processing means that new functions and capabilities can be made available to the pattern-matching engine while the support of Oracle Real-Time Decisions for external business rules and analytic models allows the decision-making to be broader than that which is embedded in the Oracle Real-Time Decisions environment itself.

Oracle Architecture

The Oracle perspective for responding in real time is shown in Figure 3. This perspective centers on an event-processing server and a decision engine. In addition, the combined Oracle Event Processing and Oracle Real-Time Decisions solution

involves a learning engine, an event processing network interface, and a decision management interface.

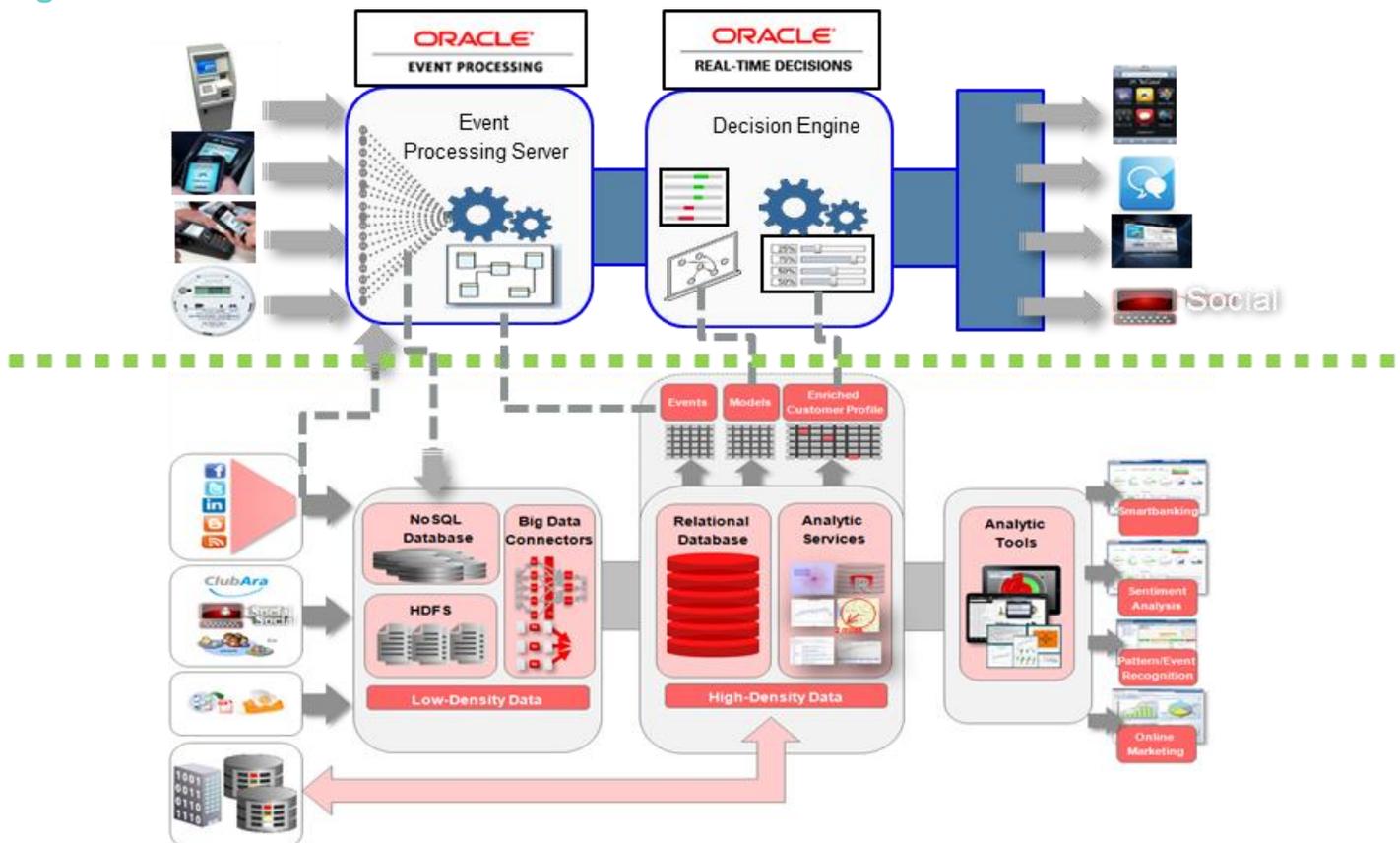
Event Processing Server (Oracle Event Processing)

The event processing server integrates, enriches, and then correlates incoming event feeds. It detects patterns in these feeds that require a response.

Decision Engine (Oracle Real-Time Decisions)

The decision engine combines business rules and predictive analytics to make decisions and deliver these decisions as scalable decision services. It supports an automated test-and-learn framework and can adjust decision-making based on performance objectives as well as results from the learning engine.

Figure 3: Oracle Solution Architecture



Source: Oracle

Learning Engine (Oracle Real-Time Decisions)

The learning engine discovers meaningful and relevant correlations in an organization's data. Designed to handle large data volumes and to automatically learn

from each transaction fed to it, the learning engine can be deployed independently or integrated with the decision engine to create adaptive analytic models.

Event Processing Network Interface (Oracle Event Processing)

The interface allows the definition of various elements that work together to process incoming events. Monitoring, extension, and integration with external data sources and caches all are able to be managed in a visual environment.

Decision Management Interface (Oracle Real-Time Decisions)

The final component is a collaborative interface for managing the decision management lifecycle. The interface empowers business, IT, and analytic teams to work together to plan, define, execute, and optimize decision management programs across channels.

Approach

The first step in developing a real-time response solution is to configure the event processing engine. This involves identifying the message streams in the environment. These are fed into the engine and correlated with each other and with data stored or cached. Patterns are defined to process these events, and an event processing network is defined.

After the event processing engine is configured, it can be connected to the decision engine. This involves defining the options to be selected between and the performance measures that will allow the decision engine to choose the best option. Any business rules that constrain the choices are specified and analytic models are built to predict behavior and segment customers or transactions. The decision-making involved is defined, managed, and deployed.

At run time, events are received, enriched, matched, and processed. When necessary the decision engine is invoked to determine the eligible choices and identify the best choice given the performance measures and analytics. This choice or recommendation can be fed back to the event processing engine or passed out as a response. The decision engine then closes the loop by recording decision performance information.

The decision engine automatically divides transactions into test and control groups, and the designer of the decision decides how the transactions will be split between the test and the control group; this logic can be adjusted over time as necessary.

Results from decision-making are fed back into adaptive analytic models and used to monitor the overall performance of the system.

Conclusion

There is a growing need for organizations to respond in real time. Consumer demands for immediate answers, the growth of local and mobile interaction, the need to control devices interacting with the real world, and the general pressure to do everything faster are pushing organizations to develop event-based Decision Management Systems. Such systems address real-time marketing needs, offer proactive customer support, detect fraud, work with the Internet of Things, and handle a myriad of other real-time response use cases.

The combination of Oracle Event Processing and Oracle Real-Time Decisions offers a powerful solution for building these systems. Oracle Event Processing detects the need for business responses by correlating and pattern matching very large numbers of events. It can be deployed from data centers to the very edge of an organization for maximum scalability. Oracle Real-Time Decisions combines business rules, for policies and regulations, with advanced analytics that learn from all of an organization's data. This combination means that optimal business decisions can be rapidly made when a need is detected. The learning engine and adaptive analytics in Oracle Real-Time Decisions help close the loop, identifying and building on successful decision strategies.

Both products' support for heterogeneous environments and for a wide variety of data types and stores means that all of an organization's data can be put to work effectively. Integrated caching and other features deliver the high performance demanded of real-time systems in a big data world. The ability to extend the products and leverage external components as well as their robustness in the face of change make them ideally suited to today's fast-changing business environment.

Organizations should be developing event-based Decision Management Systems to enable real-time response to big data. Oracle Event Processing and Oracle Real-Time Decisions offer them a powerful solution.

Appendix: Oracle Event Processing Features

Oracle Event Processing is designed to provide high throughput and low-latency processing against continuously streaming data—sometimes called complex event processing. It handles real-time correlation between incoming data streams, time sensitive alerts, aggregations and calculations, pattern seeking in the stream, etc. Critically, especially when organizations are dealing with very large volumes of streaming data, actions can be taken by processing data before it gets stored.

Oracle Event Processing is an Equinox OSGi™-based Java application server. The server uses adaptors to connect to incoming data streams (JMS, HTTP, etc.) and then converts this data to Plain Old Java Objects or POJOs for processing inside the server. Continuous Query Language (CQL), an SQL variant designed for streaming data, is used to process these streams. Listeners or “sinks” written in Java interface with other systems to take appropriate action.

Table 2: Key Oracle Event Processing Features

Key Capability	Value Proposition
Pattern matching and CQL	The power of an event processing engine is almost entirely dependent on the pattern matching capabilities it offers. Oracle Event Processing has a rich set of pattern detection and correlation abilities including geospatial, mathematical, and time-windows. The adoption of the core of CQL as a standard is an added bonus.
Streams, cache, RDBMS, and NoSQL	In real-time scenarios the data that must be processed might be in motion as events or it might be static and stored in a database or NoSQL datastore. For performance purposes, some or all of this data must be cached, and in a big data world some of it is not structured. The ability of Oracle Event Processing to enrich streams with data from all these sources is critical.
Event beans and Java extensibility	While a rich set of pattern-matching capabilities is important, no product is going to offer everything an organization needs. Oracle Event Processing's Java-based extensibility allows an organization to add Java libraries and so extend the pattern language.
Live updates to queries	After an event processing engine is up and running, changes will need to be made. Oracle Event Processing allows the queries in an event processing network to be changed and updated without stopping and restarting the server.
OSGi capabilities including edge deployment	The use of OSGi as a container for the server means that the various elements can be included or not as necessary. In particular, this allows a stripped-down server to be deployed “to the edge,” running on potentially very small devices.
Event processing network management	Event processing products tend to be fairly technical environments. Oracle Event Processing's graphical management of event processing networks simplifies management of an environment over time. Each element can be configured, and monitoring can be added easily.

Pattern Matching—CQL

Like most event processing products, Oracle Event Processing can be thought of as an inverted database in which the data is dynamic and the queries are static. In contrast, a normal database has static data being processed by dynamic queries. New data elements and new events are processed against all the defined queries for the channel defined for those events as they stream in.

Each event adaptor results in a channel that is processed by a CQL processor. Several channels can be processed by a single CQL processor, and channels can feed several CQL processors.

CQL processors run queries that can output new channels, join databases and channels, and access data stored in Oracle Coherence, a distributed caching infrastructure. A certain number of events or all the events in a time window can be kept in memory as part of each channel.

In Oracle Event Processing, these queries are defined using Continuous Query Language or CQL. This language handles a range of patterns:

- ▶ Events can be filtered to reduce a stream only to events that meet a certain criteria or set of criteria.
- ▶ Pattern matching can detect a specific trading pattern or temperature pattern; for instance, using regular expressions.
- ▶ The absence of events or missing events in a sequence can be detected as well as additional or unexpected events.
- ▶ Continuous querying can calculate running totals, averages, and so on, using standard SQL structures like Group By.
- ▶ Temporal or row-based correlation allows only recent events to be considered with events dropping out of consideration as the number exceeds the threshold or the time expires.

Figure 4: Oracle Event Processing Pattern Match Query

```
<view id="OffAndOn">
<![CDATA[
ISTREAM(
SELECT meterID, alarmType, alarmDateTime
FROM MeterAlarmsInputChannel
WHERE alarmType IN('OFF', 'ON')
)
]]>
</view>

<query id="OffNotFollowedByOnQuery">
<![CDATA[
ISTREAM(
SELECT
  off.meterID,
  off.alarmType,
  off.alarmDateTime
FROM OffAndOn MATCH_RECOGNIZE
(
  PARTITION BY meterID
  MEASURES
    OffAlarm.meterID AS meterID,
    OffAlarm.alarmType AS alarmType,
    OffAlarm.alarmDateTime AS alarmDateTime
  INCLUDE TIMER EVENTS
  PATTERN (OffAlarm NotOn*)
  DURATION 10 SECONDS
  DEFINE
    OffAlarm AS alarmType = 'OFF',
    NotOn AS alarmType != 'ON'
  ) AS off
)
]]>
</query>
```

Source: Oracle

- ▶ Event streams can be joined with data stored in a relational database, a NoSQL datastore, or an Oracle Coherence cache.
- ▶ Built-in geospatial libraries allow events to be enriched with location specific data such as presence within a geo-fence.
- ▶ Complex analysis can be performed using advanced math libraries or arbitrary Java libraries that can be added to the environment.

These patterns allow event streams to be selectively processed in multiple ways.

The pattern-matching features of CQL were accepted as an ANSI SQL standard in 2012, and they now have been implemented in Oracle Database 12c.

The end points of this network of processors are a set of output adaptors that connect to a business process, call a decision engine like Oracle Real-Time Decisions, or even just send emails. This network of adaptors, channels, processors, cache, event beans, and event sinks (event-handling Java code) is an event processing network (see the Event Processing Network Management section below), and it is the core of the product.

Streams, Cache, RDBMS, and NoSQL

Real-time responses generally mean processing data that is in-motion, streaming data. Even in real-time scenarios, however, not all data is streaming. Some of the data that will need to be considered to detect a pattern that requires a business response is data stored in an organization's data infrastructure. Some of this data is likely to be cached for performance, some is going to be structured data stored in a relational database, and increasingly, some is going to be less structured data stored in a NoSQL data store. To effectively process events and determine the need for a response, the engine will need to handle all of these.

In Oracle Event Processing, CQL patterns can join information from events, from Oracle Coherence caches, from relational databases connected through an Oracle data management layer, or from a NoSQL data store. Regardless of where the data is currently stored, or if it is just streaming data that has yet to be stored, the data can be combined to enrich event streams and drive calculations or patterns in CQL.

Oracle Coherence is included in Oracle Event Processing (converted to an OSGi bundle) because its caching capabilities are of particular value in an event processing scenario. Oracle Event Processing is able to listen for updates to the cache and respond to changes as though they are events. In addition, the cache makes reference data available in memory for improved performance, and it is possible for Oracle Event Processing to write to the cache, letting the cache handle persistence, to keep the event processing element of the system efficient.

Event Beans and Java Extensibility

In order to respond in real time, it is important that an event processing engine be able to handle a range of processing without having to reach out of a high-performance event-centric environment. Oracle Event Processing enables this by allowing Java libraries to be embedded as an OSGi bundle so that the functions in those libraries can be accessed directly in CQL. The libraries, such as geospatial libraries, can be packaged with Oracle Event Processing or added by customers. In either case this allows CQL to be extended to handle new functions.

In addition, the event processing network can include Java event beans, allowing Java processing to happen during the event processing. This might be a sink, marking the end of the event processing sequence, or it might enrich or update the data being used in the process.

Live Updates to Queries

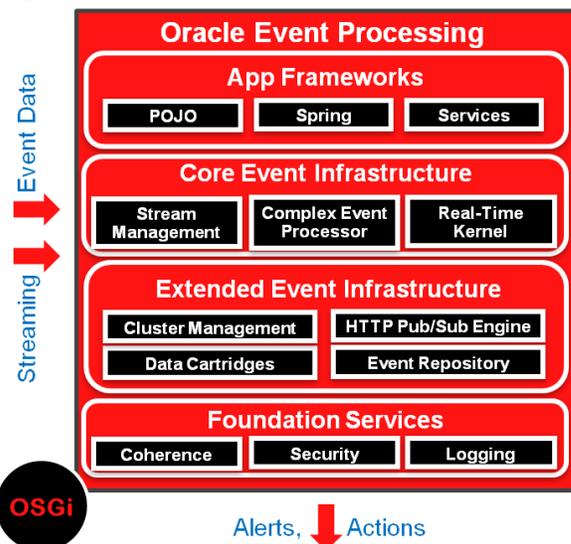
After an event processing capability is deployed, it is extremely likely that it will need to be changed. These changes may be serious, such as the integration of a new data source or a change to the kinds of events being handled, and this will require the restarting of a given event processing network. More commonly, it will be a change to a specific pattern within the event processing network. For instance, a threshold may need to be changed. Oracle Event Processing allows these kinds of changes to be deployed to a running Oracle Event Processing server without interruption.

OSGi and Edge Deployment

Because Oracle Event Processing is OSGi based, it supports being stripped down, with modules removed, to run on smaller devices. This means the same server can run on a range of devices from edge devices to gateway devices to enterprise data servers, scaling from very small to very large.

This ability to run on very small devices is particularly important for handling the Internet of Things, one of the use cases for this kind of event processing.

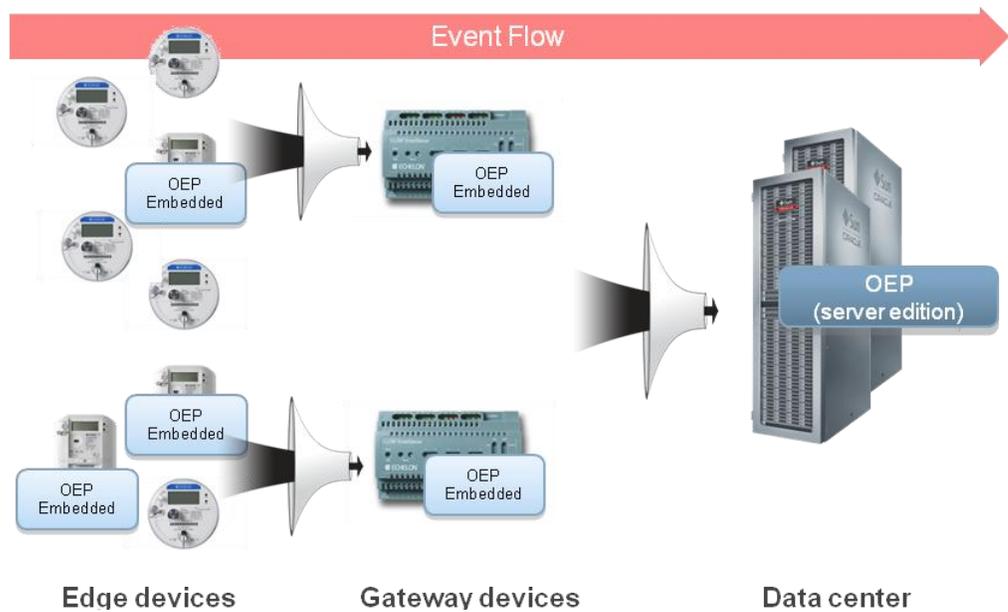
Figure 5: OSGi Architecture



Source: Oracle

This capability can be used to create a cascading architecture, such as that shown in Figure below. In such architecture, edge devices can capture and process events close to their source, feeding more significant events to gateway devices where more event processing can occur before the most significant events are handled at the data center.

Figure 6: Cascading Architecture



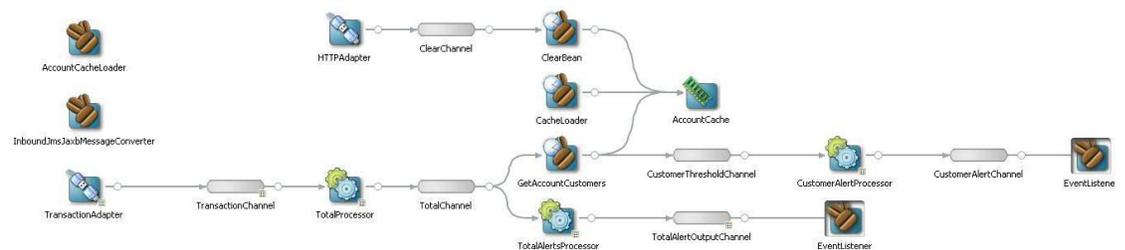
Source: Oracle

Several server components such as security, logging, JMS access, etc., are OSGi bundles reused and shared with Oracle WebLogic Server.

Event Processing Network Management

The various elements of an event processing solution are defined in Oracle Event Processing using an event processing network such as the example shown in Figure 7. The network can be reviewed and changed using the tool, and the latency and throughput between any two points on the network can be monitored at a defined time frame. This visual metaphor allows the conceptualization of the complete event processing environment and its connection to other parts of the solution.

Figure 7: Event Processing Network in Oracle Event Processing



Source: Oracle

Appendix: Oracle Real-Time Decisions Features

Oracle Real-Time Decisions is a highly developed platform for developing Decision Management Systems, especially those delivering highly targeted content in real time across a broad range of channels. A high-performance decision service engine executes business rules and predictive analytic models to make performance measure-driven decisions. A decision management interface allows the definition of learning graphs and other elements of decision-making. These learning graphs can be used by an automated learning engine to find meaningful correlations and can drive adaptive analytic models.

Table 3: Key Oracle Real-Time Decisions Features

Key Capability	Value Proposition
Business rules and predictive analytic models as peers	Some decision management platforms focus more on business rules or predictive analytics, but Oracle Real-Time Decisions provides an environment that balances the two as peers, maximizing the flexibility and power of the decision services that can be built with it.
Performance measures drive arbitration	Oracle Real-Time Decisions' use of explicit performance measures—and its support for these are predictive analytic models, explicit properties, or complex formulas—allows for effective arbitration between competing offers and decision outcomes.
Learning engine and learning graphs	Adaptive analytics and learning engines are powerful tools in decision management, but Oracle Real-Time Decisions' learning graphs allow organizations to learn at a more granular level about their marketing and offers. Combined with a tagging approach, these graphs allow organizations to evaluate the effectiveness of very specific elements of their marketing campaigns.
Adaptive analytic models	Oracle Real-Time Decisions supports analytic models that learn and identify and use predictors based on customer behavior. This helps organizations scale their use of analytics across their channels and products while reducing the cycle time to learn about market changes and new opportunities. Automation of the full analytic lifecycle allows customers to use hundreds of models in production.
Decision management interface	The decision management interface provides an enterprise view of decisions and a collaborative environment for business users to manage their decisions.

Business Rules and Predictive Analytic Models as Peers

While there are decisions that can be made only with business rules and others that can be made only with predictive analytic models, many benefits result from the use of both business rules and predictive analytic models in combination. Oracle Real-Time Decisions treats business rules and predictive analytic models as true peers, combining them to make decisions.

Business rules are used to define eligibility and to define segments used to filter customers. These business rules can depend on predictive scores from standard predictive analytic models imported into Oracle Real-Time Decisions or adaptive analytic models managed by Oracle Real-Time Decisions. A simple point-and-click, template-driven approach is used to define business rules, and predictive analytic models can be imported from R as scripts (that can be executed by Oracle R Enterprise in Oracle Database) or executed through stored procedures or web service calls. New business rules and predictive analytic models can be defined and deployed to a running system and any adaptive analytic models in use will be automatically updated during use based on the learning engine's results.

Figure 8: Offer Eligibility Rule



Source: Oracle

Performance Measure-Driven Decisions

Oracle Real-Time Decisions uses performance measures to drive decisions and to arbitrate between the eligible actions or offers available each time. The user can define a set of performance goals for each decision. These can be simple

properties of an action such as a priority or value, a predictive analytic model such as retention risk or acceptance likelihood, or any combination of these.

Deterministic scoring rules (an additive scorecard) or arbitrarily complex formulae can be defined if necessary, allowing a range of performance goals to be specified for the decision. The weight of each performance goal then can be set for a decision, defining how the engine should trade off the impact of a decision against each performance goal. A single set of weights can be defined or, as shown in Figure 9, separate weights can be defined for each of a series of segments defined using rules-based filters.

Figure 9: Performance Goals by Segment



Source: Oracle

It should be noted that business users can add segments and change these weights in live Oracle Real-Time Decisions decision services without any IT involvement.

Learning Engine and Learning Graphs

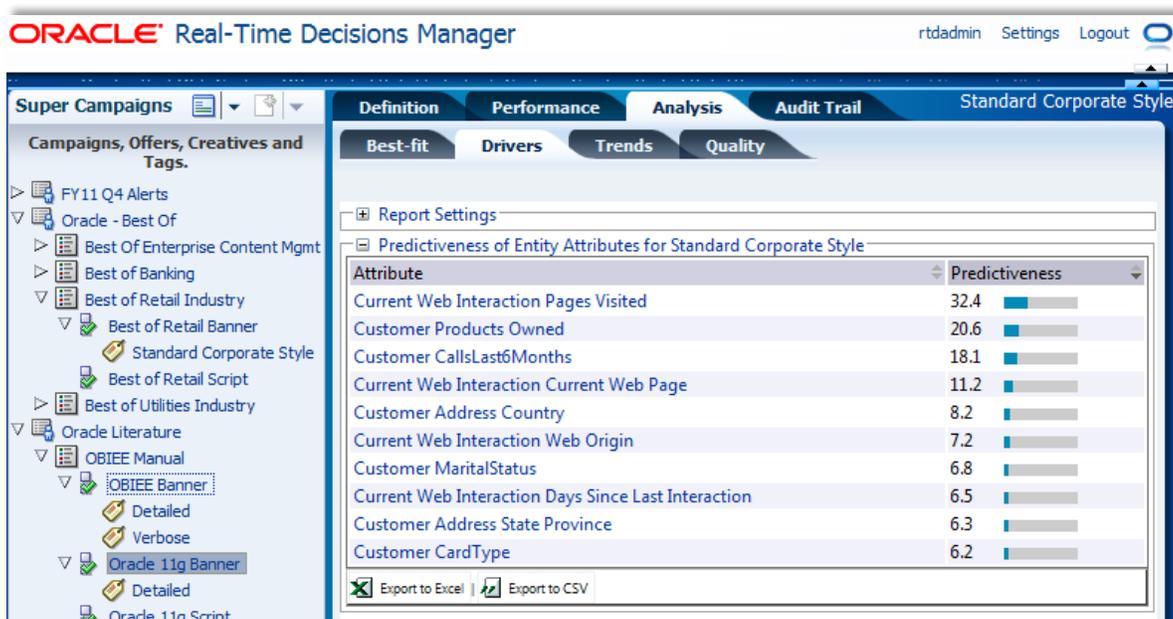
The Oracle Real-Time Decisions learning engine is usable as a standalone learning application or it can be integrated with the decision service engine **Error!**

Reference source not found. to drive adaptive analytic models. The learning engine's core capability is its ability to derive correlations from the data fed into it.

The learning engine is configured to process business events defined by the user and fed into the engine using a standard API. The machine learning algorithms at the heart of the engine then derive correlations between these business events and other data elements—that a call was transferred, a successful sale was made, and so on. When deployed standalone, the learning engine becomes a self-service tool for business users to understand the correlations between customer attributes or other behavior in their systems and the business results they want. Even if they are not, for instance, ready to automate a decision, business users can deploy the learning engine to learn what works and what does not. If they are using the high-performance decision service engine and feeding results from it into the learning engine, then Oracle Real-Time Decisions provides a predefined link to use the correlations identified to drive adaptive analytic models in the engine.

For instance, Figure 10 shows that the most relevant attributes for predicting who will click on a particular banner—pages visited, products owned, etc. These are built automatically by the engine and can consider demographic data, transactional information, and information about the current interaction.

Figure 10: Predictive Drivers in the Learning Engine



Source: Oracle

These insights can be combined with traditional business intelligence tools to, for instance, find sets of customers that match the predictive drivers. In this way the

learning engine is essentially doing automated data discovery and supporting very targeted data analysis.

The learning engine is a high-performance service that supports multiple CPUs/cores, enabling it to process millions of learning records a day—Oracle has tested up to 58 million learning records in a day.

Learning graphs are increasingly used to show knowledge as a set of related concepts and then analyze the direct and indirect connections between these concepts. When using Oracle Real-Time Decisions for marketing, for instance, the user can define a graph that connects campaigns and the offers within them, channels, and the slot types and placements available; creative for an offer that is suitable for a slot type; tags that categorize these pieces of content; and the specific slots available. The learning graph is configured to match the relevant nomenclature and data models. When an event occurs—a user clicks on an ad on a website, for example—Oracle Real-Time Decisions applies the learning from this event across the whole graph. The engine might therefore learn that the particular creative is effective for a certain kind of customer, that it is effective in that slot, that creative tagged a certain way is effective, or that the offer it represents was particularly appealing. The learning potential of the event is thus multiplied as the engine uses the graph to propagate what it learns across, up, and down the graph.

Oracle has found that the use of tags to categorize creative is particularly effective in the context of learning graphs. The fact that a piece of creative has been tagged as “humorous” or “green,” for instance, can be extremely predictive of the kind of customer to whom it will appeal. These tags can be defined by nontechnical users and deployed into running systems where they are immediately used for learning.

The use of learning graphs means that the learning engine turns one click or interaction into multiple learning opportunities, improving the precision of what it learns and improving the rate at which it can adapt models to make them more predictive.

Adaptive Analytic Models

While the decision service engine can use predictive analytic models defined outside of Oracle Real-Time Decisions, it also can use Oracle Real-Time Decisions' adaptive analytic models. In this mode, Oracle Real-Time Decisions builds a model based on what seems to predict the desired behavior no matter what it is. Oracle Real-Time Decisions uses automated versions of the usual data mining techniques to build these models as it integrates with the learning engine to adapt these models as events happen and new correlations are identified. The models will prioritize those choices most likely to meet the performance goals defined by offering both performance improvement and personalization without manual intervention. Overlapping time windows can be defined so that models recycle and the engine

supports seasonality. Partitioning allows multiple models to be managed for different customer segments.

As is usual with automated model development and tuning, there is not much incentive to limit the number of predictors, so Oracle Real-Time Decisions brings in whatever predictors it finds. Oracle Real-Time Decisions manages the performance of adaptive models by having a separate deployed model and updating it when a statistically significant threshold is passed or in a time-based way (say every minute). By reducing the time and cost to develop a model to nearly zero, Oracle Real-Time Decisions can use many hundreds or thousands of models and continually evaluate and improve their predictive power.

Decision Management Interface

The decision management interface provides an enterprise view of decisions and a collaborative environment for business users to manage their decisions. A thin client, role-based user interface supports a broad-based search, version control, audit trails of changes, and multiuser collaboration. Visualization and reporting are provided and can be extended using Oracle Business Intelligence Suite, Enterprise Edition.

Figure 11: Editing in the Decision Management Interface

The screenshot displays the Oracle Decision Management Interface. On the left, a tree view under 'Super Campaigns' shows various campaign categories, with 'Real Time Decisions (RTD)' expanded to show 'RTD Banner'. The main window is titled 'Definition' and has tabs for 'Performance' and 'Analysis'. Below the title bar are 'OK' and 'Cancel' buttons. The 'Attributes' tab is selected, showing a form with the following fields:

- * Approval Status: Draft (dropdown)
- * Code: rtd1 (text)
- Start Date: 8/24/2012 (calendar icon)
- End Date: 8/24/2013 (calendar icon)
- * Type: Cross-Sell (dropdown)
- Region: (None) (dropdown)
- Product: (text)
- Promotion: (text)
- Cost: 5 (text)
- Revenue: 10 (text)
- Marketing Priority: 10 (text)

A tooltip for 'Marketing Boost Score For Creative' is positioned over the Marketing Priority field.

Source: Oracle

Within the interface, the decision repository is displayed using a folder-based structure specific to the installation. Each user sees a different view of this, based on his or her role, and can get basic information about everything in the repository. The offers or actions defined for decisions are at the core of this, and each is managed with its supporting creative and eligibility rules. Multiple perspectives allow a focus on offers by campaign or by channel, for instance, and each change is recorded for a complete audit trail. Reporting and visualization support a wide range of decision performance measures and allow detailed analysis of the effectiveness of specific offers or creative to enable users to see which does best.

References

OSGi Alliance <http://www.osgi.org/>

Taylor, J. (2012). *Decision Management Systems Platform Technologies Report*. Palo Alto CA: Decision Management Solutions.

Taylor, J. (2012). *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics*. New York, NY: IBM Press.

Contact Us

If you have any questions about Decision Management Solutions or would like to discuss engaging us, we would love to hear from you. Email works best but feel free to use any of the methods below.

Email : info@decisionmanagementsolutions.com

Phone : +1 650 400-3029

Web : www.decisionmanagementsolutions.com

