Siena: a tool for modeling and executing artifact-centric business processes

PhD Seminar -- Università di Roma "La Sapienza"
Terry Heath - Senior Software Engineer
IBM T.J. Watson Research Center
December 15th 2009
Seminar Agenda

- **Artifact-Centric Approach**
  - Introduction to Business Entities

- **Comparison of Business Process Management Approaches**
  - *Process-Centric* approach using Hotel Scenario
  - *Artifact-Centric* approach using Hotel Scenario

- **Introduction to Siena**
  - Overview
  - Architecture

- **Siena Demo**
  - Review Hotel example

- **Siena Details**
  - Meta-Model
  - Services

- **Siena Examples**
  - Alessio and Patrizia
A Key Challenge in Business Process Management

(Many Stakeholders in an Enterprise)

Business Strategy
- “Be more green”
- “Use our differentiators”

Executive

Process Owner
Business Architect
Solution Designer

Business Goals
Business Architecture
Business Optimization

Bridge the Gap
between Business and IT

Speak in terms of
- “Functional Decomposition”
- “Business Components”
- “Workflow”
- “Activity-flow”
- “DB schema”

IT Architect
Systems Integrator

Customers
Employees
Partners
Resources

“Impedance Mismatch”

Basic Challenge: Today’s approach to BPM environments is fundamentally disjointed (Many disjoint Models)

If Guest is paying by AMEX
Then give 5% discount

Core BP model is based on activity flows

The data being manipulated is often an afterthought, not related to other conceptual models

Lack of coherence adds substantial complexity to an already very complex environment

One conceptual model for rules and policies

Another conceptual model for analytics and dashboards
Solution: Unified Business Construct

Business Entity: (e.g. Purchase Order)

Alignment of Models
Contextualized as a Business Entity
Coherence Achieved
“Business Entities”: data + process combined to form a new, “holistic” foundation for BPM

- **Business Entities are Unifying Business Constructs**
  - Provides a skeleton that cuts across the Business
    - e.g., **Guest Stay**
      - From CheckIn to CheckOut
      - Blending of Data, Rules, Process, Measurements in the context of a Guest Stay

- **Includes specification of both**
  - The *information model*, to hold relevant data about an artifact as it moves through the workflow, and
  - The *possible lifecycles* it might follow

- **Insight**: Gives business managers a unified, end-to-end view of their business operations
- **Communication**: Numerous stakeholders have a common basis for understanding
- **Actionable**: Natural mapping to organization & IT levels
# Brief comparison of BPM approaches

## Process-Centric Approach

- **Business Data is**
  - NOT the primary focus
  - Business data is merely an after thought

- **Process Steps are the main concern**
  - *What* do humans *do* in the business
  - *What* systems *need* to be integrated

## Artifact-Centric Approach

- **Business Data is**
  - The PRIMARY focus

- **Process steps occur in context**
  - The *“Business Entity”* needs *which* humans to do something to it.
  - The *“Business Entity”* needs to integrate with *what* certain systems.
Review of Hotel Scenario for comparison

- **Posting charges to the guest folio during a hotel stay**

- **Base scenario**
  - Guest checks in
  - Room charges are posted by the Night Audit process
  - Guest dines in the hotel restaurant
  - Guest checks out

- **Scenario evolution 1 – Handling of “lost” charges**
  - Guest has breakfast after checking out

- **Scenario evolution 2 – Handling of charges by “drop-ins”**
  - A non-guest dines at the hotel restaurant
Process-Centric approach for Hotel scenario
Check in- Checkout Process
(Process-Centric Approach)

Waiting for Checkout Request

Long Running Flow:
- Remains active until Guest checks out
- The Guest Stay information is hidden in the long running process instance data
Night Audit Process – Modeled as separate process (Process-Centric Approach)

- Multiple instances of the process created for each day for each guest
- No direct link between check in process and night audit process.
Combined check in- Checkout and Night Audit Process
(Process-Centric Approach)

Night audit process modeled as part of main flow to avoid multiple process instances being created.

Infinite loop for night audit

Waiting for Checkout Request
Point of Sale (POS) Process
(Process-Centric Approach)

POS modeled as separate process as this can be instantiated independently any number of times.
Summary
(Process-Centric Approach)

- Discrete/Disjoint Processes

- Data is an after thought

- Guest stay information lost in long running process instances

- Lots of additional coding needed to integrate to Databases and Services
Artifact-Centric Hotel Scenario

Hotel

Guest Stay
Folio

Guest
Check In
Check Out

Front Desk Clerk
Create
Settle

Create

Create

Paid Charges

Pended Charges

Audit Report

Get and Post Charges To Folio

Audit

Night Auditor

Non Guest
Purchase

Purchase

Cashier
Create
Create

© 2009 IBM Corporation
Entity-Centric approach of Hotel Scenario

- Identify key Business Entities
  - Guest Stay
    - States: Started, CheckedIn, RoomAssigned, CheckedOut
    - Information: Stay_ID, CheckInDate, CheckOutDate, Guest Name, Guest Profile, Guest Type, Room Rate, Room Preferences, Room Number Assigned, Folio Info
  - Guest Folio
    - States: Started, Active, Settled
    - Information: Folio_ID, Guest Name, Room Number Assigned
  - Charge
    - States: Start, Pended, Paid, Posted, Lost, Cancelled
    - Information: Charge_ID, Date Incurred, Charge Type, Room Number, Payment Type, ItemInfo(code, desc, qty, cost), TaxInfo(Tax Rate, Desc, TaxTotal)
  - Night Audit
    - States: Started, InProgress, Completed
    - Information: Audit_ID, Stay_ID, Folio_ID, GuestName, Room Number, DataAuditStarted, DateAuditCompleted, reconciledCharges(1..n)
Guest Stay Entity
(Artifact-Centric approach)

Lifecycle

Information Model

Guest Stay

- Started
- Checked In
- Checked Out

Guest ID
Date Checked In
Date Checked Out
Guest Name
Guest Profile
Room Rate
Room Type
Room Number Assigned

- Default info
- Guest info
- Room Info
- Folio Info
**Folio Entity**
(Artifact-Centric approach)

- **Lifecycle**
- **Information Model**

- **Guest Folio**
  - Started
  - Active
  - Settled

- **Default info**
- **Guest Info**
- **Room Info**
- **Charge Info**
Charge Entity
(Artifact-Centric approach)

Charge

Lifecycle

Information Model

- Started
- Pended
- Paid
- Canceled
- Posted
- Lost

Charge ID
Folio ID
Guest Name
Guest Type
Room Number
Item Code
Item Description
Quantity
Cost
Tax Rate
Tax Type
Tax Amount

Default info
Guest Info
Charge Info
Tax Info
Night Audit Entity
(Artifact-Centric approach)

Night Audit

Lifecycle

Information Model

- Started
- Room Charges Created
- Charges Reconciled
- Completed

Folio info

Guest Info

Reconciled Charge Info
Business Entity Lifecycles and Business Entity Interactions

Guest Stay

- Started
- CheckedIn
- CheckedOut

Create Folio

Guest Folio

- Started
- Active
- Settled

Add Charges

Night Audit

- Started
- Room Charges Created
- Charges Reconciled
- Completed

Begin night Audit

Settle Audit

Charge

- Started
- Pended
- Paid
- Lost
- Canceled

Create POS Charge

Get Pending Charges For Guest

Post Charge

Create Room Charges

© 2009 IBM Corporation
Some comparison points

- **Process-Centric Approach**
  - Process flows act as controllers
  - Data is an after thought
  - Some Operational data hidden in long running process data
  - Humans work on “blocked” tasks in long running flows
  - Performance of long running flows not desirable

- **Artifact-Centric Approach**
  - Entities act as controllers
  - Data is Core
    - Business Entities accessible in DB
  - All Operational data stored in Business Entities
    - Queryable, Trackable, Measurable
  - Humans work on Business Entities that are ready for their contribution
  - Performance of Entities (info, lifecycle, micro flows) considered acceptable
Business-Entities provide improved communication among stakeholders in the business

- **“Along” the artifact:**
  - People at “PENDED” can discuss meaningfully with people at “POSTED”
  - Can discuss attribute values produced, needed by different tasks

- **“Across variations”:**
  - Different regions can communicate using shared abstract model
    - *(Variation of Rules in lifecycles and Process Steps)*

- **“Up/down management chain”**
  - Artifact approach lends itself to more abstract / more detailed specifications

### Diagram:

- **Charge ID**
- **Folio ID**
- **Guest Name**
- **Guest Type**
- **Room Number**
- **Item Code**
- **Item Description**
- **Quantity**
- **Cost**
- **Tax Rate**
- **Tax Type**
- **Tax Amount**

- **Default info**
- **Guest Info**
- **Charge Info**
- **Tax Info**

- **States:**
  - Started
  - Pended
  - Paid
  - Canceled
  - Posted
  - Lost
Business Rules Constrain Access, Lifecycle, and Behavior

*Business rules define task details & variations*

- Rules define how lifecycles can be traversed
- Rules can also define how flows, and data can be manipulated
Introduction to Siena
(Light-weight Artifact-Centric Modeling and Execution tool)

- **Empower SME’s to easily Innovate new processes**
  - Tools and Runtime often too heavy and hard to understand
  - Innovators currently dependent on IT teams

- **Radical Simplification of Tools and Runtime**
  - **Siena Core Meta Model**
    - Describes semantics for all modeling constructs of Business Entities
  
  - **Model Management Engine**
    - **Web UI Modeling** tool for producing Business Entity models
  
  - **Model Execution Engine**
    - Default Execution UI for deploying and executing Business-Entity models
    - Directly executes Business Entity models
    - No coding necessary, No code generation
    - Small footprint can be easily hosted anywhere including laptop
Innovator’s toolkit for Business Process Modeling

( Democratization of Innovation -- Eric Von Hippel )

Manufacturer-centered innovation

Innovator-centered innovation
Radical Simplification of Tools and Runtime
(Supporting Business Process Management Applications using Entity Centric Modeling)

Dreaming of being lighter and more Agile

Reduced set of BPM abstractions to define and create BPM solutions.

Tooling Stack

10 Gigs Download/Disk Space, 1-2 Days successful installation, At least 2 Gigs Memory
Siena Architecture Diagram

**Modeling Clients**
- Artifact Model Editor

**Execution Clients**
- Custom UI
- Default UI (dynamically rendered)
- UI Toolkit (Artifact-centric widgets & helpers)
- Google Web Toolkit

**Model Execution Engine**
- Artifact Model Management
- Data Access
- Behavior Services (lifecycle, flow, GSM engines)
- Access Control
- Rule Evaluation (JEXL, OCL)
- Monitoring & Analytics
- Extension Points

**External Services**
- REST & WSDL
- Apache Tomcat
- WAS
- Geronimo

**Artifact Instances**
- XSD
- XML

**Siena Models**
- Artifact Model Management
- Data Access
- Behavior Services (lifecycle, flow, GSM engines)
- Access Control
- Rule Evaluation (JEXL, OCL)
- Monitoring & Analytics
- Extension Points
What is a Business Entity: a Unified Business Construct

Business Entity: (e.g. Purchase Order)

Alignment of Models
Contextualized as a Business Entity
Coherence Achieved
Review: What is a Business Entity? (Deeper Inspection)

- It’s a Unifying Business Construct:
  - **Structured by**
    - Core business data models
    - Artifact lifecycles
      - State machines (Siena)
      - Declarative stages (Project ArtiFact™)

  - **Providing Services**
    - Transition services
    - Data services
    - Flow services

  - **Protected by Access Control**
    - Users and Roles
    - Entitlements
      - Data access rights
      - Service access rights

  - **Constrained By Business Rules:**
    - Data, services, lifecycles, flows, behavior

  - **Has measurable features**
    - Data, lifecycles, flows, tasks
Siena Entity *(The Core of Siena)*

Services Offered:
- Business Service
- Business Roles
- Business Rules
- Security
- Process Modeling

Services Consumed:
- Service Integration
- Business Metrics
- UI
What makes up a Siena Application?

- **Siena Meta-Model as XSD**
- **Artifact Schema as XML**
  - **XSD**
  - **XSD**
  - **XSD**
- **Data Models**
- **Data Services**
- **Organization**
- **Metrics**
- **Flow Services**
- **External Services**
- **Siena (Application) XML**
  - **External Services**
  - **Artifact**
  - **Information Model**
  - **Lifecycle Model**
  - **...**
- **Deployment Zip File**
- **Modeling Clients**
- **Direct Deployment and Execution of Models**

- **Service I/O XSDs**
- **Artifact XSDs**
Siena Schema (Meta-Model)

Composite Application

Siena Meta-Model as XSD

Modeling Clients

- Produce Model Instances
- Constrained by Meta-Model

Uses Meta-Model to Execute Model Instances
Generate Java from Siena Meta-Model

- **Siena Meta-Model as XSD**
  - XSD 2 Java
  - Generate

- **Java Classes**
  - XSD 2 Java
  - Generate

- **Artifact Model Management**
  - Data Access
  - Behavior Services (lifecycle, flow, GSM engines)
  - Access Control
  - Rule Evaluation (JEXL, OCL)
  - Monitoring & Analytics
  - Extension Points

- **Model Execution Engine**
  - Apache Tomcat
  - Geronimo
  - REST & WSDL

- **Modeling Clients**
  - Java Classes used to create Siena Model Instances
  - Java Classes used to execute Siena Model Instances
Model Driven Architecture

Model is XML and XSDs

- Siena (Application) XML
  - External Services
  - Artifact
    - Information Model
    - Lifecycle Model
    - ...

- Service I/O XSDs
- Artifact XSDs
Traditional Approach: *Generate Model Into Code*

**Innovator**

Now what do I do with all this stuff?

- JavaScript
- HTML
- WSDL
- XML
- EJBs
- Servlets
- ACL
- BPEL
- XSD
- DDL
- SQL
- JAVA

The Siena Model
Siena Approach: *Direct Deploy and Execute Models*

**Innovator**

So Simple!!!!

I can Innovate and Test Immediately!

**The Siena Model**

Direct Deployment and Execution of Models

- Artifact Model Management
- Data Access
- Behavior Services (Lifecycle, flow, GSM engines)
- Access Control
- Rule Evaluation (JEXL, OCL)
- Monitoring & Analytics
- Extension Points

Apache Tomcat WAS Geronimo

Model Execution Engine
Platform Independent Entity-Centric Model
(Can be used to Reason against)

At Design time look for:
• Life cycle deadlocks
• Flow deadlocks
• Artifact interaction deadlocks

At Runtime time look for:
• Instance level forecasting of potential problems
Siena Demo: Example of an Entity-Centric Solution

- Review Hotel Design
- Run Hotel Design
Demo Questions and Answers
Procurement

- Purchase Order
- LineItem
- Shipment
Insurance
Banking

- Account
- Deposit
- Withdrawal
- Check
- Loan
Everything is a service in Siena

- Data Access
- Lifecycle transitions
- Flows
- External Services
Services

- **Service Definitions**
  - Transition Services
  - Flow Services
  - Data Access Services
  - External Services

- **Service Invocations**
  - From Tasks
  - From States

**REST & WSDL Services**

**Transitions**

**Flow**

**Data**

**External Services**
Registering External Services

- **REST**
  - Specify Service End Point URI
  - Import XSD

- **WSDL**
  - Point to Remote WSDL File
  - Import WSDL

Diagram:
- REST service Schema
- Artifact Model Editor
- Import
- WSDL File
- Record new External Service into Model
- Artifact Models
Binding and Mapping External Services

- **REST**
  - Specify Service End Point URI
  - Import XSD

- **WSDL**
  - Point to Remote WSDL File
  - Import WSDL

**Diagram**

- **REST End Point**
- **SOA End Point**
- **External Services**
- **Data**
- **Flow**
- **Input Mapping**
- **Output Mapping**
- **Input Mapping**
- **Output Mapping**

**Artifacts**
- Artifact Model Editor
- SmartGWT
- The Artifact Instance Data

**Tools**
- IBM DB2
- Apache Derby

© 2009 IBM Corporation
Business Entities give context for Service Invocations
External Service Integration (REST and WSDL)

Guest Stay

Get Guest Geo Info

External Services

Get Guest Geo Info

Yahoo GeoCode

Charge

Posting: Started Pended Paid Lost Canceled

Update Account Receivables

Large Collection of Externals Services Contextualized by Business Entities
Determine Entity Interactions

- Direct link between check in and night audit.
Basic Flow Patterns

Business Entity

Created

InProgress

Completed
Advance Flow Pattern

Business Entity

- One Flow can invoke multiple transitions based on Flow outcome
- Each Transition can in turn have Independent Transition Flows as well
Basic Data Access Patterns

Business Entity

- Data Access Service Always Available
- State Specific Data Access Service

Created

InProgress

Completed
Future Artifact-Centric work

- Further Siena Web Tooling Features
- Optimize Siena Engine
- Project ArtiFact™
  - Declarative Approach (no wires or transitions)
    - Guards, Stages and Milestones
What is Project ArtiFact™ GSM

- **Hierarchical Units of Work**
  - Units of Work (*Stages*)
    - Launch by Conditions (*Guards*)
    - Completed by Expressions (*Milestones*)
Declarative Artifact-Centric as a unifying basis for future BPM

Rules
If high priority customer and paying by AMEX permit ship before payment

Analytics
Rules in vocabulary of the artifacts
Analytics can use artifact info models directly

Declarative Artifacts
Theory of optimization

Biz design
Data Modeling
Process Modeling

Declarative Artifacts might play the role for BPM that Relational Model played for Database Mgmt
Hierarchical Stages (Units of Work) with contextual Rules

A stage focuses on a natural, small cluster of related rules
More on the “guarded” style for lifecycle specs

- **Can put a variety of rules / conditions into the “guards”, e.g.,**
  - Can only enter Customer login stage once
  - If you change Cart you must revisit Shipping Pref
  - Cannot enter Payment Pref until either you are logged in or put stuff in Cart

- **Can vary the guards based on region, customer category, etc.**
  - Variation at any level of hierarchy

Flow charts and state machines useful in some contexts; “Ad hoc” style useful in other contexts
Client Toolkit: Siena API Façade

(RESTful Siena)

- **SienaServiceClient** API
  - Generic RESTful API to call Siena
  - Support for XML Input, XML Output, JSON Input, JSON Output
  - API:
    - `SienaServiceClient sienaClient = new SienaServiceClient();`
      - Restful Service Façade
    - `sienaClient.invokeXml();`
      - Invoke Services (flow services, data services, transition services)
    - `sienaClient.retrieveListXml();`
      - Retrieve Artifact Instance Lists
    - `sienaClient.retrieveXml();`
      - Retrieve a single Artifact Instance
    - `sienaClient.saveXml();`
      - Save a single Artifact Instance
Inspect Code: using eclipse

```java
static String appName = "ClientTest";
static String dataItemId = "MyArtifact";
static String serviceId = "MyArtifact-ANY-to-Created-0Transition";
static String inProgressServiceId = "MyArtifact-Created-to-InProgress-0Transition";

String ArtifactXML = "<MyArtifact
    attribute1='Via Labicana' attribute2='Roma' attribute3='Terry' attribute4='Heath'>
</MyArtifact>");

// Make a New Client
SienaServiceClient client = new SienaServiceClient("http://localhost:8080/SienaWeb", "defaultAdmin", null);

// Invoke Any to Created
String invokeXmlOutputMessage = client.invokeXml(appName, serviceId, URLEncoder.encode(ArtifactXML));

// Extract newly created ID from new Instance
String id = parseIDFromXML(invokeXmlOutputMessage);

// Invoke Created to InProgress
String inProgInputXML = "<MyArtifact ID='" + id + '" />";
String inProgOutputXML = client.invokeXml(appName, inProgressServiceId, URLEncoder.encode(inProgInputXML));
```
Run Example Client from Eclipse

- Jump to IDE
- RUN
Client Toolkit: Siena API Façade

(RESTful Siena)

- **WSDL Siena**
  - **Solution Specific WSDL files**
    - Generated into deployed solution
    - Use your favorite IDE to bind to WSDL files and invoke
    - Generate JavaProxy Web Service Client
    - Begin to Invoke WSDL Operations onto the Solution Specific Artifacts
Let's now inspect the model using an XML Editor.
Explore the Development Environment in Eclipse
Artifact Relationship Patterns

- **(One to One)**
  - Insurance Claim can related to 1 Fraud

- **(One to Many)**
  - *Parent / Child*
    - Purchase Order
      - Line Items

- **(Many To Many)**
  - *Purchase Order*
    - Each Purchase Order can be delivered in 1 or more shipments
  - *Shipment*
    - Each shipment can contain Line Items from different Purchase Orders
Other Siena Examples

- More Siena Examples (Patrizia and Alessio)
  - Radio Sapienza
  - Relay Race
  - Color Bricks
  - Clinical
Radio Sapienza Overview (Patrizia and Alessio)

Advertisers → Radio Sapienza

Music Companies

Music Listeners

????(subscription fee)
Relay Race Overview (Patrizia and Alessio)

- All Players Run at Random Speeds
- Siena Controls Relay Race Servlets/External Services
**CLINIC**

- Manages information about the examinations: data of patients, description of the reports, admissions to a ward.
- Manages the ambulances: external services to find destination address and to visualize the map.
COLOR BRICKS

- Plays a game respecting the constraints.
- The user can choose the number of cells and the dimension of the matrix.
- Purpose: Siena is able to manage a big amount of instances.
RADIO SAPIENZA ARTIFACTS

- PLAYLIST
- TRACK
- PLAYER
Information Model: **PLAYLIST**

- **Name** (string):
  - name of the playlist.

- **DurationTot** (long):
  - total duration of the playlist.

- **PlayerID** (long):
  - ID of the player that is playing the playlist.

- **TrackList** (TypeTrack):
  - list of tracks that compose the playlist.

- **TrackPlayed** (TypeTrack):
  - informations of the track that is actually played.
Information Model: **TRACK**

- **TempPlaylistID** (long):
  - ID of the playlist that is playing the track.

- **StartTime** (dateTime):
  - date and time in which the track is played

- **PauseTime** (dateTime)

- **ResumeTime** (dateTime)

- **RemainingDuration** (long):
  - remaining duration of the track after a “resume” action.

- **TrackInfo** (TypeTrack):
  - informations of the track.
Information Model: **PLAYER**

- **NumberTracks** (int):
  - Number of the tracks played.

- **PlaylistID** (long):
  - ID of the playlist that the player is playing.
EXTERNAL SERVICE: **BrowseAmp**

- **WinAmp** is a media player.
- It is controlled by the plugin “BrowseAmp”.
- **BrowseAmp offers a RESTful service**
  - Allows us to control WinAmp with simply URLs
    - Play
    - Stop
    - Pause
    - Resume
OTHER SERVICES: ODDCAST E ICECAST

- We use two free softwares in order to simulate a radio station on internet.
- Oddcast sends the parameters to WinAmp that it is playing on the port 8002.
- IceCast broadcasts on internet the stream of data that arrives on that port.
- The users can listen the playlist using the link: [http://151.100.59.92:8002/Radiosapienza.m3u](http://151.100.59.92:8002/Radiosapienza.m3u) (address of the server in which all the applications run).
State Diagram: TRACK
State Diagram: PLAYER
RELAYRACE: ARTIFACTS

- PLAYER
- TEAM
- RACE
Information Model: PLAYER

- **Name** (String):
  - Player’s name.

- **Speed** (Float):
  - Player’s velocity.

- **Time** (Long):
  - Space / Speed (m/s).

- **TeamID** (Long):
  - ID of the player’s team.
Information Model: TEAM

- **Name** (String):
  – Name of the Team.

- **TotalTime** (Long):
  – Time spent to finish the race.

- **RaceID** (Long):
  – ID of the race in which the team is playing.

- **PlayerList** (PlayerType):
  – List of players of the team.
Information Model: **RACE**

- **Name** *(String):*
  - Name of the Race.

- **TeamList** *(TeamType):*
  - List of team that takes part in the race.

- **Results** *(Results):*
  - Arrival ranking of the teams at the end of the race.
EXTERNAL SERVICES

- We use Java JSP and Java Servlet in order to implement external services useful for our application.
- Siena manages the race, controls the competition and sends to the JSP pages all the attributes required.
- The external services are:

  - **RaceService**: 
  - **ShowRace**: 
    > [http://localhost:8080/OpenURL/Open](http://localhost:8080/OpenURL/Open) opens a window that shows the race.
  - **Winner**: 
    > opens a window with all the informations about the winner.
State Diagram: PLAYER

CREATE

TRAINING

STOP

RUNNING

ON

COMPETITION

ASSIGN TEAM

TO TRAINING

STOP RUNNING

TO RUNNING
State Diagram: TEAM

- CREATED
- READY
- STOP
- PLAYING

TO READY
TO CREATED
TO PLAYING
RACE FINISHED
RACE SERVICE
INVOKE
INVOKE
INVOKE
INVOKE

WINNER
CONTROL
WINNER

Relay Race
State Diagram: RACE

CREATED → SHOW → START → STOP

- ADD TEAM: CREATED → SHOW
- READY TO START: SHOW → START
- IN RACE: START
- PLAY AGAIN: START → RUN AGAIN → START
- RUN AGAIN: STOP → START
- RESET: SHOW → RESET RACE

Actions:
- SHOW RACE
- RACE SERVICE

Relay Race