Enabling OODA Loop with Information Technology

Eric S. Chan, Dieter Gawlick, Adel Ghoneimy, Zhen Hua Liu
Safe Harbor Statement

The following is intended to outline our general research or product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Pre Information Technology Process: OODA Loop

OODA Loop

• Observe, Orient, Decide, and Act (OODA) Loop
  • Observe the entities and environment,
  • Orient the participant to the observations, by cultural tradition, generic heritage, previous experience, analysis and synthesis, new information
  • Decide on the directives based on the hypotheses that best explains the observations, and
  • Act on the directives to interact with the entities and environment, to test the hypothesis

• The importance of the Loop and Provenance is substantiated by fighter pilots
  • (loop) rapid iteration of the loop to get inside the adversary’s OODA loop
  • (provenance) regular debriefing of the pilots after the missions to propagate the effective technique
FIHD, CARE Loop (IT version of OODA Loop)

FIHD = Facts, Information, Hypotheses, and Directives

CARE = Classification, Assessment, Resolution, and Enactment
Knowledge Intensive Database System 1

- Normalizes the data and knowledge
  - four categories of data (fact, information, hypothesis, directive)
  - four categories of knowledge (classification, assessment, resolution, enactment) that transforms the data
    - classification knowledge transforms fact to information
    - assessment knowledge transforms information to hypothesis
    - resolution knowledge transforms hypothesis to directive
    - enactment knowledge transforms directive to fact

- Normalizes the application structure into OODA, CARE, FIHD loops
  - classify the quantitative facts to derive the qualitative information,
  - assess the information to infer the hypotheses,
  - resolve the hypotheses to formulate the directives,
  - enact the directives to collect new facts, and
  - repeat the loop
Knowledge Intensive Database System 2

- Represents the entity model of the world
  - manage entity model in a multi-temporal database
  - concepts hierarchy and concepts lattice
  - OLAP operations in multidimensional data cubes

- Propels a faster iteration of the OODA loops in real-time
  - process management engine
    - continuously interacts with the environment to assess and adapt to the changes
  - manual or semi-automatic human interaction, tacit knowledge profiling
    - interaction is more powerful than algorithms
  - automatic agents, knowledge representation, machine learning
Knowledge Intensive Database System 3

- Materializes the OODA, CARE, FIHD loops in the data for provenance of the data and knowledge evolution
  - involves multiple iterations of the CARE loop
    - substantiated by bug database and customer service request lifecycles
  - annotates the CARE and FIHD loops in data
    - what are the fact, information, hypothesis, and directive in the problem report?
  - provenance of data and knowledge evolution

- Enables the development of evolutionary applications
  - knowledge is application
    - capture knowledge as much as possible declaratively and not in procedural code
  - data and knowledge are intertwined
    - desired application behavior evolves from the convergence of data and knowledge
      - when knowledge changes, data is re-analyzed
      - when data changes, knowledge is re-characterized
  - application development framework
    - graphical programming
    - meta-programming
KIDS Ontology

\[
KIDS = (\text{Actor, Entity, CARE, Metadata, Reification, Profile})
\]

\[
CARE = (\text{Data, Knowledge})
\]

\[
\text{Data} = (\text{Fact, Information, Hypothesis, Directive})
\]
\[
\text{Knowledge} = (\text{Classification, Assessment, Resolution, Enactment})
\]
Fact = (Entity × FSD^n) u (Entity × Feature^n)
Information = Entity × Feature^n × ValidTime × FigureOfMerit
Hypothesis = Entity × Feature^n × ValidTime × FigureOfMerit
Directive = Entity × Feature^n × ValidTime × FigureOfMerit

FSD = Value^n × ValidTime × FSDType
Feature = Value × ValidTime × FeatureType

ValidTime = [DateTime, DateTime u {∞, NA})
KIDS Knowledge Categories

\[
\text{Classification} = \{ f \mid f : \text{Fact} \rightarrow \text{Information} \}
\]
\[
\text{Assessment} = \{ f \mid f : \text{Information} \rightarrow \text{Hypothesis} \}
\]
\[
\text{Resolution} = \{ f \mid f : \text{Hypothesis} \rightarrow \text{Directive} \}
\]
\[
\text{Enactment} = \{ f \mid f : \text{Directive} \rightarrow \text{Fact} \}
\]
\[
\text{SymptomResolution} = \{ f \mid f : \text{Information} \rightarrow \text{Directive} \}
\]

\[
Kfun = \text{Classification} \cup \text{Assessment} \cup \text{Resolution} \cup \text{Enactment} \\
\cup \text{SymptomResolution}
\]
KIDS Tacit Knowledge Profile

Profile = (ActorProfile, KnowledgeProfile, Personalization)

ActorProfile : Actor → Entity × Feature^n × ValidTime × FoM
KnowledgeProfile : Kfun → Entity × Feature^n × ValidTime × FoM

Personalization : Kfun × Actor → Kfun

Personalization(Kfun, Actor)≡ curry(Kfun)(ActorProfile(Actor))
KIDS Meta-Program

\[
\text{MetaData} = (FSDType, \text{FeatureType}, \text{Influence})
\]

\[
FSDType = \text{Name} \times \text{Encoding} \times \text{Language}
\]

\[
\text{FeatureType} = \text{Name} \times \text{Type}
\]

\[
\text{Influence} = (\text{Input}, \text{Output})
\]

\[
\text{Input} = DType \times Kfun
\]

\[
\text{Output} = Kfun \times DType
\]

\[
\text{DType} = FSDType \cup \text{FeatureType}
\]
KIDS Reification Provenance

Reification = (CARE-Loop, Classified, Assessed, Resolved, Enacted)

CARE-Loop = (Classified × Assessed × Resolved × Enacted)^n

Classified = Fact × Classification × Information × Actor × TxnTime
Assessed = Information × Assessment × Hypothesis × Actor × TxnTime
Resolved = Hypothesis × Resolution × Directive × Actor × TxnTime
Enacted = Directive × Enactment × Fact × Actor × TxnTime

TxnTime = [DateTime, DateTime ∪ {∞, NA}]
KIDS Normalizes the Knowledge Representations 1

- Classification knowledge - deductive reasoning
  - Support Vector Machines
  - Naïve Bayesian Network
  - Neural Network
  - Clustering, Association Rules, Decision Trees
  - Multivariate State Estimation Technique (MSET)

- Assessment knowledge - abductive reasoning
  - Bayesian Belief Network
  - Least-Squares Optimization or Regression of solutions for inverse problems

- Resolution knowledge – decision theory
  - Influence Diagrams (Bayesian Belief Network with decision nodes)
  - Dempster-Shafer theory
  - Decision Trees
  - Prognosis of Remaining Useful Life (RUL)

- Enactment knowledge - control structures
  - scripts, plans, schedules, GOLOG, BPEL, BPMN

February, 2014
Enabling OODA Loop with Information Technology
KIDS Normalizes the Knowledge Representations 2

• Situation Knowledge - situation theory
  • Entity Model
    • data cubes
    • dimensions
    • concepts hierarchy and concepts lattice
  • Situation Theory Ontology
    • situation calculus
  • Provenance
    • reifications
    • bi-temporal database

• Tacit Knowledge
  • Knowledge profiles
  • Preferences
  • Personalization
KIDS Information Fusion

Information Fusion of
• Observation
• Objective
• Prediction, and
• Simulation

Information Fusion of
• Dom0
• OracleVM
• Enterprise Scheduler Server
**KIDS Propels the Process Interactions**

**Influence table enables dynamic scheduling of processes**

<table>
<thead>
<tr>
<th>Input</th>
<th>Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS boot log (FSD Fact)</td>
<td>OVM Crash Watcher (Classification)</td>
</tr>
<tr>
<td>OS Watcher Log (FSD Fact)</td>
<td>OVM Memory Watcher (Classification)</td>
</tr>
<tr>
<td>Enterprise Scheduler Service Log (FSD Fact)</td>
<td>ESS Process Watcher (Classification)</td>
</tr>
<tr>
<td>OVM Memory Spike (Information)</td>
<td>OVM Memory Diagnosis (Assessment)</td>
</tr>
<tr>
<td>ESS Process Spike (Information)</td>
<td>OVM Memory Diagnosis (Assessment)</td>
</tr>
<tr>
<td>OVM OutOfMemory Prediction (Information)</td>
<td>OVM Memory Diagnosis (Assessment)</td>
</tr>
<tr>
<td>OVM Crash (Information)</td>
<td>OVM Crash Diagnosis (Assessment)</td>
</tr>
<tr>
<td>Dom0 has Elastic Memory (Information)</td>
<td>Elastic Memory Advisor (Resolution)</td>
</tr>
<tr>
<td>Needs Elastic Memory (Hypothesis)</td>
<td>Elastic Memory Advisor (Resolution)</td>
</tr>
<tr>
<td>Unlock Memory in Dom0 to DomU (Directive)</td>
<td>Dom0 Memory Manager (Enactment)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVM Crash Watcher (Classification)</td>
<td>OVM Crash (Information)</td>
</tr>
<tr>
<td>OVM Memory Watcher (Classification)</td>
<td>OVM OutOfMemory Prediction (Information)</td>
</tr>
<tr>
<td>OVM Memory Watcher (Classification)</td>
<td>OVM Memory Spike (Information)</td>
</tr>
<tr>
<td>ESS Process Watcher (Classification)</td>
<td>ESS Process Spike (Information)</td>
</tr>
<tr>
<td>OVM Memory Diagnosis (Assessment)</td>
<td>Needs Elastic Memory (Hypothesis)</td>
</tr>
<tr>
<td>OVM Crash Diagnosis (Assessment)</td>
<td>OVM OutOfMemory Crash (Hypothesis)</td>
</tr>
<tr>
<td>Elastic Memory Advisor (Resolution)</td>
<td>Unlock Memory in Dom0 to DomU (Directive)</td>
</tr>
<tr>
<td>Dom0 Memory Manager (Enactment)</td>
<td>Dom0 and DomU Memory (Feature Fact)</td>
</tr>
</tbody>
</table>
KIDS Entity Model in a Temporal Database

- Situation Calculus is dynamic First Order Logic
  - can leverage temporal database

- Entity model of software and hardware components in Oracle Cloud is dynamic
  - new software releases
  - patches for bug fixes
  - hardware upgrades
  - capacity scale out
  - dynamic resource management

- Monitoring anomaly to avert SLA violation
  - time-series model of loads, system changes, and system responses
    - load distribution – e.g. Poisson arrival process
    - seasonal trends – daily, weekly, monthly cycles
    - configuration changes
    - software patches
    - hardware upgrades
Entity Model Enables BIG Data Analytics

- Entity Extraction from BIG Log Data

- Concepts hierarchy and lattice of Fusion Application
  - pillar dimension
  - pod dimension
  - resource dimensions

- OLAP operations in multidimensional data cubes
  - Roll-up
  - Drill-down
  - Slice and Dice
  - Pivot
  - Drill-across, Drill-through
Entity Extraction from BIG Log Data

Example: Entity Identification in WebLogic server.log

Log file metadata includes Pod, Domain, Server information.

<112wNGH4Klp96313VJJOB8911126u001a11> <1391393882108> <BEA-101017> 
<[ServletContext@664091965][]> Root cause of ServletException.

javax.io.IOException: javax.el.ELException: oracle.jbo.ReadOnlyAttrException: JBO-27008: Attribute PersonId in view object WorkerList1 cannot be set

a12345.oraclecloud.com identifies the OVM
TalentManagementServer_1 identifies the JVM
HcmTalentApp identifies the application
hcmTalent identifies the web module
/hcmTalent identifies the url
112wNGH4Klp96313VJJOB8911126u001a11 identifies the Execution Context ID
BEA-101017 identifies the source of the exception
JBO-27008 identifies the cause of the exception
Entity Model of Fusion Application Pillar

Concepts Hierarchy

- Farm
  - partOf
  - Pillar
    - partOf
    - Domain
      - partOf
    - Server
      - partOf
    - Application
      - partOf
    - WebModule
      - partOf
CREATE DIMENSION pillar_dim

LEVEL webModule IS (pillar.webModule_name)
LEVEL application IS (pillar.application_name)
LEVEL server IS (pillar.server_name)
LEVEL domain IS (pillar.domain_name)
LEVEL pillar IS (pillar.pillar_name)
LEVEL farm IS (pillar.farm_name)

HIERARCHY pillar_rollup (webModule CHILD OF
application CHILD OF
server CHILD OF
domain CHILD OF
pillar CHILD OF
farm);
Entity Model of Fusion Application Pods

Concepts Lattice

February, 2014
Enabling OODA Loop with Information Technology
Slide 24
CREATE DIMENSION pod_dim
    LEVEL webModule IS (pillar.webModule_name)
    LEVEL application IS (pillar.application_name)
    LEVEL server IS (pillar.server_name)
    LEVEL domain IS (pillar.domain_name)
    LEVEL javaVM IS (pod.javaVM_name)
    LEVEL oracleVM IS (pod.oracleVM_name)
    LEVEL cluster IS (pod.cluster_name)
    LEVEL pod IS (pod.pod_name)
    LEVEL dataCenter IS (resource.dataCenter_name)

HIERARCHY pod_rollup (webModule CHILD OF application CHILD OF server CHILD OF domain CHILD OF pod CHILD OF dataCenter)
HIERARCHY cluster_rollup (webModule CHILD OF application CHILD OF javaVM CHILD OF cluster CHILD OF domain CHILD OF pod CHILD OF dataCenter)
HIERARCHY vm_rollup (webModule CHILD OF application CHILD OF javaVM CHILD OF oracleVM CHILD OF pod CHILD OF dataCenter)

JOIN KEY (pod.domain_name) REFERENCES domain
JOIN KEY (pod.dataCenter_name) REFERENCES dataCenter;
Entity Model of Virtual and Physical Resources

Concepts Lattice
Entity Model of Physical Machines

- **ServerNode**
  - `name`: String
  - `displayName`: String
  - `numberOfSockets`: Integer

- **Processor**
  - `name`: String
  - `displayName`: String
  - `numberOfCores`: Integer
  - `numberOfThreadsPerCore`: Integer

- **MachineModelEnum**
  - `Sun_Fire_X4170_M2`
  - `Sun_Fire_X4270_M2`
  - `Sun_Server_X4_2L`
  - `Sun_Server_X4_2L_V2`

- **ProcessorModelEnum**
  - `Intel_Xeon_E5_2650`
  - `Intel_Xeon_E5_2650_V2`
  - `Intel_Xeon_E5_2690`
  - `Intel_Xeon_E5_2697`
  - `Intel_Xeon_E5_2697_V2`
  - `Intel_Xeon_E7_8870`
  - `Intel_Xeon_X5675`
  - `Intel_Xeon_L5640`
CREATE DIMENSION compute_resource_dim

LEVEL javaVM IS (pod.javaVM_name)
LEVEL oracleVM IS (pod.oracleVM_name)
LEVEL dom0 IS (compute_resource.dom0_name)
LEVEL computeNode IS (compute_resource.computeNode_name)
LEVEL exalogicRack IS (compute_resource.exalogicRack_name)
LEVEL processor IS (processor.processor_name)
LEVEL dataCenter IS (resource.dataCenter_name)

HIERARCHY compute_resource_rollup ( 
  javaVM CHILD OF 
  oracleVM CHILD OF 
  dom0 CHILD OF 
  computeNode CHILD OF 
  exalogicRack CHILD OF 
  dataCenter)

ATTRIBUTE computeNode DETERMINES
  (computeNode_name, serverModel, ramCapacity, ramCapacityUnit,
  numberOfSockets)

ATTRIBUTE processor DETERMINES
  (processor_name, processorModel, cpuFrequency, cpuFrequencyUnit,
  numberOfCores, numberOfThreads)

JOIN KEY (compute_resource.oracleVM_name) REFERENCES oracleVM
JOIN KEY (compute_resource.processor_name) REFERENCES processor
JOIN KEY (compute_resource.dataCenter_name) REFERENCES dataCenter
CREATE DIMENSION database_resource_dim

LEVEL database IS (database_resource.database_name)
LEVEL databaseNode IS (database_resource.databaseNode_name)
LEVEL exadataRack IS (database_resource.exadataRack_name)
LEVEL processor IS (processor.processor_name)
LEVEL dataCenter IS (resource.dataCenter_name)

HIERARCHY compute_resource_rollup
  database CHILD OF
  databaseNode CHILD OF
  exadataRack CHILD OF
  dataCenter

ATTRIBUTE databaseNode DETERMINES
  (databaseNode_name, serverModel, ramCapacity, ramCapacityUnit, numberOfSockets)

ATTRIBUTE processor DETERMINES
  (processor_name, processorModel, cpuFrequency, cpuFrequencyUnit, numberOfCores, numberOfThreads)

JOIN KEY (database_resource.processor_name) REFERENCES processor
JOIN KEY (database_resource.dataCenter_name) REFERENCES dataCenter
KIDS Materializes OODA Loops in Data

- Intelligent Behavior = Data + Knowledge + Process
- Captured by an invariant structure of data, knowledge, and process
  - Data: fact, information, hypothesis, directive
  - Knowledge: classification, assessment, resolution, enactment
  - Process: observe, orient, decide, act
- Annotate Log Data with CARE Data
- Materialize the OODA loops in Data for provenance
- Enable a faster OODA loop in real-time
Annotate the FIHD and CARE Data

Fact: server.log and node_manger.log
Information: OOM Exception
Directive: restart by node manager

Fact: node_manager.log
Information: High level drift
Directive: restart by operator

Fact: Heap Dump JVM23456.hprof
Information: Leak site HashMap XYZ.cache
Hypothesis: Memory Leak (Bug 11111)
Directive: Implement soft reference
Enactment: Apply Patch-12345

Fact: GC data
Information: High memory variance
Hypothesis: Need LRU in Cache (Bug 22222)
Directive: Soft ref on hash value, not HashMap
Enactment: Apply Patch-45678

Missing Data
Conclusion

- **KIDS Ontology**
  - leverages existing database, knowledge, and social interaction systems
  - tackles variety problem of BIG data
    - significant amount of data
    - significant amount of rapidly evolving knowledge
    - large scale state tracking for provenance
    - rich social interaction and collaboration