



Complex Event Processing

Design Specification



Empowering People with Information that *Moves!*

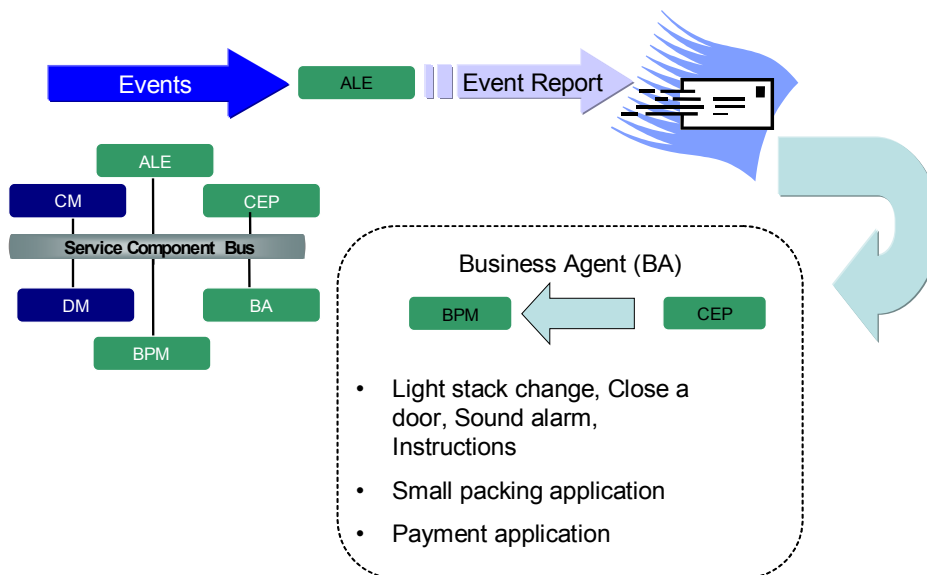
Tom Rose (tom.rose@i-konec.com)

Saturday, April 22, 2006

TABLE OF CONTENTS

MIDDLEWARE.....	3
Overview.....	3
Complex Event Processing	4
Overview.....	4
Rules Engine.....	5
Business Context Objects (BCO).....	5
Event Stream Working Set.....	6
PROCESS.....	6

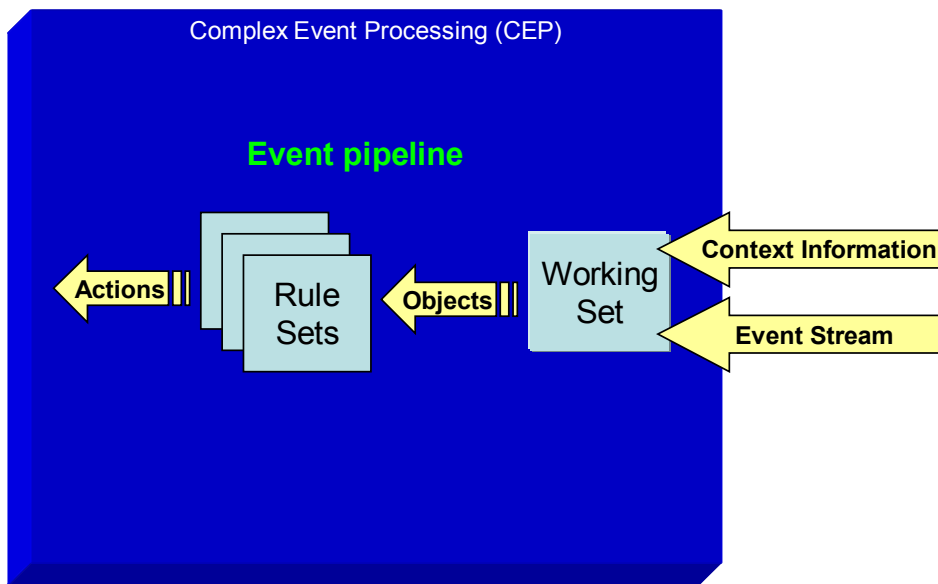
Middleware



Overview

The middleware has several components, and this document is focused on the complex event processing (CEP). The CEP component will utilize the ALE services to obtain a subscription to the event stream; it will also utilize the integration components (not shown in diagram) to obtain enterprise information for business context determination, and to execute subsequent business rules. See Middleware design document for additional overview information (i.e. Pallet manifest is considered business context information).

Complex Event Processing



Overview

There are two major types of rules, discrete and temporal. The discrete rules, as the name implies will operate on discrete independent event cycles, and have no relation to events that have happened in the past. Temporal rules depend on the current event cycle, and event cycles from the past for a specified amount of time.

Rule sets are created and may have both types of rules, however, a temporal rule definition will a temporal working set of events. It may be advantages to separate temporal and discrete rule sets, and the determination for this may appear during performance testing of the configuration.

Working sets are the objects that the rules are fired against and are static while the rules are firing. The working sets will consist of sensor events as well as enterprise context information such as an advanced shipping notice, or pallet manifest.

Note: The context information will be accessed through a Data Access Object (DAO) and the underlying implementation will implement caching. The context information may be remote or local. Depending on performance requirements the context information may need replicated to local repositories from enterprise systems. The integration components will abstract interaction with enterprise systems, and provide near real time interface or participate in a replication model to populate local repositories.

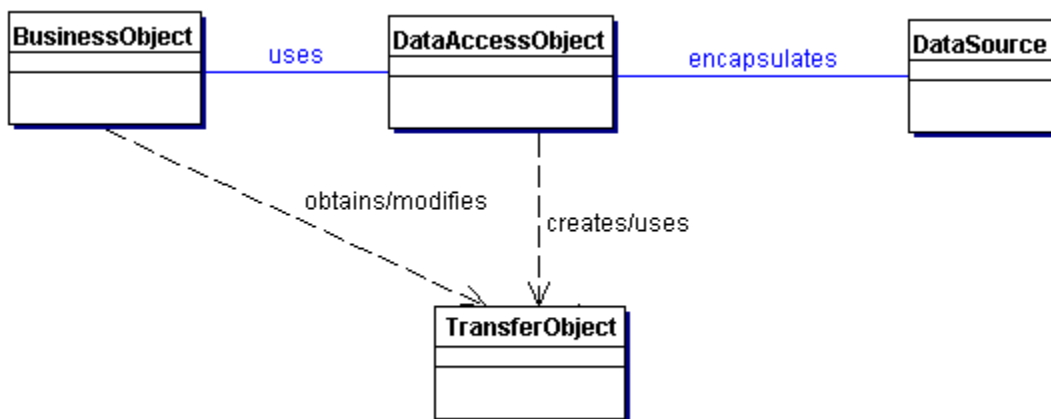
Standard objects such as ASN will be created, however, other custom objects to create rules against can be created. Customization development kit will be provided to give guidance in creating DAO, as well as integration components.

The Working Set and subsequent rules assigned it can also be assigned as a filter at ECSoec definition that is incorporated into generation of and ECReport. As well as act as an ALE Client for to determine other more complex events that must be generated.

Rules Engine

Using [Drools](#) as an implementation of Rete algorithm. The rules engine will allow definition of complex event detection, and the subsequent actions which can include starting a business process.

Business Context Objects (BCO)



The BCO will be made available to the rules and utilize the DAO to obtain the information, the DAO may access the information locally or remotely depending on configuration. The implementation will follow the [DAO Design Pattern](#).

Event Stream Working Set

The working set for the rules is created by subscription to ECSpec through the ALE service. A Message Driven Bean (MDB) will be created when a Working Set is associated with an ECSpec. The ALE service will deliver the event cycle to the Working Set MDB. The working set will be populated with the events and then the RuleSet notified the Working Set has been updated. The Working Set may be temporal or created just for the current event cycle. It may be associated with many ECSpec, and it also may be populated with additional data from local or remote repositories.

Process

1. Define rule set (rules, conditions, and consequences, when to fire).
2. Define working set (Data Feeds, Time To Live)
3. Generate MDB for working set.
4. Accept information from event stream through ALE subscription to WS-MDB.
5. Define local or remote DAO to support information stored in repository to WS-MDB