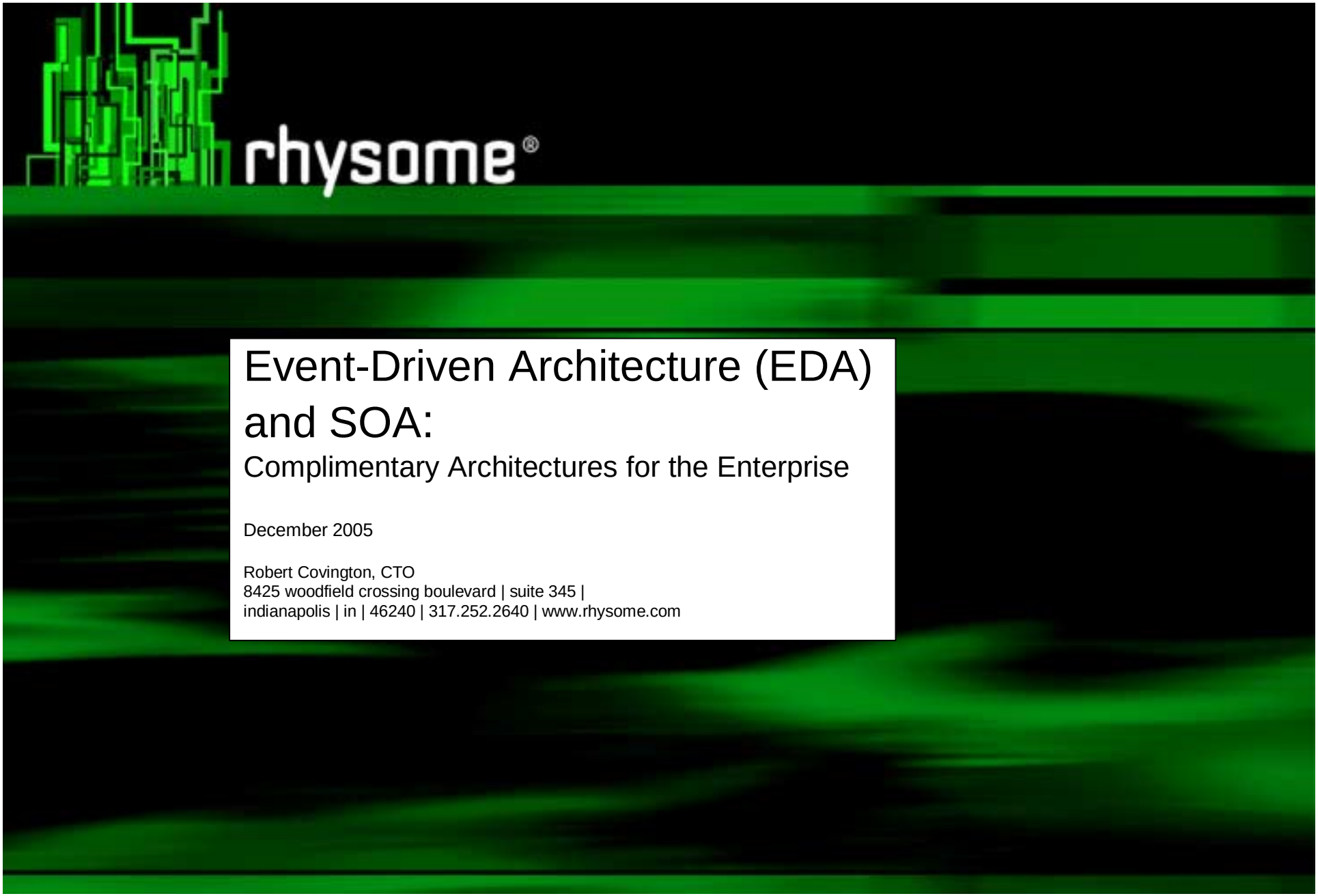

Event-Driven Architecture (EDA) and SOA:

Complimentary Architectures for the Enterprise

December 2005

Robert Covington, CTO
8425 woodfield crossing boulevard | suite 345 |
indianapolis | in | 46240 | 317.252.2640 | www.rhysome.com

Why are we talking about EDA here?

- Event-Driven Architectures (EDA) complement Service Oriented Architectures (SOA).
- Understanding which applications are optimized for each architecture and how they interface to each other allows for improved architecture designs.
- SOA standards need to be developed to better define an interface to EDA.
- EDA standardization is just beginning

Agenda

- Overview of EDA
- EDA and SOA
- OMG Opportunities
- Next Steps

What is an event?

Dictionary Definition:

> An important incident: an occurrence, especially one that is particularly significant, interesting, exciting, or unusual[1]

Characteristics in Enterprise Infrastructures:

- Events generally reflect a change of state in an object
- Events are fine grained – "sub-transactional"
- Events have a temporal and locational elements
- Events may not always represent data
- Events may not always generate a message
- Events may not be associated with a transaction
- Some events are emitted from the source, some are derived from 3rd party systems.

Examples:

- A customer order transaction fails
- A database schema changes
- A network device going down
- A tornado hits Evansville, IN
- A pattern of fraudulent transactions
- A machine has been turned off

1. Microsoft Encarta Dictionary,2005

Types of Events in an Enterprise

- Physical Events
  - The power has been turned off
  - The temperature is below 0 Celsius
  - A tornado hits Evansville, IN
- Transactional Events
  - A new customer has been added
  - An claim has been filed
  - A check has been received
- Aggregate Events
  - The NASAQ average has increased 20% over the last 48 hours
- Relational Events
  - Auto claims filed in Louisiana in the last 7 days are greater than 25% for the previous week AND life insurance claims in Louisiana are over 40% for the previous week
- Complex Events
  - Combinations of the above in an event hierarchy

Event Relationships

Causality

- An event is caused by another event if the first event could not have occurred with out the occurrence of the second event.
- Examples describing causal relationships
  - The presence of heat causes water to boil.
  - My pushing of the accelerator caused the car to go faster.
- Causality relationships are dynamic:
  - May evolve over time as new events are introduced
  - May be difficult to predict utilizing linear methods
    - – You can't always determine if someone has a fever by comparing their temperature to 98.6 F.

Event Relationships (continued)

- Causality Types
  - Observed causality
    - Event A has been observed to cause Event B
    - Examples
      - Changing a database schema effects a group of applications
      - Water at sea level boils at 100C
  - Inferred causality
    - Based on statistical analysis, there is a high probability of a causality relationship between 2 events.
    - Example
      - When Bob turns 21, he is less likely to have an automobile accident
  - Concurrency
    - Events which occurred at the same time but did not cause each other.
    - Example
      - I landed in San Francisco and the NYSE fell 30 points

What problems does an EDA solve?

- Capturing events from many diverse sources – including those that do not emit event streams – in real-time.
- Efficiently processing and storing fine grained asynchronous events in near real-time:
  - Optimized for dynamic, parallel, asynchronous processing.
  - Able to scale to hundreds of thousands of events per second.
- Efficiently correlating information from a large number of events over a long period of time into useful information in near real-time.
  - Use of declarative rules reduces correlation modeling complexity and enables dynamic modeling.
  - Some implementations implement both rules based and machine learning based correlation techniques
- An EDA is optimized for problems where the Variety, Volume, and Velocity of information is an issue
- An EDA is optimized for enterprise handling exceptions in a declarative way

rhysome®

Overview of EDA

EDA and SOA

OMG Opportunities

Next Steps

SOA vs. EDA

Service Oriented Architectures (SOA) enable applications to move from centralized, tightly coupled, and monolithic to distributed, loosely coupled, and modular.

SOA makes connecting linear, predictable sequences and business processes efficient, as long as all applications have been SOA-enabled.

This evolution, while dramatic does not address one of the fundamental problems with traditional application design - it's sequential process orientation.

In order to process asynchronous, less reliable business conditions - those that may or may not occur, or may occur in any sequence over time – an Event Driven Architecture is required.

Further, EDA allows for *multiple*, *less predictable, asynchronous events to happen in parallel and trigger a single action.*

SOA and EDA

- Service Oriented Architectures (SOA):
  - Expose legacy and new applications with a standard interface.
  - Optimized for course grained objects and transactions.
  - Improve flexibility with dynamic composition of applications. IF - THEN
  - Allow data, applications, and workflows to extend automatically and in real-time beyond the enterprise.
  - Optimized for linear predictable sequences.
  - Focused on the present.
  - Enables dynamic applications. –
- EDA coupled with SOA provides the following capabilities:
  - EDA provides a feed to SOA
  - Optimized for finely grained decoupled event processing capable of identifying and responding to asynchronous, non-predicable or complex sequences of business activity.
  - Improves SOA scalability
  - Focused on correlating complex relationships of events based on past trends and future predictions. WHEN-THEN
  - The ability to integrate non-transactional events into business processes.
  - Enables dynamic modeling.

Contrasting SOA and EDA

Service Oriented	Event Driven
Focuses on one-to-one connections	Supports many-to-many connections
Flow routing directed by the client (sender)	Flow control is determined by the recipient based on the message and event rules
Uses linear execution path through hierarchy of modules	Supports dynamic, parallel, asynchronous flow through network of modules
Closed to new, unforeseen input once process flow starts	Reacts to new, external input arriving at unpredictable times

application type properties

	Structured Applications	Object Oriented	Component Based	Service Oriented	Event Driven
Coupling	Tight	Tight	Medium	Loose	Decoupled
Granularity	Very Fine	Fine	Medium	Course	Any
Contract	Defined	Public / Private	Public	Published	Published
Reusability	Low	Low	Medium	High	High
Dependencies Determined at	Compile Time	Compile Time	Compile Time	Run Time Source	Run Time Subscriber
Scope	Intra-Application	Intra-Application	Inter-Application	Inter-Enterprise	Inter-Enterprise

rhysome®

Overview of EDA

EDA and SOA

OMG Opportunities

Next Steps

OMG opportunities

- SOA WG/SIG?
  - Formalize a standard for modeling the interfacing SOA and EDA
- Analysis & Design TF/Ontology SIG
  - Standardize the Ontology for EDA
    - Use Ontology Definition Metamodel (ODM)
- Business Process Definition Metamodel
  - Causality Relationship Modeling
- Agent SIG
  - Integration of EDA into Agent standards
- Extending OMG's COS Event Service Specification to incorporate modern EDA capabilities

rhysome®

Overview of EDA

EDA and SOA

OMG Opportunities

Next Steps

Next Steps

- Identify other working groups inside of OMG regarding EDA.
- Propose a specifications for SOA and EDA integration.
- Determine the feasibility of starting an EDA Work Group inside the OMG.