



# Automated Methods for Integrating Systems (AMIS) Project Overview

*(presenter)*

Manufacturing Systems Integration Division  
National Institute of Standards & Technology

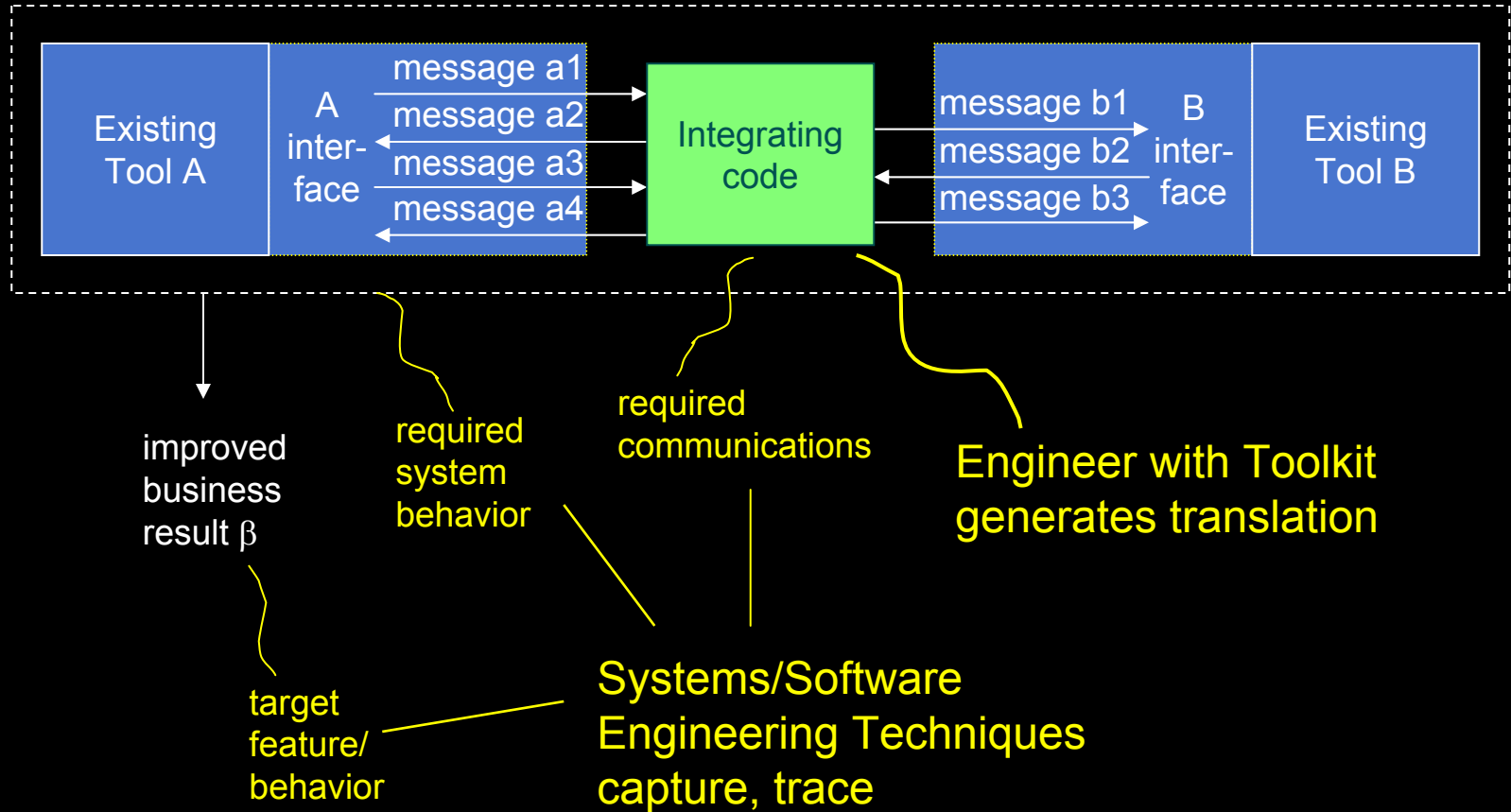
# Motivation

- Problem: Integration of systems in manufacturing enterprises

*Integration is a constant process driven by:*

- New technologies
  - New capabilities
  - New ways of doing business
  - New partners
- Old solution: Standards
- Alternative: Automate Integration

# Typical Integration Problem



# AMIS approach

- **Ontology Extraction:**
  - Capture Local Interaction Models of existing software systems or standards
- **Specify Joint Action Model for the integration**
- **Semantic Mapping:**
  - Map the Joint Action Model to the Local Interaction Models of the components
- **Connector Transformation:**
  - Generate Wrappers or Interceptors to link the engineered interface implementations

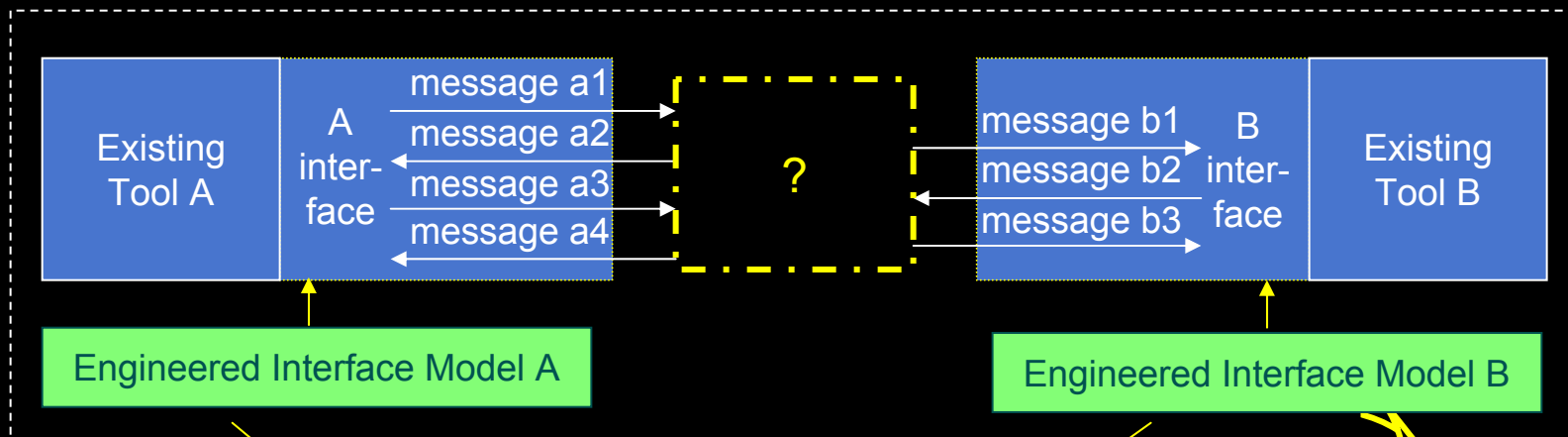
# AMIS approach

- **Ontology Extraction**
- Specify Joint Action Model
- Semantic Mapping
- Connector Transformation

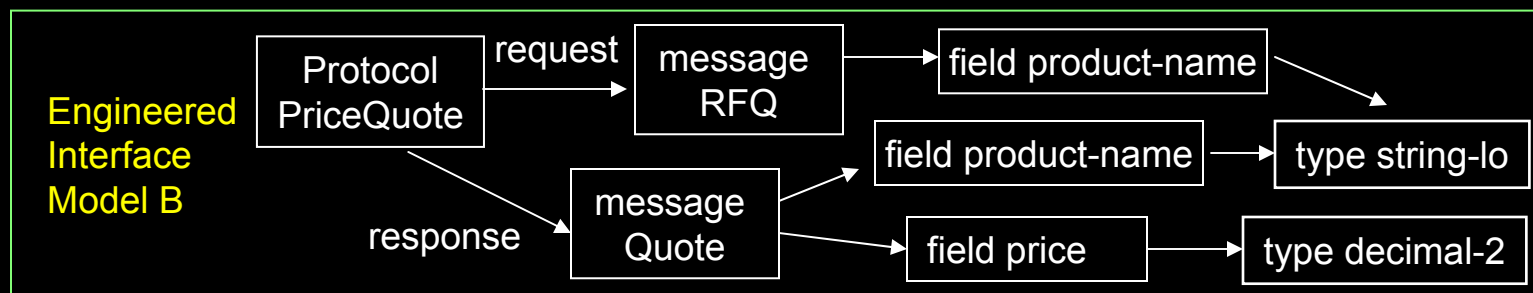
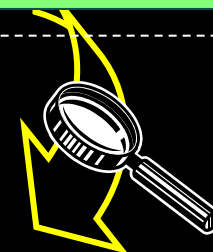
# "Ontology Extraction"

- Capture *Local Interaction Model* of existing software or standard
  - Capture “business” concepts underlying the software design = *Local Conceptual Model*
  - Capture “technical” interaction concerns = *Local Engineered Interface Model*
  - Link business concepts (entities, properties, relationships, actions) to technical representations
  - Capture all models and links in a form suitable for automated reasoning

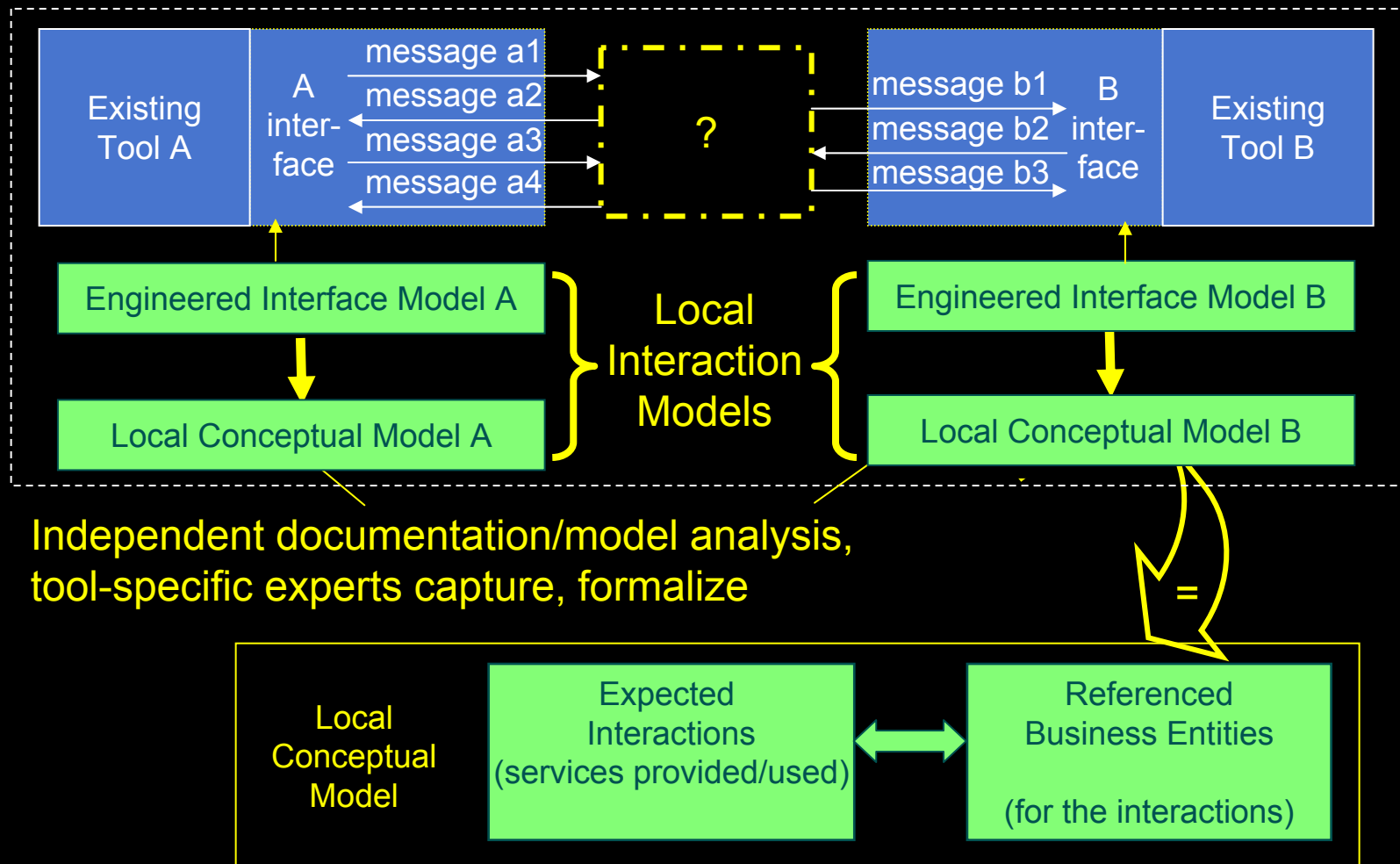
# Engineered Interface Capture



Software model (Express, UML, WSDL, IDL) analysis techniques capture, produce standard form



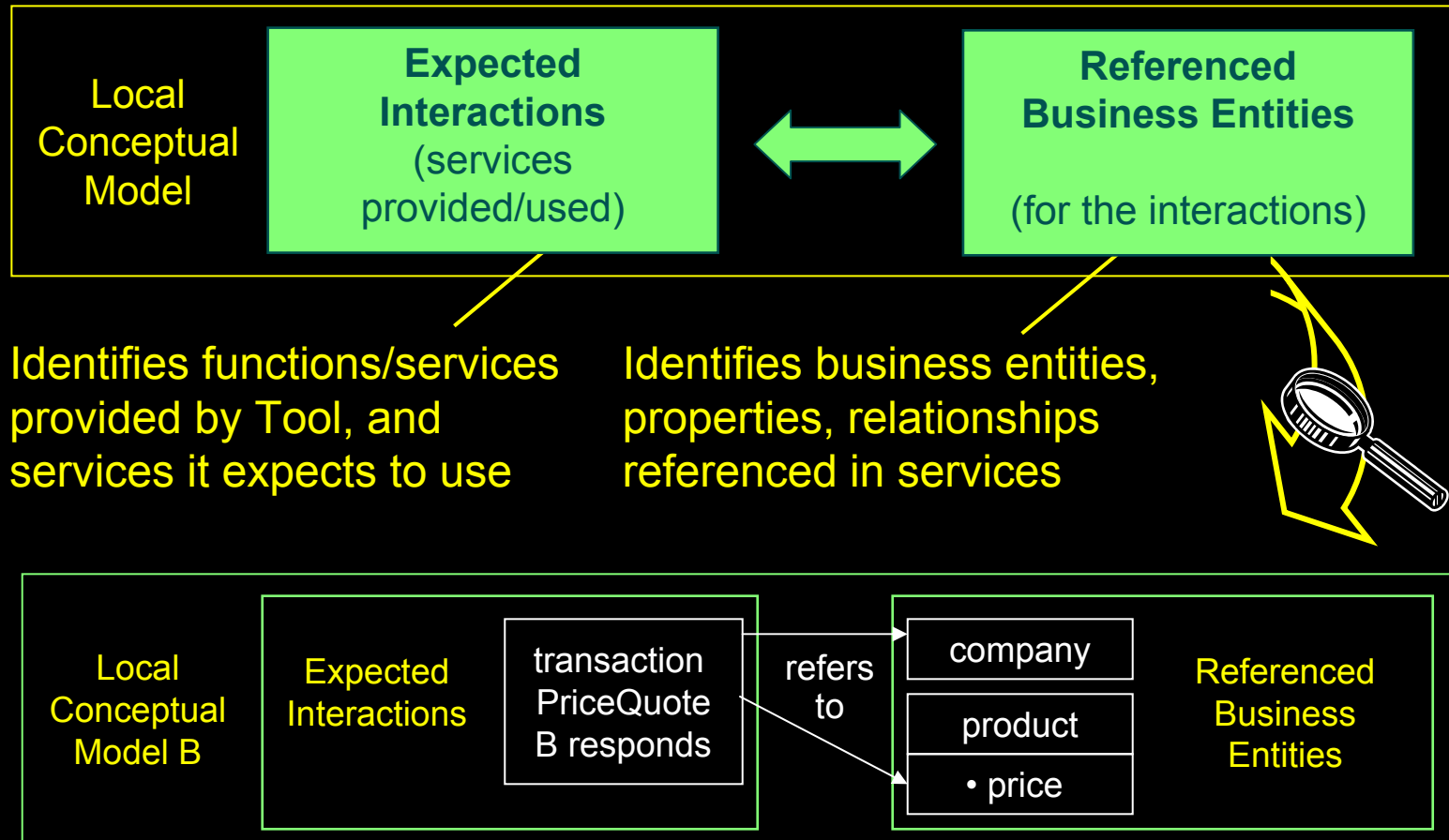
# Local Conceptual Model Capture



Independent documentation/model analysis,  
tool-specific experts capture, formalize

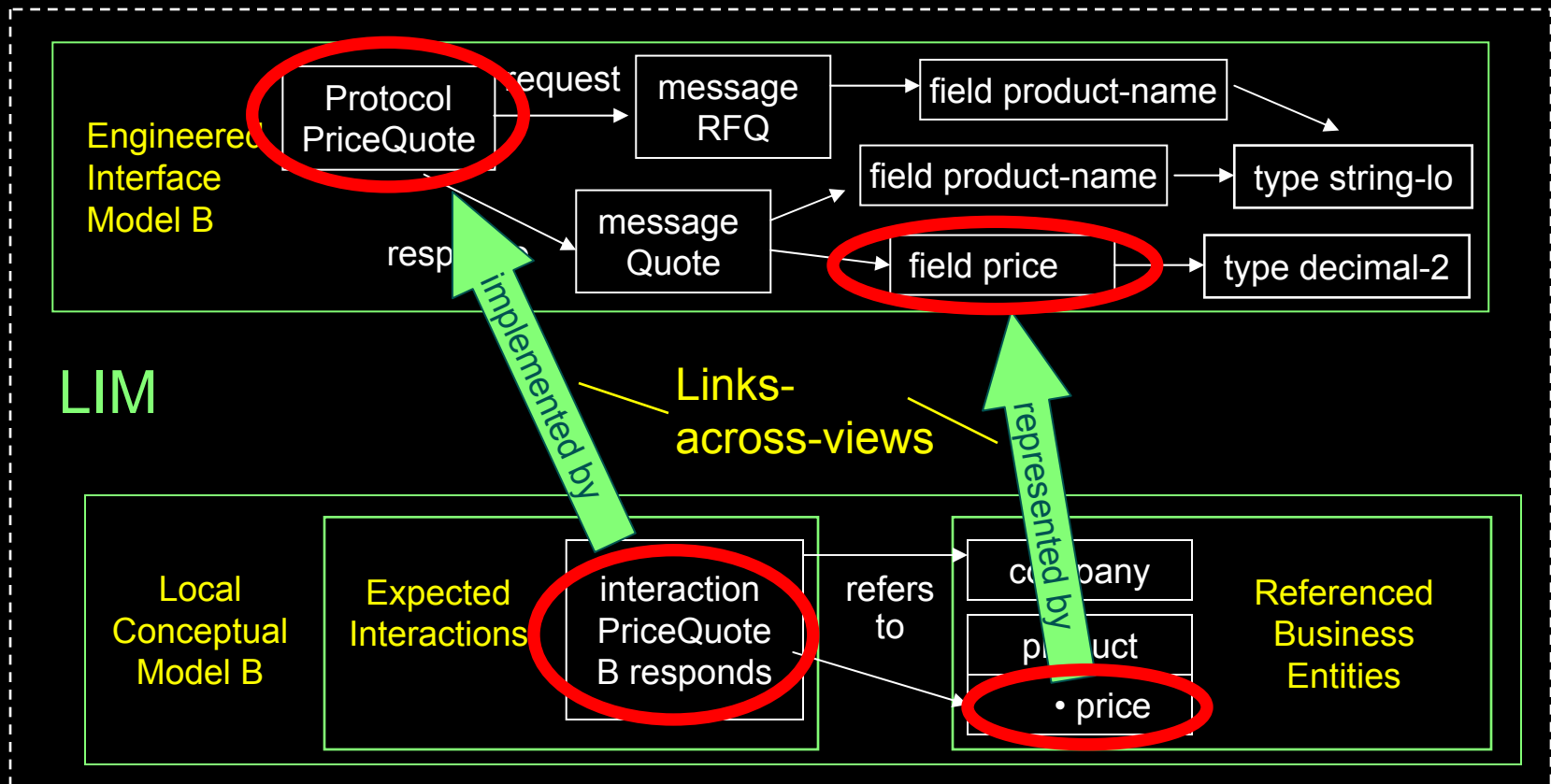


# Local Conceptual Model



# Links-across-views Capture

Relates local business elements to engineered interface elements  
Captured with the Local Interaction Model



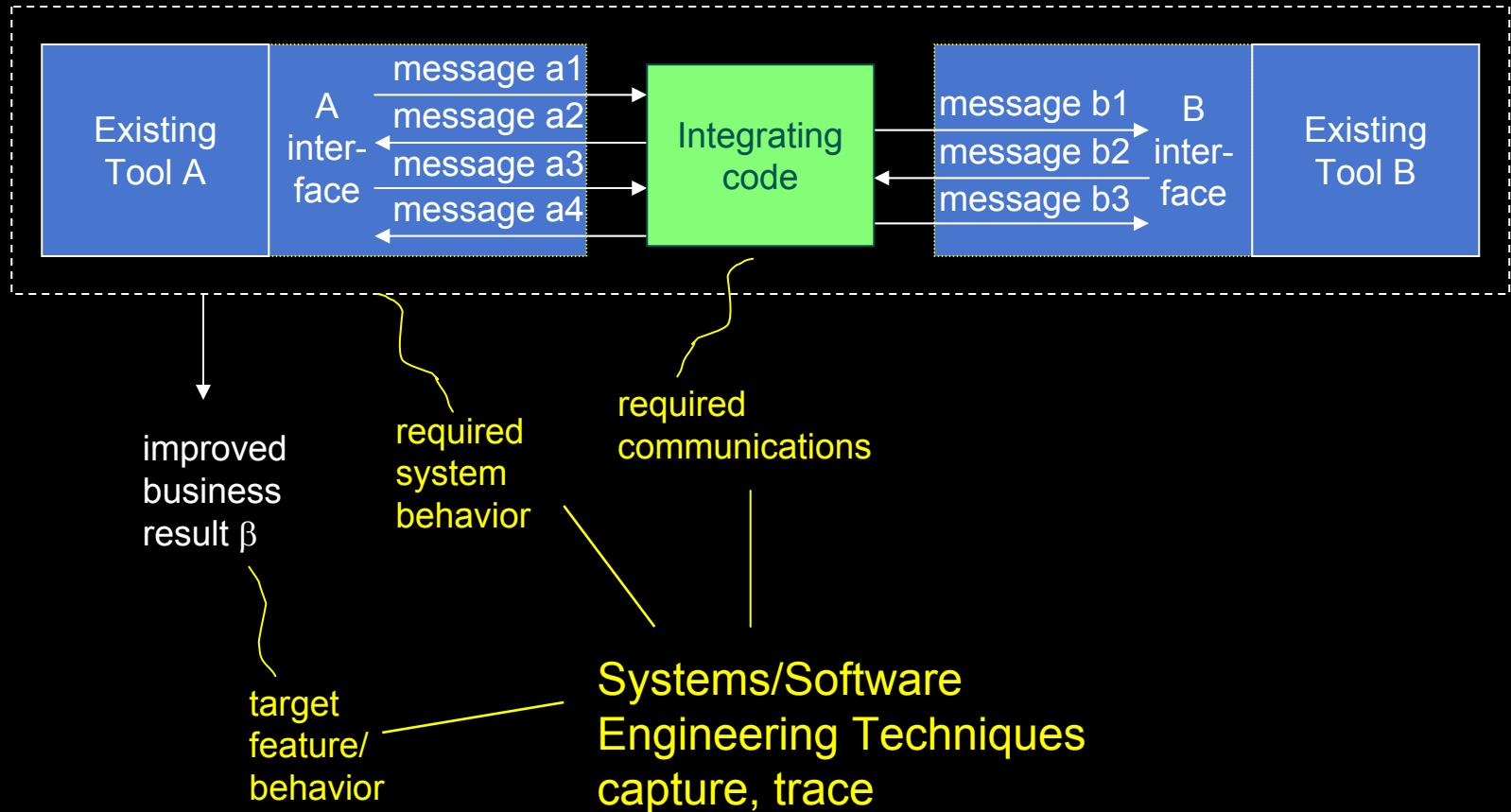
# AMIS approach

- Ontology Extraction
- Specify Joint Action Model
- Semantic Mapping
- Connector Transformation

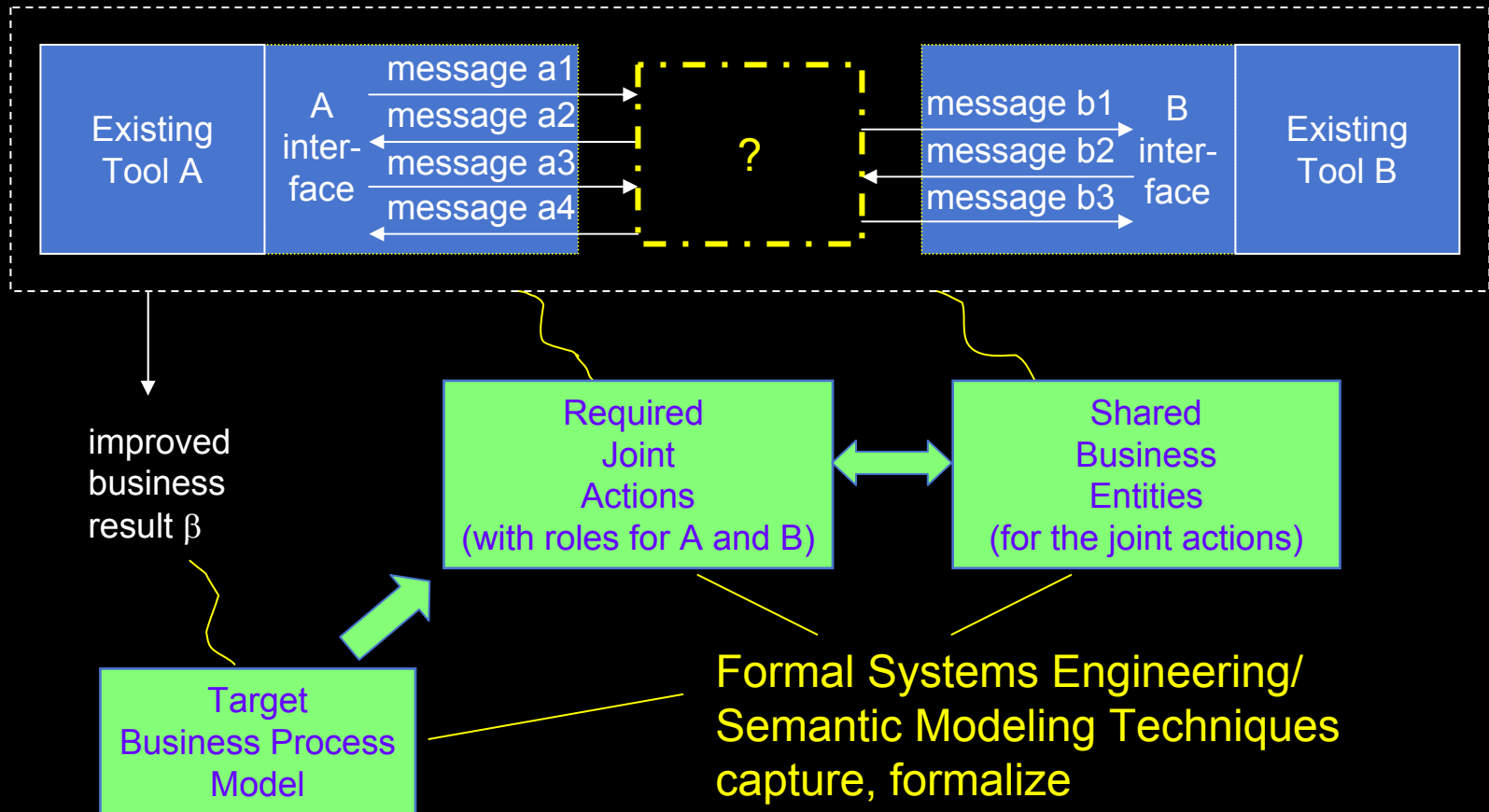
# Specify the Joint Action Model

- Specify the *Joint Action Model* for the integration
  - Capture the required interactions in business terms
  - Capture the business entities used and affected
  - Joint Action Model is a conceptual model, will have same basic structures as a Local Conceptual Model, but perhaps different organization and terms

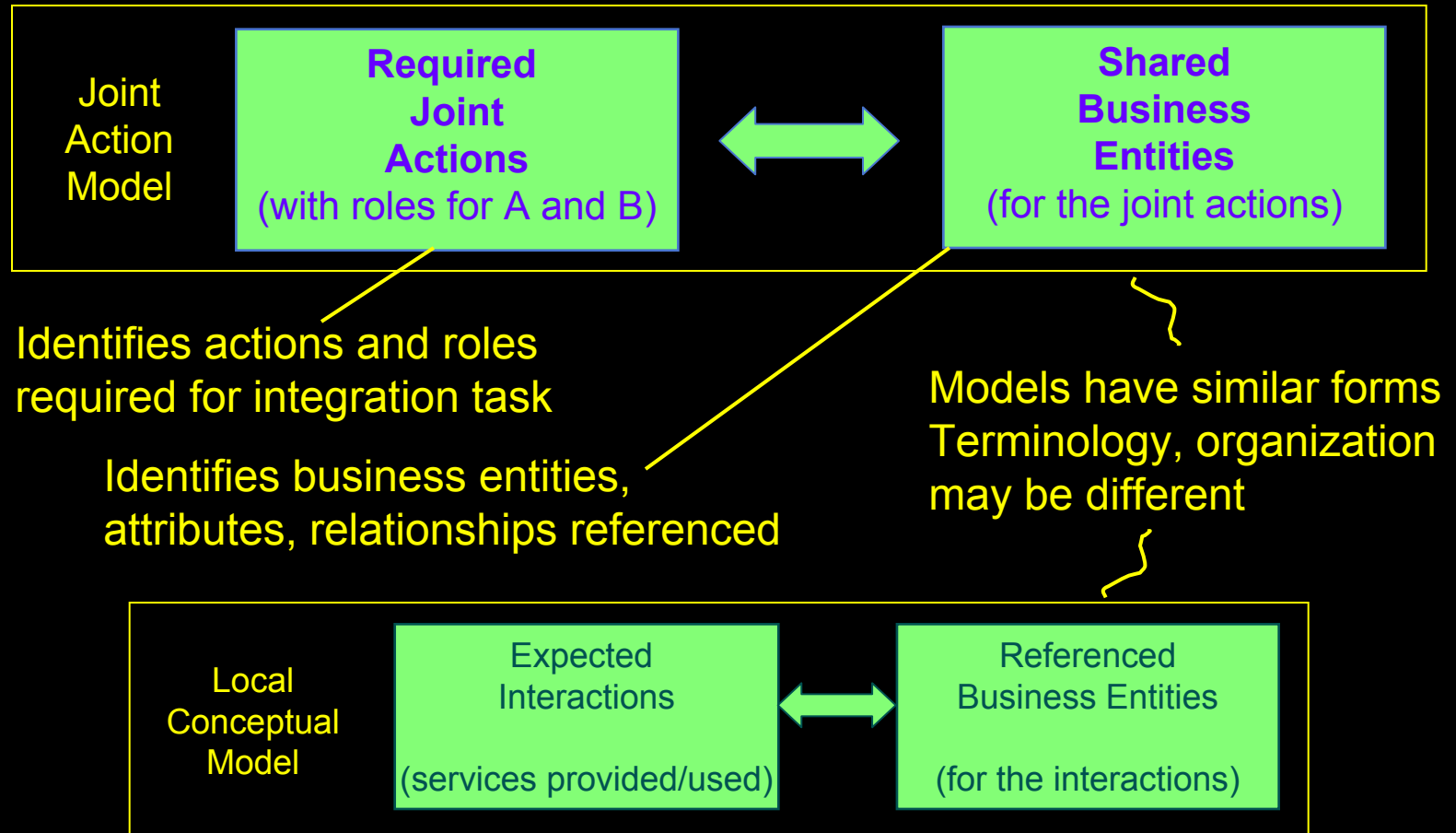
# Human Conceptualization



# Joint Action Specification



# Joint Action Model



# AMIS approach

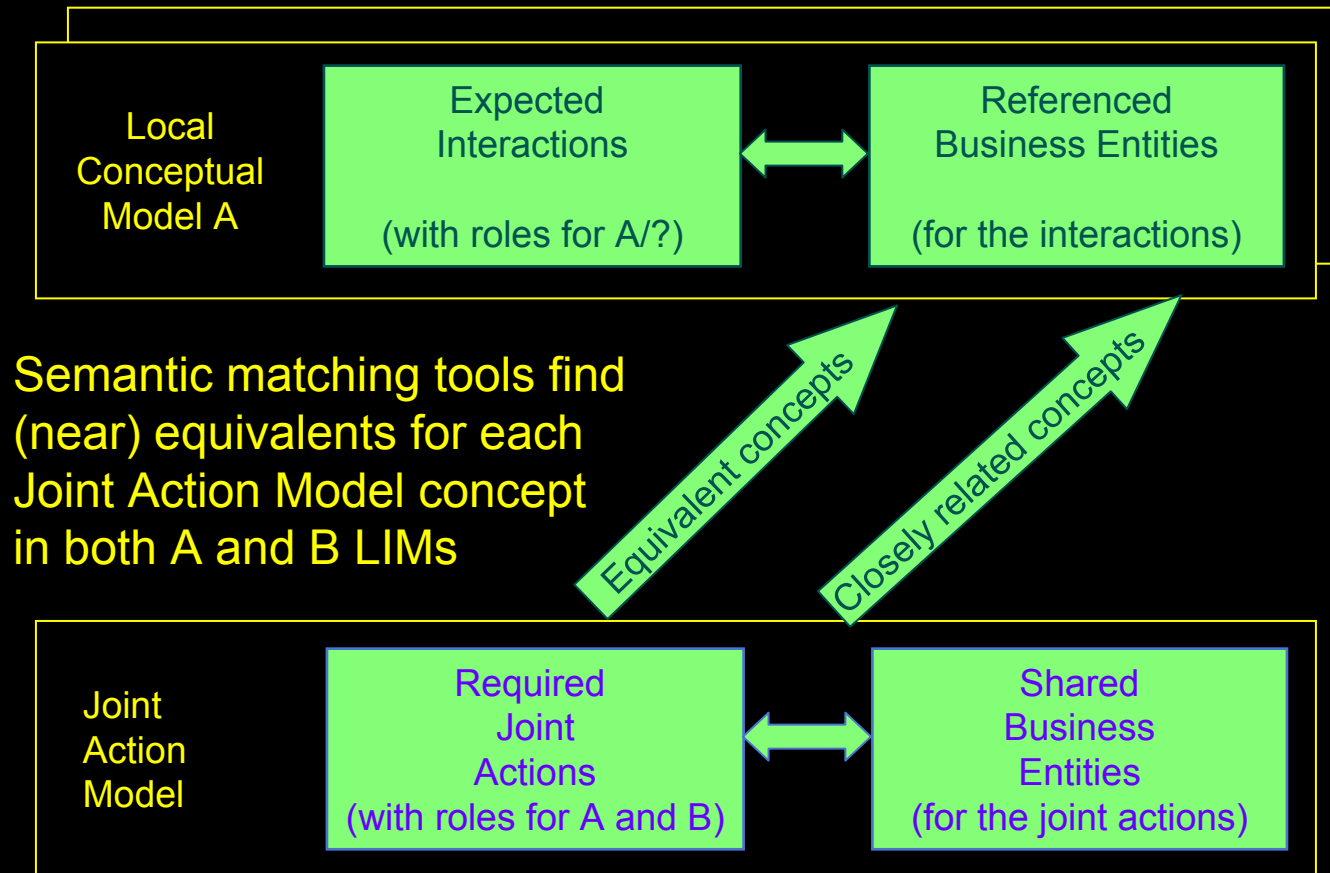
- Ontology Extraction
- Specify Joint Action Model
- **Semantic Mapping**
- Connector Transformation



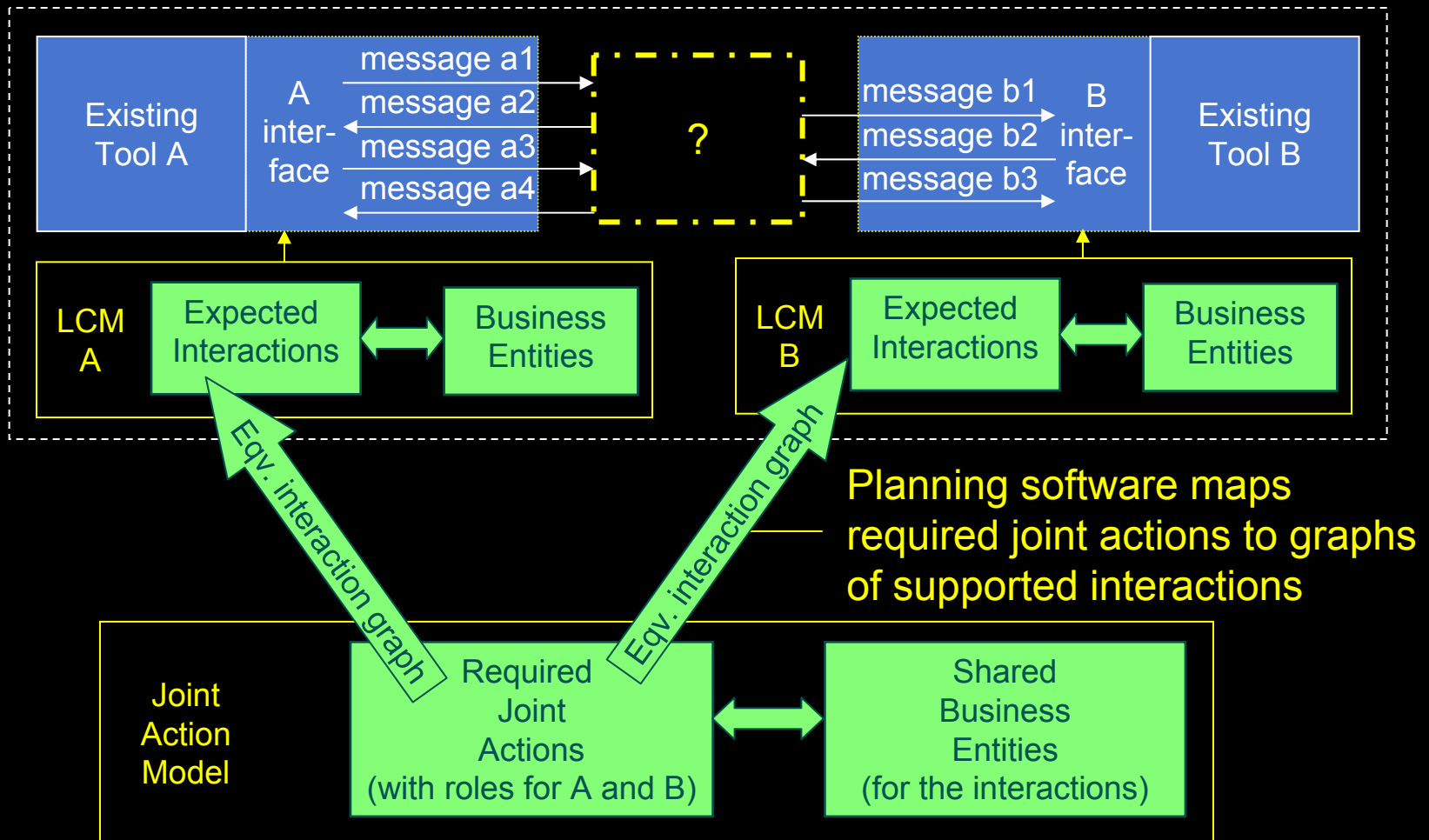
# Semantic Mapping

- Match Joint Action Model terminology with Local Interaction Model terminologies
- Interpret Joint Actions to graphs of expected interactions specified in LIMs
- Compose Semantic Maps for terms/actions with Links-across-views => Integration-end maps
  - define engineering mappings for JAM actions, messages and information units
  - one mapping for each participating system

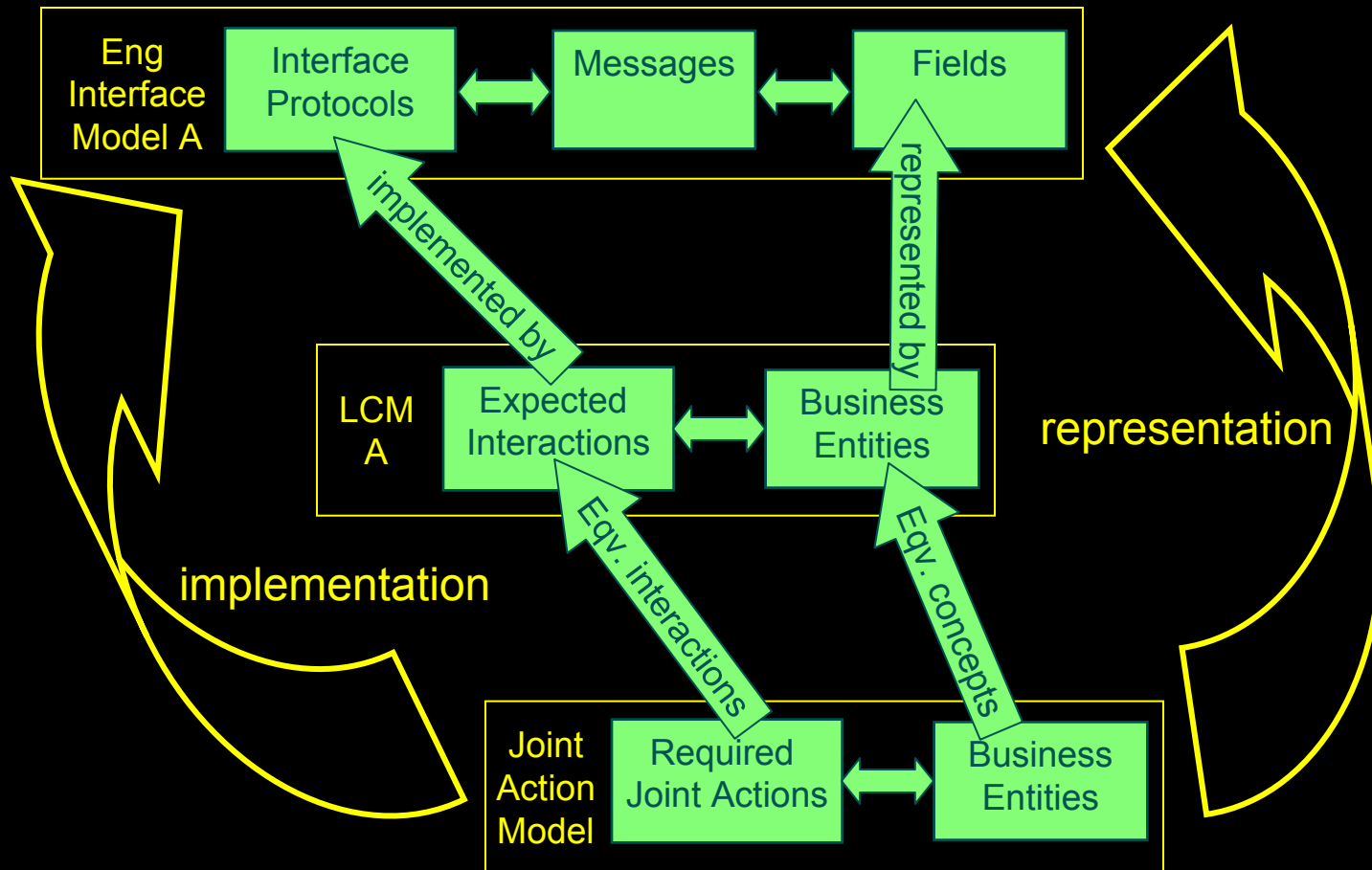
# Terminology Mapping



# Interaction Planning

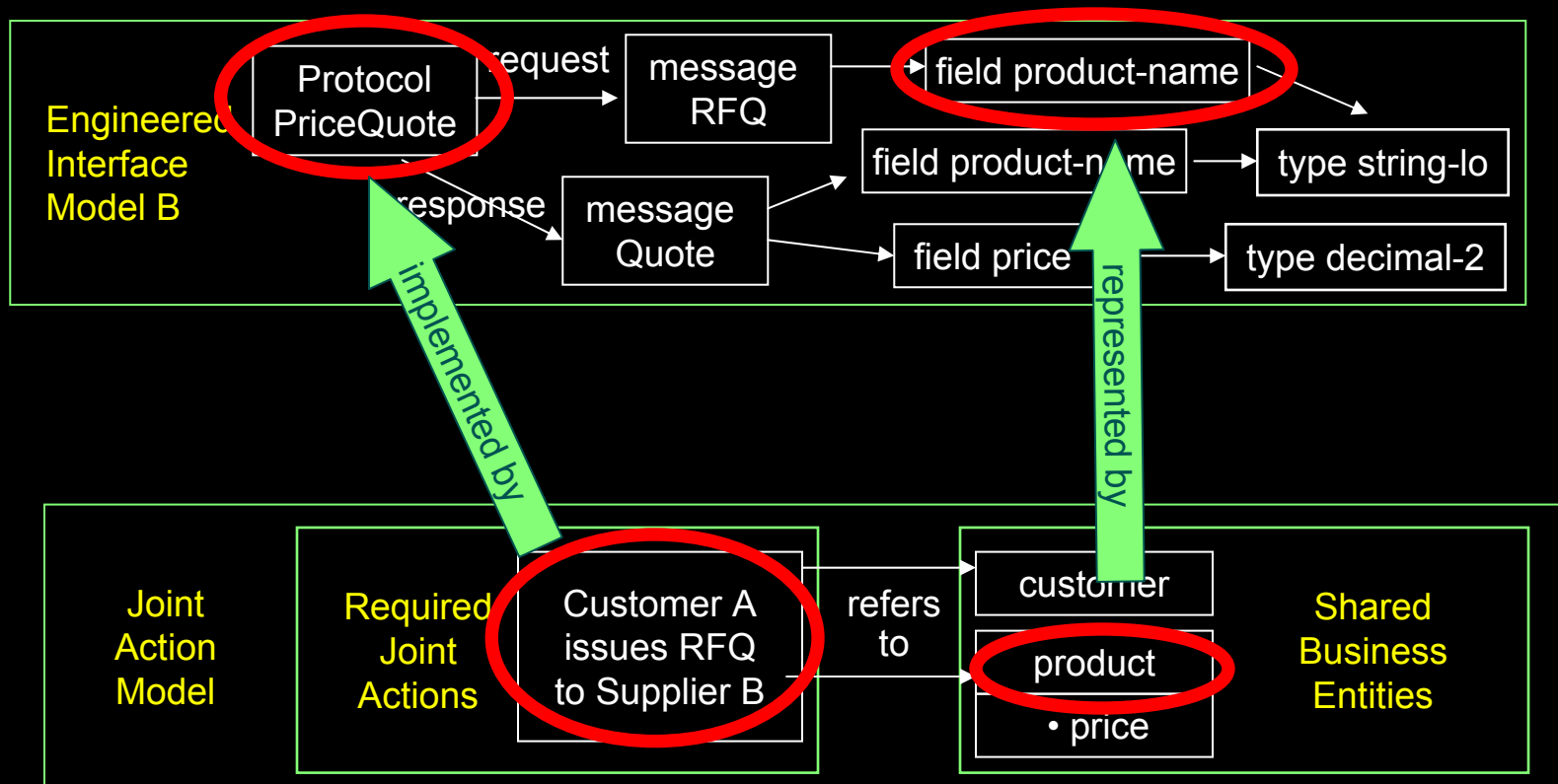


# Integration-End Mapping



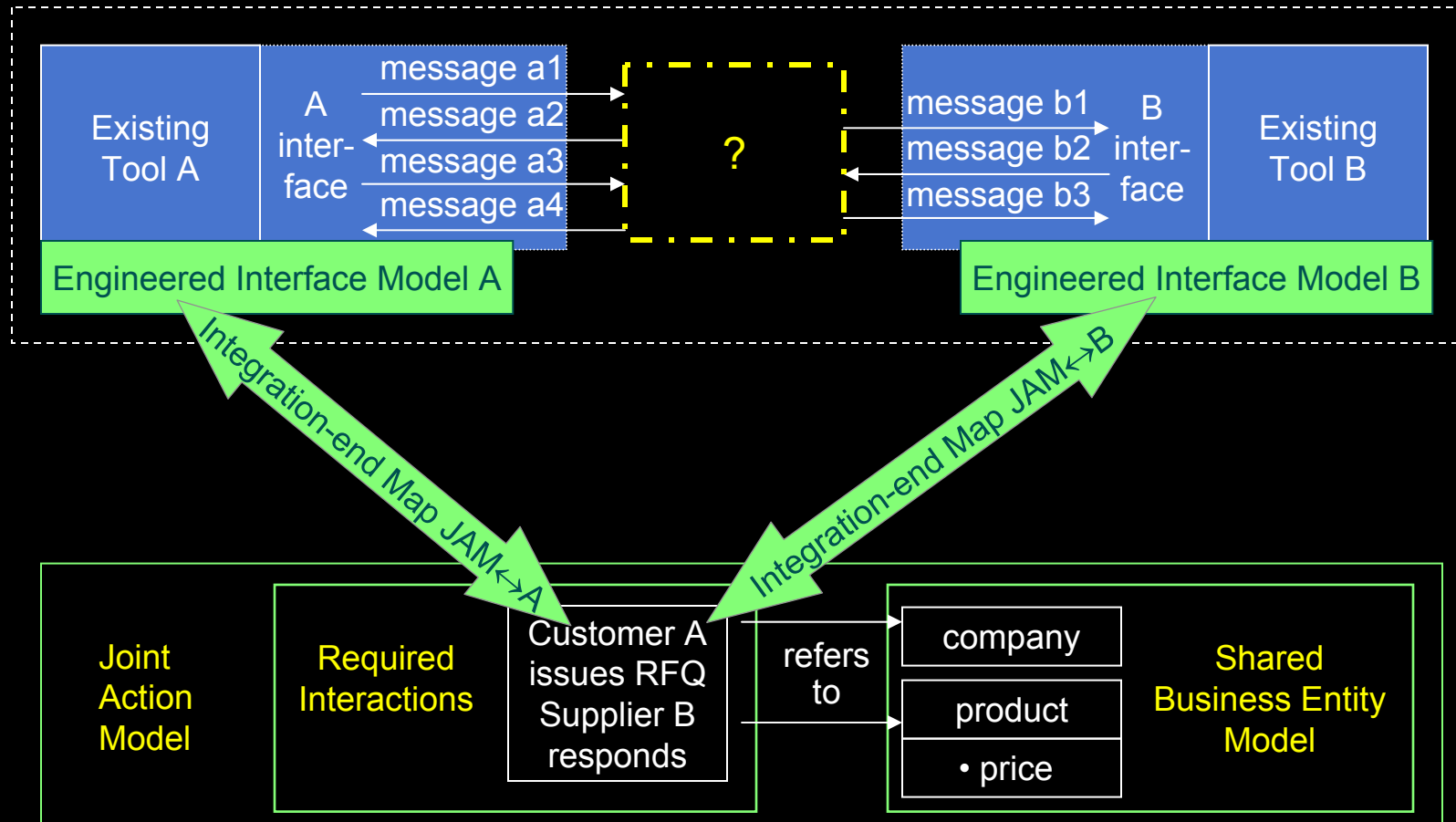
# Integration-End Map example

Relates Joint Action elements to engineered interface elements for one end (component) of the joint action



# Integration Mapping

JAM transaction is mapped to messages/fields in each EIM  
Defines technical integration requirements



# AMIS approach

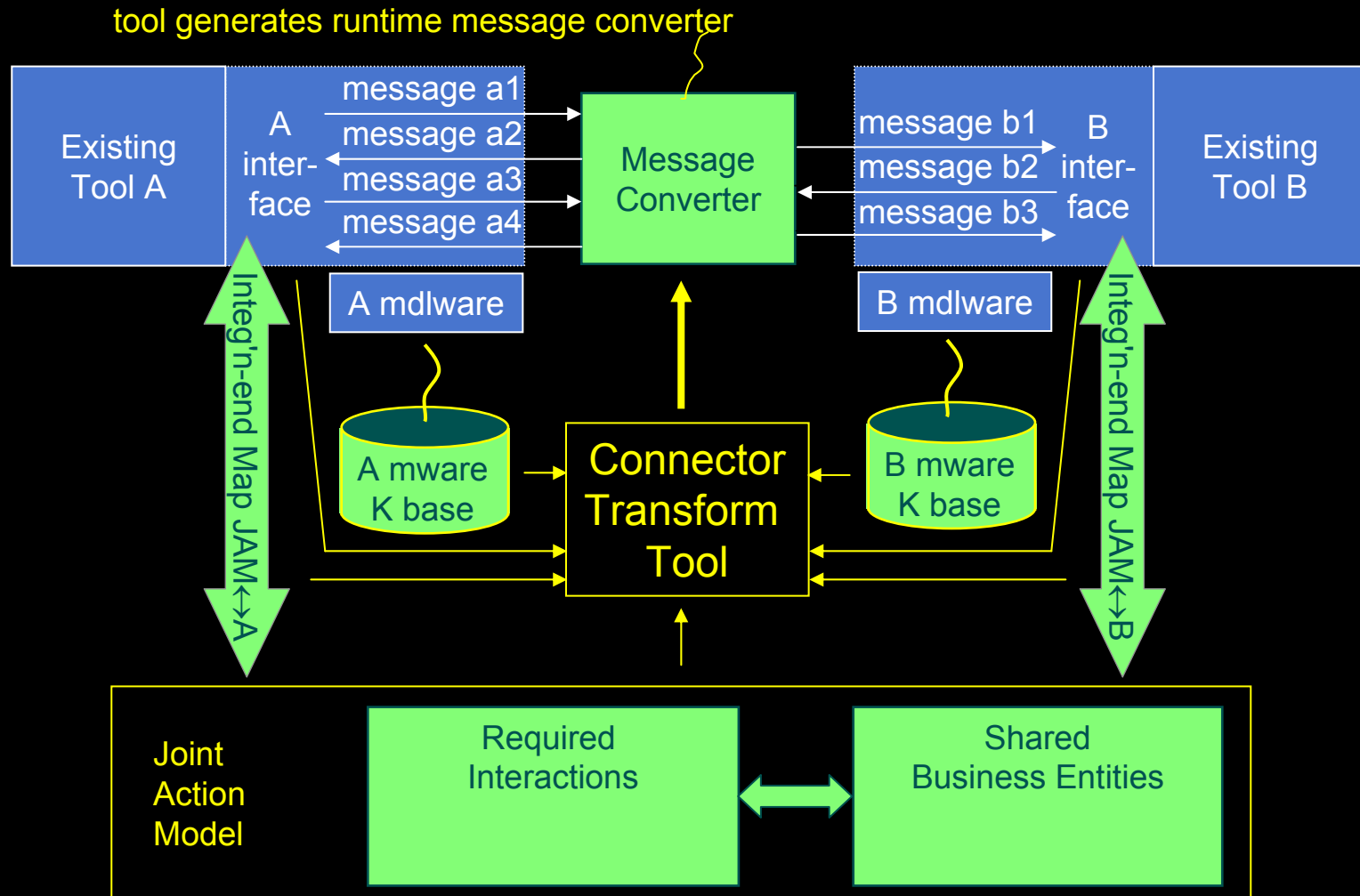
- Ontology Extraction
- Specify Joint Action Model
- Semantic Mapping
- **Connector Transformation**

# Connector Transformation

- Generate Wrappers or Interceptors to link the engineered interfaces
  - Based on the JAM, the Local EIMs and the Integration-End mappings
  - Formalize all information required for transformations of syntax, structure, and choreography, down to lowest level of abstraction
  - Tools generate dynamic message converters
  - Expand to support dynamic “technology” conversion; specific knowledge bases for “middleware technologies” must be developed

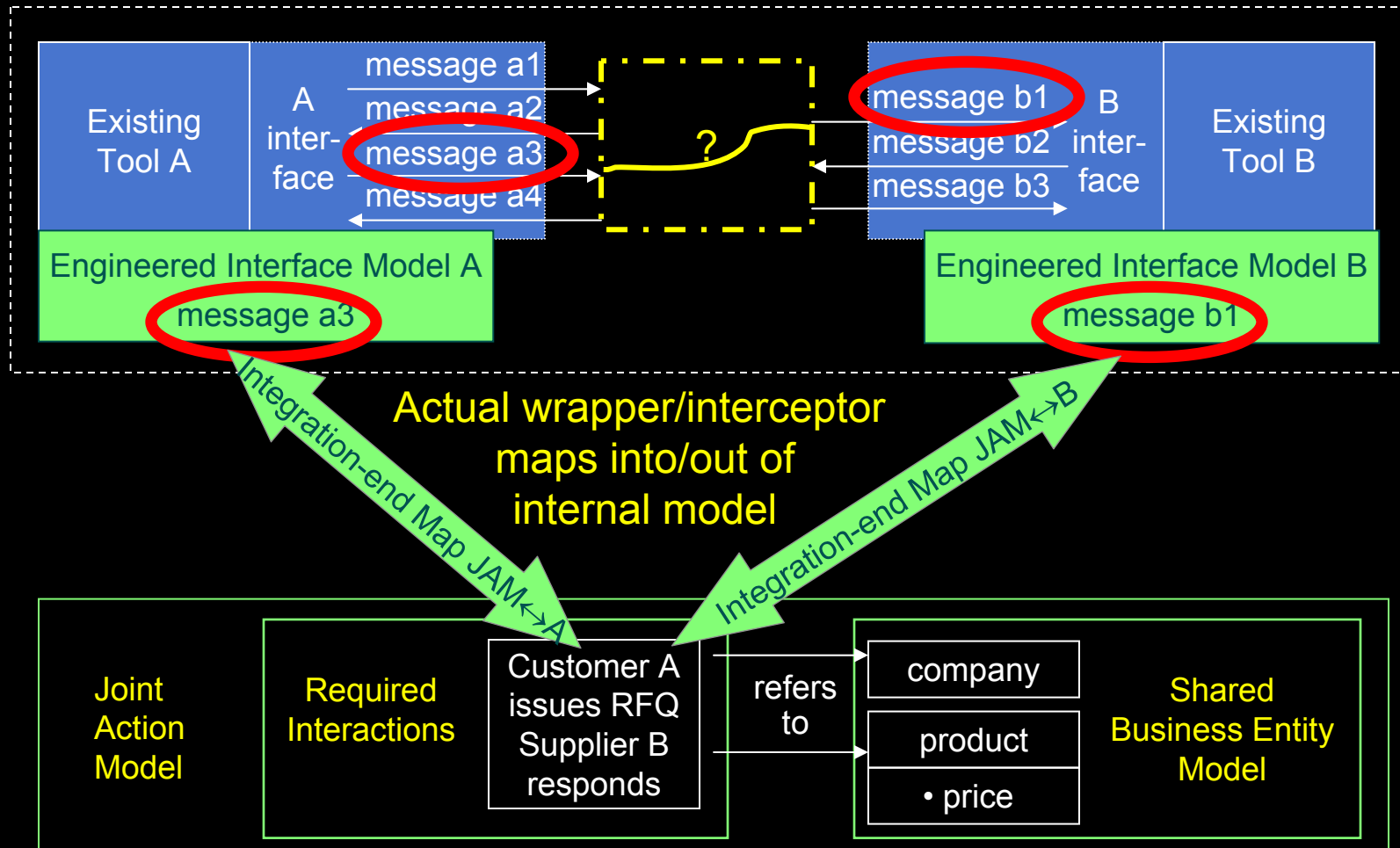


# Connector Transform Tool



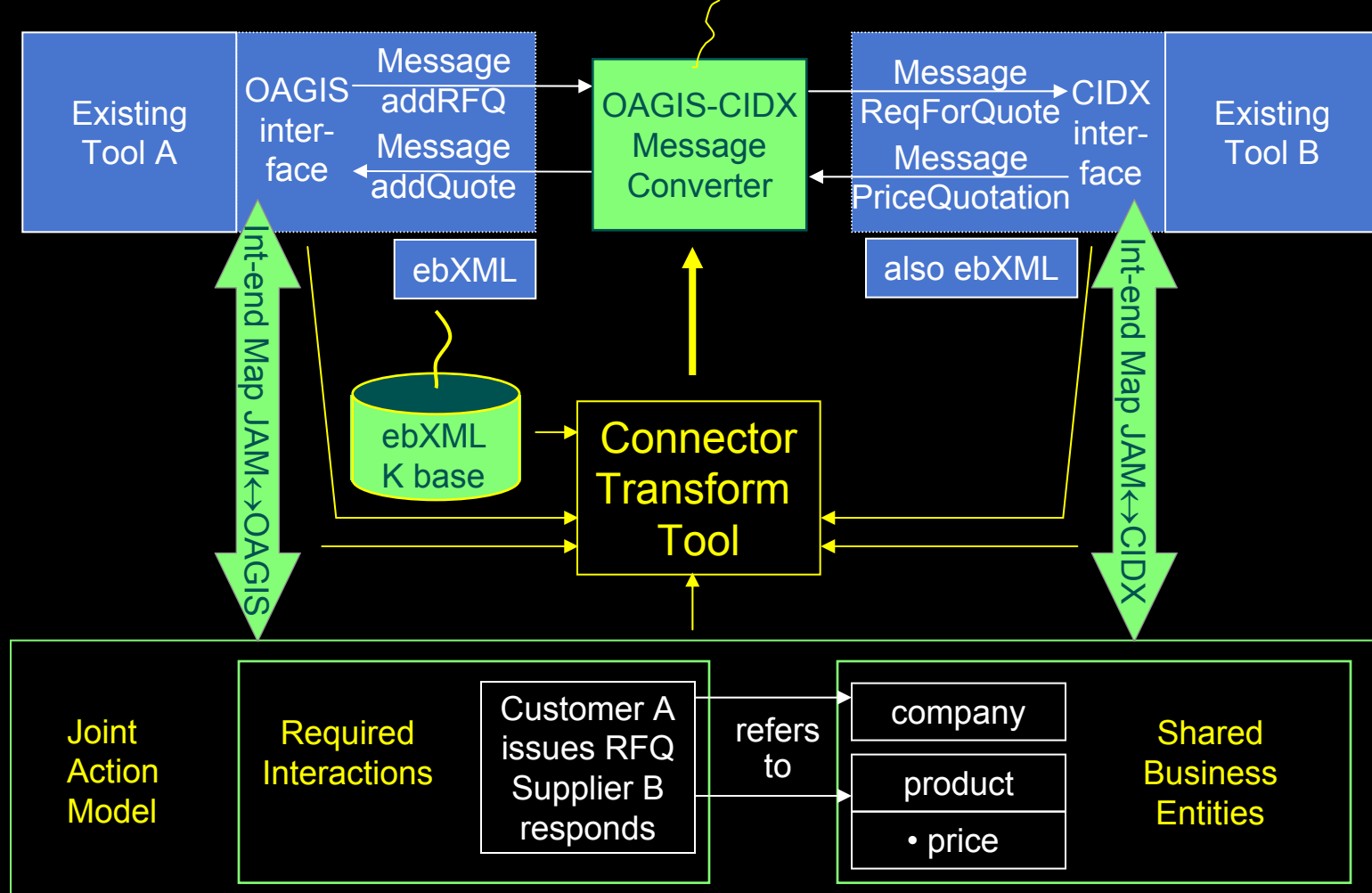
# Nominal Message Mapping

Composing integration-end mappings defines required message mappings



# Example Message Converter

tool generates runtime message converter for CIDX ↔ OAGIS



# Conclusions

- Challenges
  - Constructing a useful knowledge framework for Local Interaction Models and Joint Action Models
  - Extracting Local conceptual models and engineering models from specifications as received
  - Defining “semantic mapping” algorithms
  - Building a sufficient knowledge base for transformations
  - Automating analysis and resolution of technical mismatches

# Conclusions

- Value added
  - Improve interface/service specifications
  - Improve knowledge capture for existing software systems and standards
  - **If it works**, greatly reduce the time and cost of “systems integration” projects
  - **Otherwise**, identify the unsolved problems and provide knowledge for new toolkits

# AMIS Project Staff

- Ed Barkmeyer – project leader
  - edbark@nist.gov
- Peter Denno – ontology extraction
  - pdenno@nist.gov
- Dave Flater, Don Libes – connector transforms
  - dflater@nist.gov, libes@nist.gov
- Evan Wallace – semantic mapping
  - ewallace@nist.gov
- Nenad Ivezic and NIST Testbed staff
  - scenario and JAM development

# Co-Researchers

- University of Maryland MIND Lab

# Papers

- Barkmeyer, et al., *Concepts for Automating Systems Integration*, NIST IR 6928, (2003),  
[www.nist.gov/msidlibrary/doc/AMIS-Concepts.pdf](http://www.nist.gov/msidlibrary/doc/AMIS-Concepts.pdf)
- Denno, P., Steves, M., Libes, D., Barkmeyer, E., *Model-Driven Integration Using Legacy Models*, IEEE Software, (2003)
- Libes, et al., *The AMIS Approach to Systems Integration: an Overview*, NISTIR xxxx (2004), in publication