Design of Distributed Command and Control for Collaborative Situation Assessment

Presented at ICCRTS

Dr. Georgiy Levchuk (Aptima, Inc.)
Dr. Krishna Pattipati (UConn)

June 19, 2013

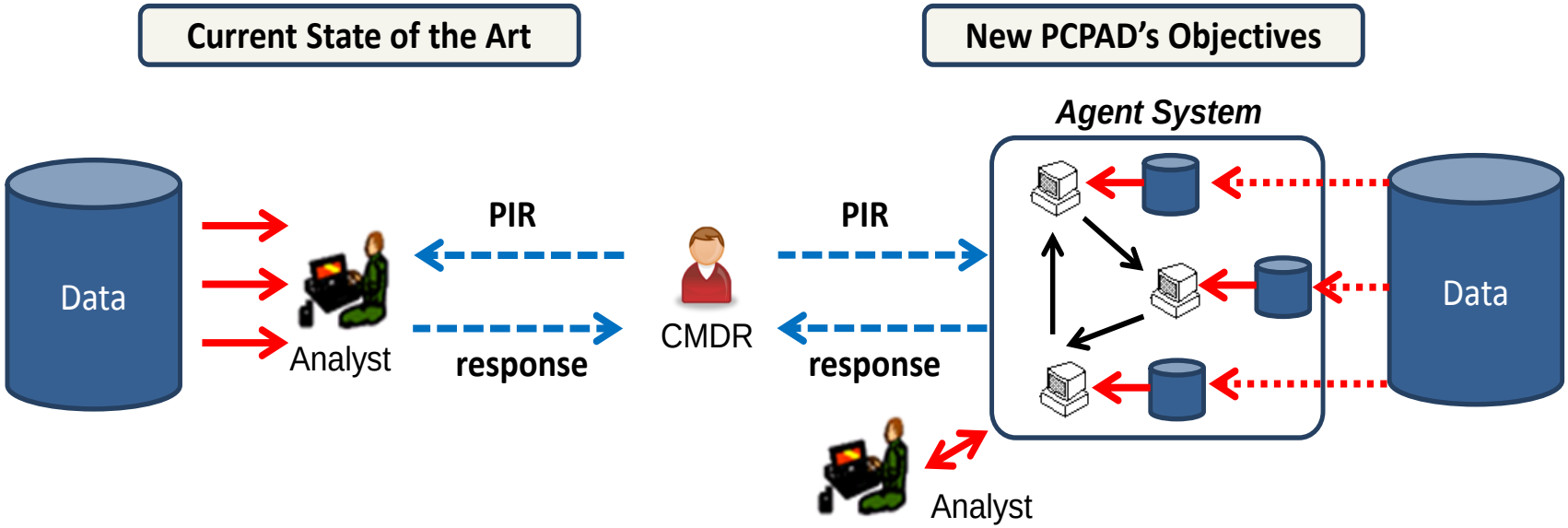
APTIMA®
Human-Centered Engineering

Summary

- Research objectives
  - Develop methods for analysis of large scale *relational* (interdependent) data for “denied environments”
- Example data analysis process
- Solution overview
  - Joint collaborative inference process
  - Agent C2 organization
- Example analysis results

Research Objectives

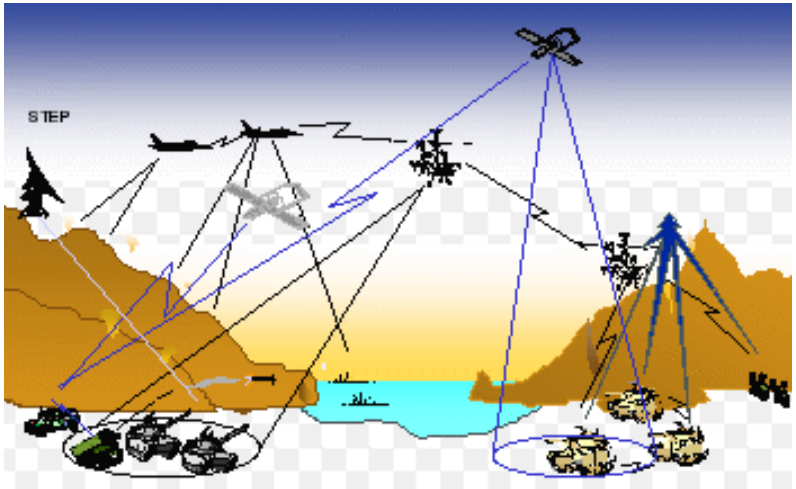
- Denied environment:
  - No persistent surveillance; no “cloud”, data is noisy, analysis resources are distributed and unreliable
- Implement large-scale data analysis supporting PCPAD process in a distributed and collaborative manner
  - Agents are assigned subsets of the PIR-based queries, analyze their own collected (or assigned) data, and collaborate during the search
- Robust to communication and agent/resource failures



Applications

- Robotics interactive cooperative control

- Multi-sensor management



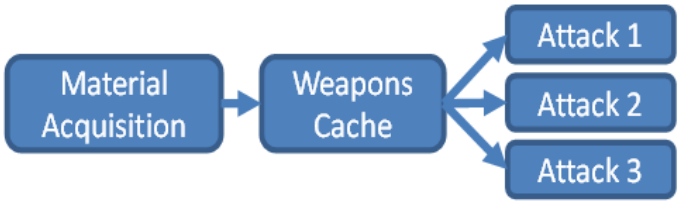
- Any search involving data from multiple sources and modalities

Example of PCPAD process-1

- Convert PIR into EEI network
  - A PIR is an abstract statement (Fig. a)
  - EEIs are explicit req-s, such as “What are locations of weapons caches, material acquisition activities, and attacks conducted by hostile military groups in AOI?”

“what tactical operations are conducted by adversaries within the AOI?”

(a) Priority Intelligence Requirement



(b) Model of enemy activity defines interdependent EEIs to address PIR

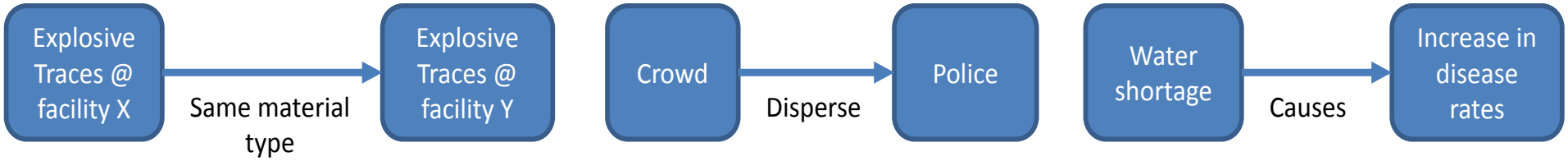
EEIs	Observation features / events	Sensor
Material Acquisition	Meetings, transportation, loading reported by locals	NLP processing over HUMINT, social media
Weapons Cache	Explosives residue, small arms stash, loading/offloading, guards	Explosives analysis team with residue tracers, Patrols, WAMI, UAV, imagery processing algorithms
Attack	Hostility (ambush, small arms fire, suicide bombing) reported in media and SIGACTS	NLP processing over SIGACTS; structured database queries

(c) Enemy activities can be observed via different events and sensors

Example of PCPAD process-2

- EEIs are interdependent
  - These dependencies complicate distribution of the data analysis

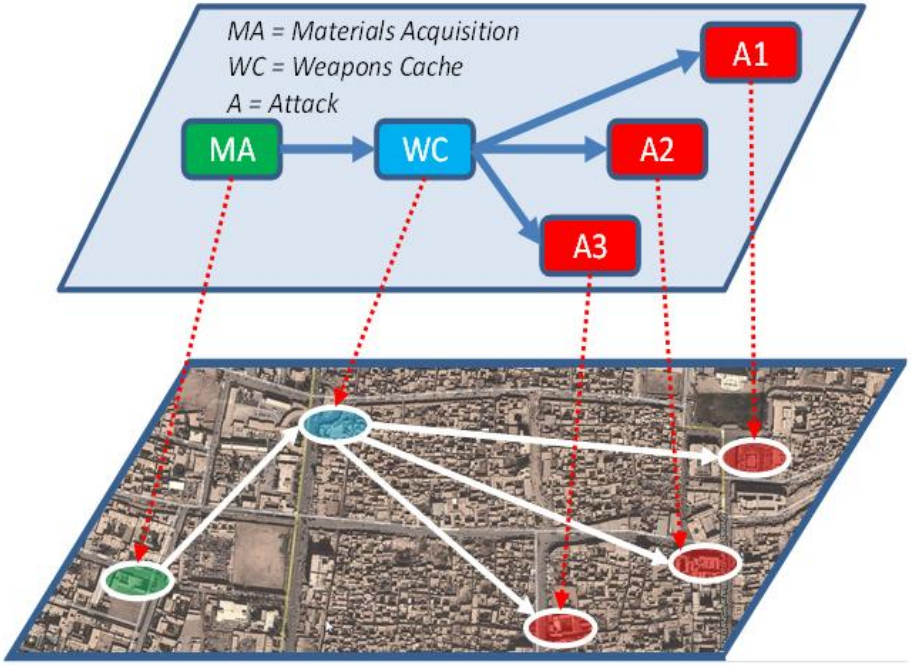
- Example dependencies
  - Defined as part of queries
  - May require additional processing to extract from data
  - Can “cross” subsets of data



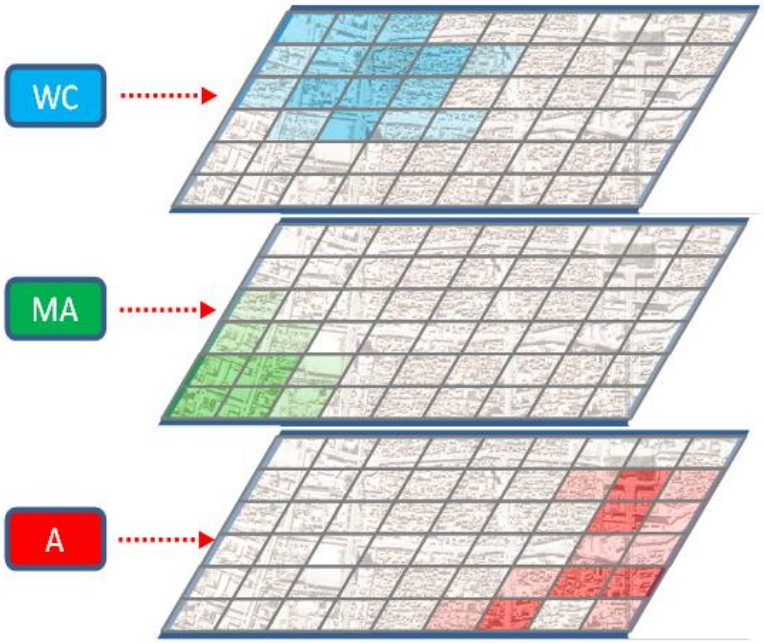
(a) Feature correlations **(b) Interactions** **(c) Influence**

Example of PCPAD process-3

- PIR response = EEI match in data
  - Fig. (d) describes an ideal outcome of PCPAD where only a single location of each activity is identified
  - Fig. (e) presents an example of approximate answer



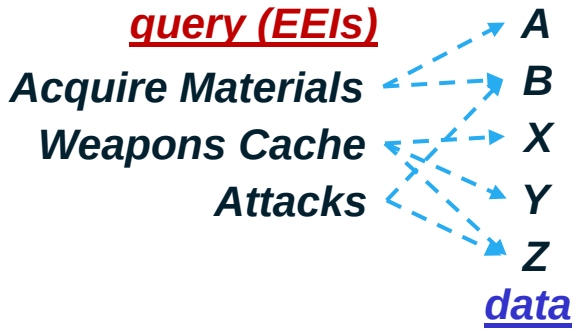
(d) Example of ideal output of PCPAD cycle

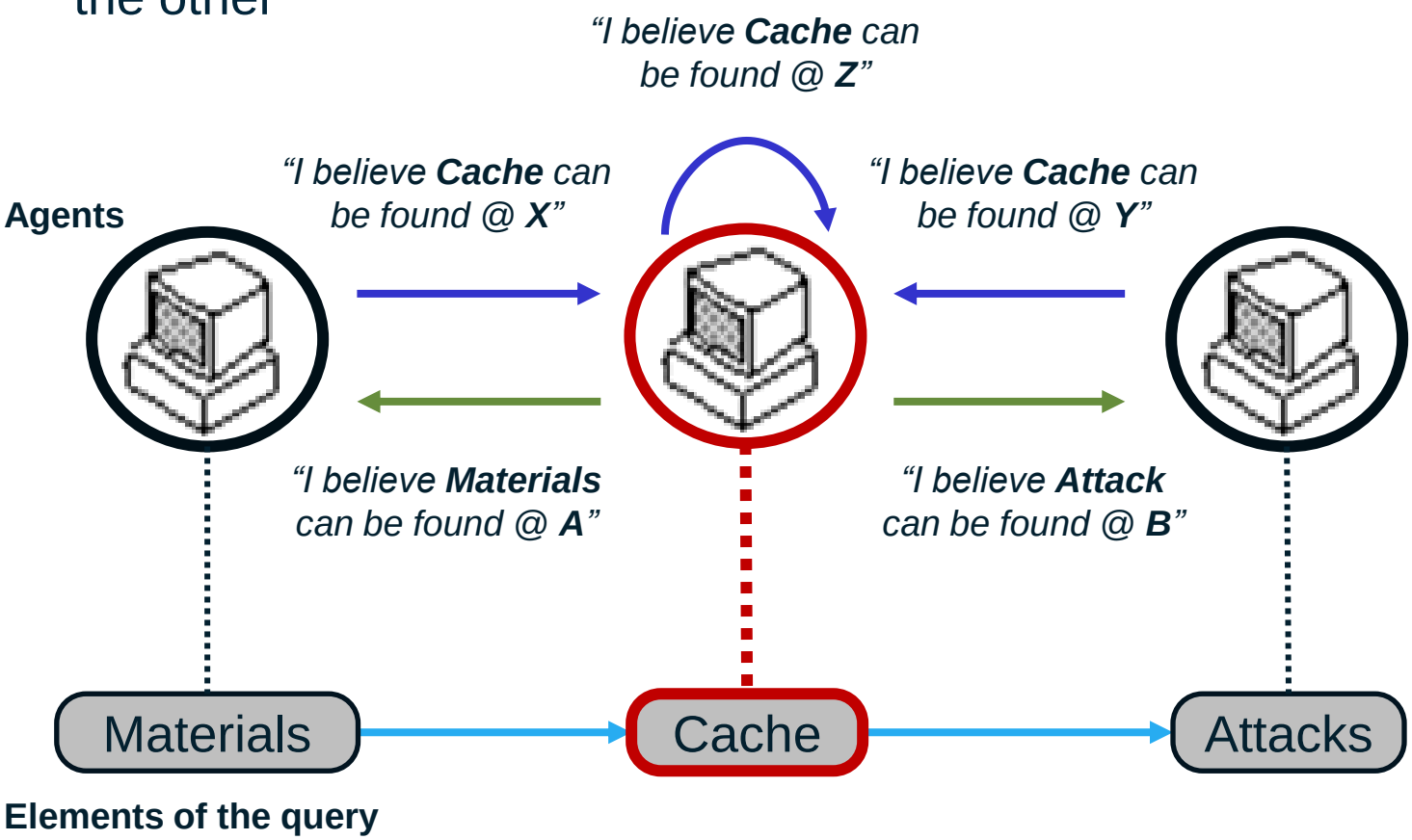


(e) Example of actual output of PCPAD cycle

Desired Process

- Query & data distributed among agents
- Agents collaborate via communicated messages
- Messages encode how one agent influences the other

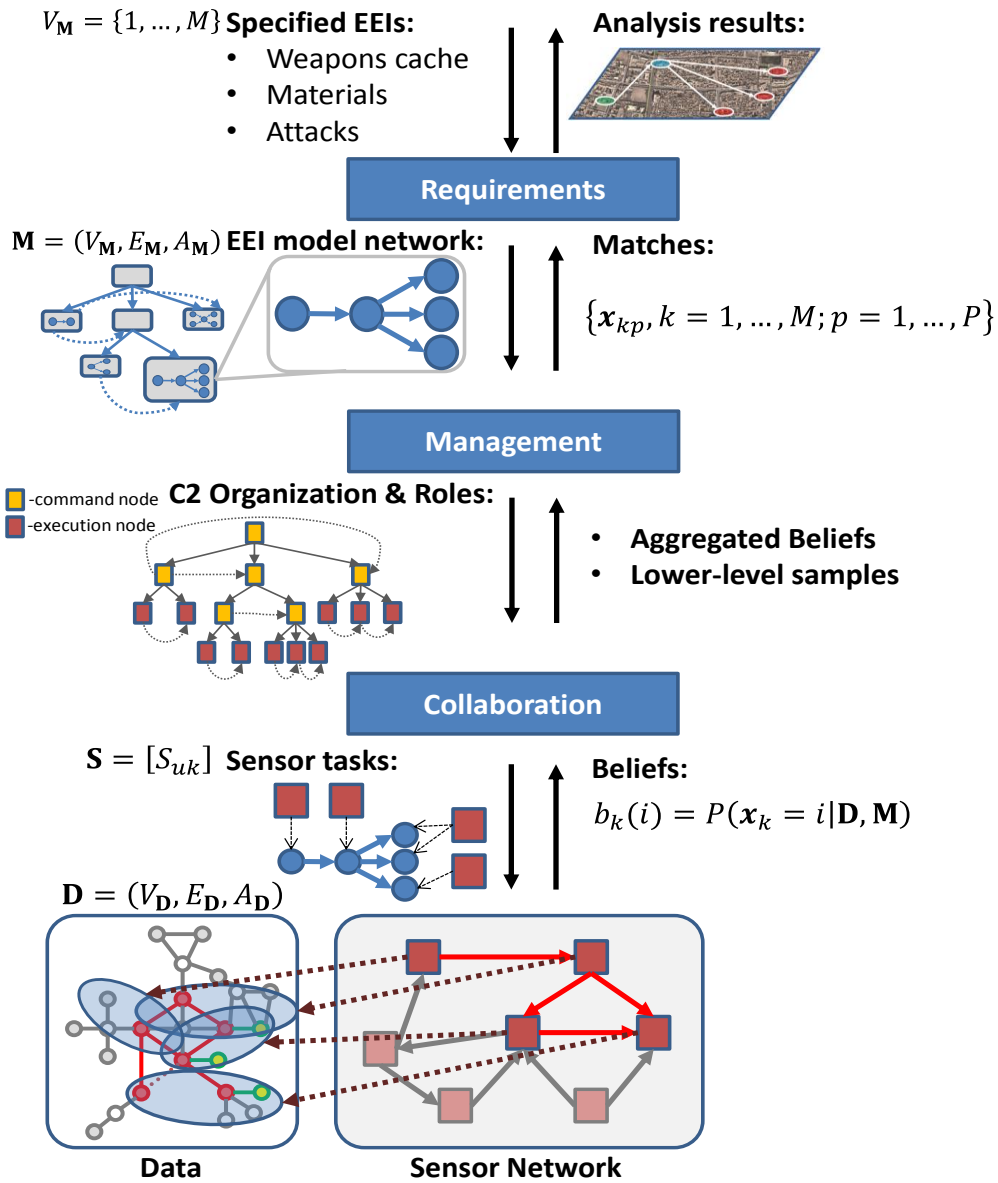




- Messages are influence mechanisms
- Can incorporate the trust into this framework by weighting the message impact

Solution Overview

- Search assets (agents) organized into C2 structure
  - Task assignment
  - Data assignment
  - Re-planning responsibilities
  - Communication requirements
- Collaboration based on dependencies among assigned tasks
  - Based on belief propagation model
  - Conforming to agent interaction framework (FIPA)



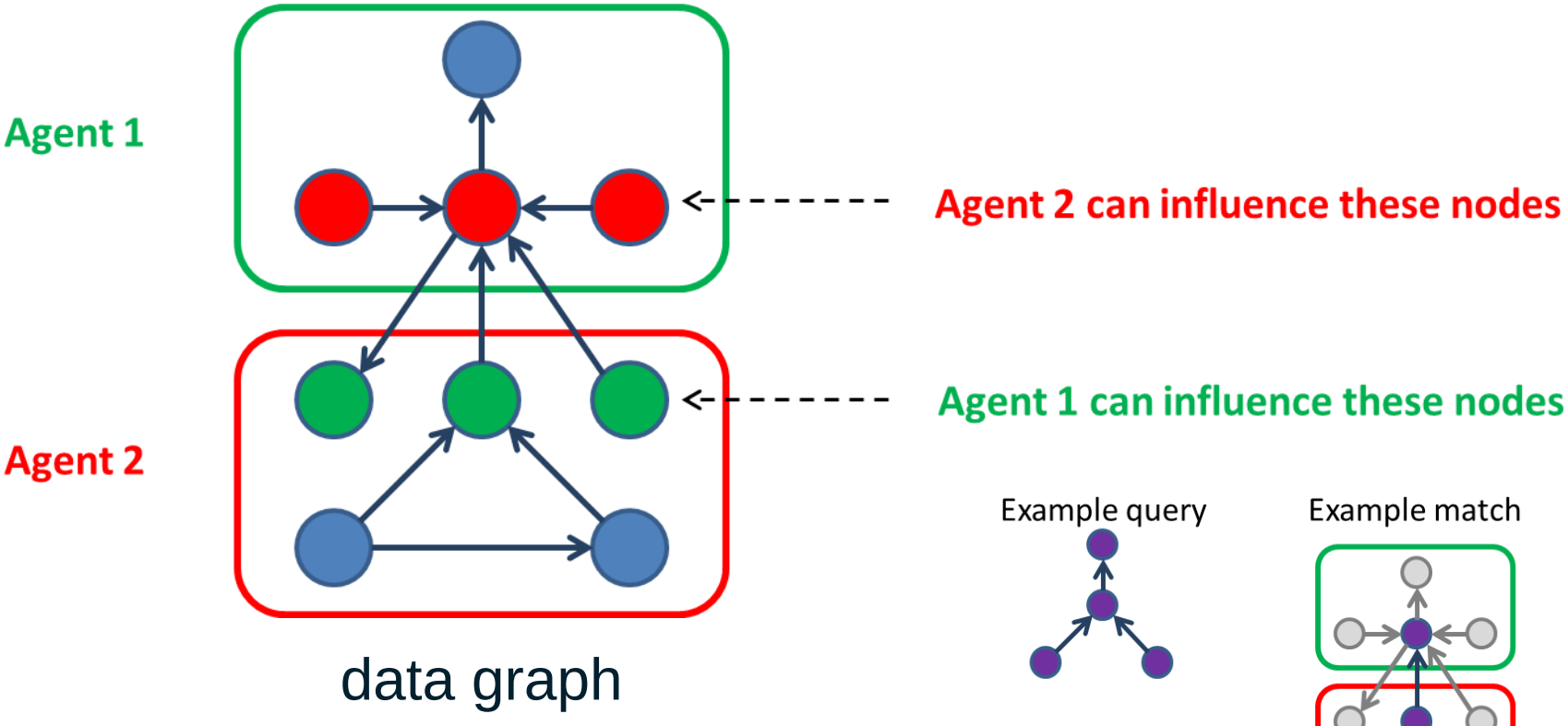
6-step Design Process

APTIMA
Human-Centered Engineering

- **Step 1: Assignment.** Initialize agents/sensor resources. Design C2 organization and assign EEIs (subqueries) to agents
- **Step 2: Observation.** The agents obtain access to (or collect) the data
- **Step 3: Mismatch calculations.** The agents compute the metric of mismatch between assigned EEIs & their dependencies and data nodes & links
- **Step 4: Local inference.** The agents compute inferences in the form of marginal posterior probability of association between query (EEI) nodes and data nodes. Agents incorporate received messages in this process
- **Step 5: Collaboration.** The agents compute the collaboration messages & send to other agents
- **Step 6: Joint inferences.** Final results are generated based on posterior probabilities provided by the agents to their supervisor

Why Should Agents Communicate?

- **Influence each other**
  - Over overlapping and neighboring data variables the agents have access to



Belief Propagation Model (BPM)

- a variant of DCOP (distributed coordination problem)

APTIMA
Human-Centered Engineering

Set of agents:

$$\boldsymbol{A} = \{A_1, A_2, \dots, A_k\}$$

Set of variables:

$$\boldsymbol{X} = \{x_1, x_2, \dots, x_n\}$$

Set of functions:

$$\boldsymbol{F} = \{F_1, F_2, \dots, F_m\}$$

$$F_i(\mathbf{x}_i),\ \mathbf{x}_i \in \boldsymbol{X}$$

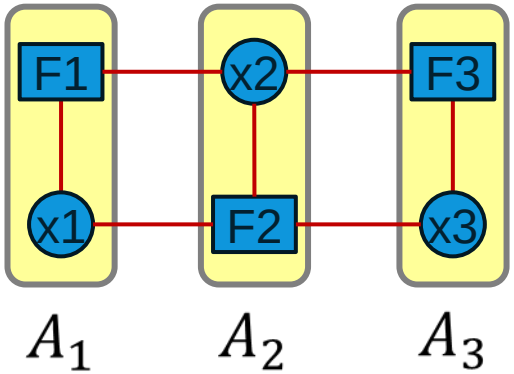
Objective function:

$$x^* = \arg\max \sum_i F_i(\mathbf{x}_i)$$

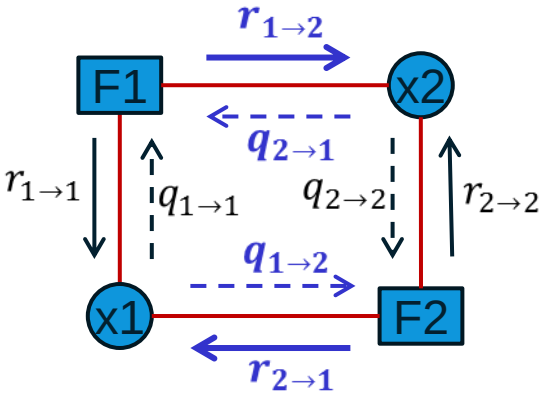
$$q_{i\to j}(x_i) \propto \sum_{k\in N_i\setminus j} r_{k\to j}(x_i)$$

$$r_{j\to i}(\mathbf{x}_i) \propto \max\left[F_j(\mathbf{x}_j) + \sum_{k\in N_j\setminus i} q_{k\to j}(x_k)\right]$$

Factor graph and allocation to agents:



Messages:



Questions:

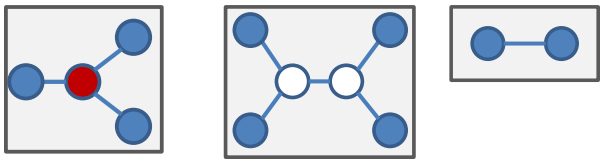
- What are our variables?
- What are messages?
- How to assign agents to variables?
- How to enable efficient dynamic coordination?

Model Details-1: Queries & Data

APTIMA
Human-Centered Engineering

- Query
  - EEIs = "query graph" $\mathbf{M} = (V_{\mathbf{M}}, E_{\mathbf{M}}, A_{\mathbf{M}})$
  - $V_{\mathbf{M}}$ are EEIs (nodes)
  - $E_{\mathbf{M}}$ are dependencies (links) between EEIs (distance, time, flow, similarity, etc.)
  - $A_{\mathbf{M}} = \{a_{km}^{\mathbf{M}}\}$ define expected observation attributes of EEIs $a_{kk}^{\mathbf{M}}$ and their relations $a_{km}^{\mathbf{M}}$
- Data
  - Attributed graph $\mathbf{D} = (V_{\mathbf{D}}, E_{\mathbf{D}}, A_{\mathbf{D}})$
  - $V_{\mathbf{D}} = \{1, \ldots, N\}$ (where $N \gg M$) are data variables (entities, locations, tracks, events, etc.)
  - $E_{\mathbf{D}}$ are observed relations between these variables
  - $A_{\mathbf{D}} = \{a_{ij}^{\mathbf{D}}\}$ define actually observed attributes of entities $a_{ii}^{\mathbf{D}}$ and relations $a_{ij}^{\mathbf{D}}$.

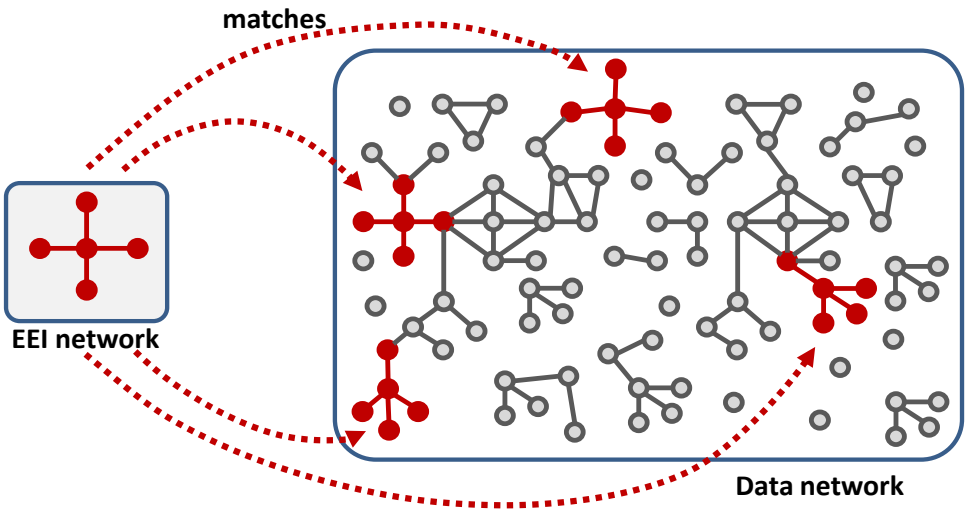
Queries/Patterns



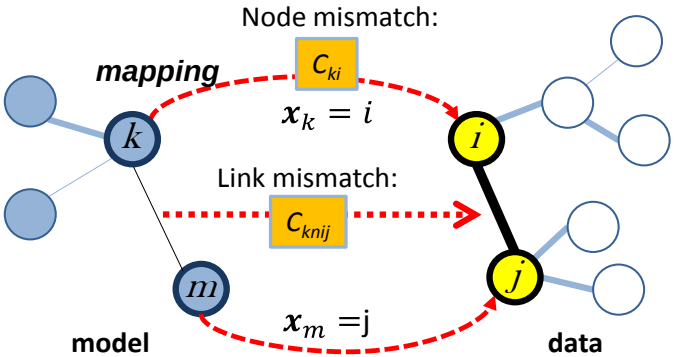
Data

Model Details-2: The Output

- Output of searching for the data supporting EEI $k \in V_{\mathbf{M}}$ is represented using set of mapping variables $\{\boldsymbol{x}_k \in V_{\mathbf{D}}\}$
- A single output (response to a PIR) is defined as a set of responses to all EEIs in the query $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$



(a) Generating responses to PIRs by finding subgraphs in data matching EEI network



(b) Variables of network matching

Model Details-3: Objective Function

APTIMA
Human-Centered Engineering

- The matching is evaluated based on probabilistic model, maximizing posterior probability of output

$$P(\boldsymbol{X} = \boldsymbol{match} | \mathbf{D} = \boldsymbol{data}, \mathbf{M} = \boldsymbol{PIR/query}) \cong$$
$$\frac{1}{Z}\prod_{k\in V_{\mathbf{M}}} P\left(a^{\mathbf{D}}_{\boldsymbol{x}_k \boldsymbol{x}_k} | a^{\mathbf{M}}_{kk}\right) \prod_{(k,m)\in E_{\mathbf{M}}} P\left(a^{\mathbf{D}}_{\boldsymbol{x}_k \boldsymbol{x}_m} | a^{\mathbf{M}}_{km}\right) = \frac{1}{Z}\prod_{k\in V_{\mathbf{M}}} \boldsymbol{e}^{-C_{k\boldsymbol{x}_k}} \prod_{(k,m)\in E_{\mathbf{M}}} \boldsymbol{e}^{-C_{k,m;\boldsymbol{x}_k,\boldsymbol{x}_m}}$$

- Equivalent to Min attribute mismatch between query and mapped data subgraph:

$$-\log P(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M | \mathbf{D}, \mathbf{M}) \cong Q(X) = \sum_{k\in V_{\mathbf{M}}} \underline{C_{k\boldsymbol{x}_k}} + \sum_{(k,m)\in E_{\mathbf{M}}} \underline{C_{k,m;\boldsymbol{x}_k,\boldsymbol{x}_m}}$$

Node mismatch **Link mismatch**

Computed by specific raw data processing algorithms

- $C_{ki} = -\log P\left(a^{\mathbf{D}}_{ii} | a^{\mathbf{M}}_{kk}\right)$ is a measure of mismatch between attributes of EEI $k \in V_{\mathbf{M}}$ and data variable $i \in V_{\mathbf{D}}$
- $C_{km;ij} = -\log P\left(a^{\mathbf{D}}_{ij} | a^{\mathbf{M}}_{km}\right)$ is a measure of mismatch between EEI relation $(k,m) \in E_{\mathbf{M}}$ and data relation $(i,j) \in E_{\mathbf{D}}$
- For Gaussian noise modeling, these mismatch measures are computed as L2 norm: $C_{ki} = \left\|a^{\mathbf{D}}_{ii} - a^{\mathbf{M}}_{kk}\right\|$, $C_{km;ij} = \left\|a^{\mathbf{D}}_{ij} - a^{\mathbf{M}}_{km}\right\|$.

Centralized Solution-A

APTIMA
Human-Centered Engineering

- Belief propagation:
  - Find marginal probabilities for which the joint posterior probability is maximized:

$$b_k(i) = P(\boldsymbol{x}_k = i | \mathbf{D}, \mathbf{M})$$

  - Marginal probability vector $b_k = [b_k(i), i \in V_{\mathbf{D}}]$ represents a belief, or a distribution, about location of EEI $k \in V_{\mathbf{M}}$ in the observed data
  - Incrementally update the estimates of $b_k(i)$
  - Compute log-probabilities for ease of computations:

$$\mu_k(i) = \log b_k(i)$$

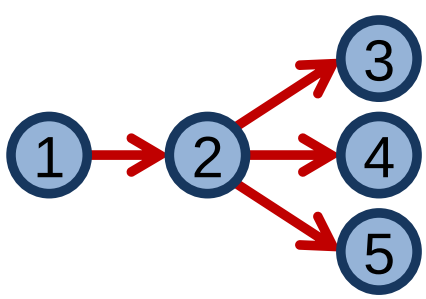
Centralized Solution-B

- Factor graph model
  - Variable nodes correspond to EEIs $k \in V_{\mathbf{M}}$; maintain and update messages (log of marginal beliefs) and send these to factor nodes:

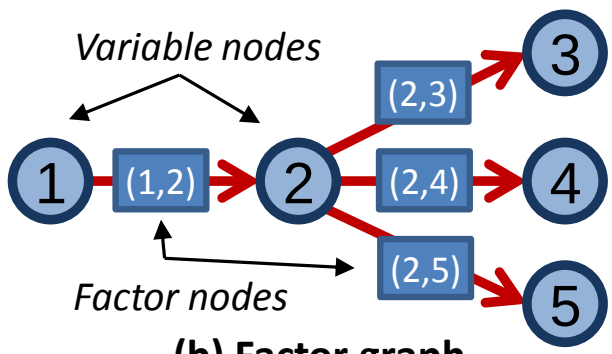
$$\mu_k = [\mu_k(i), i \in V_{\mathbf{D}}]$$

  - Factor nodes are defined for each link $(k, m) \in E_{\mathbf{M}}$ between EEIs; these nodes maintain and update two factor messages, representing the marginal log-probabilities of matching model link (m, k) to the data link that ends or starts in node j:

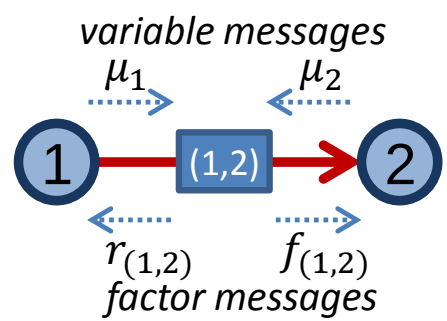
$$f_{(m,k)} = [f_{(m,k)}(j), j \in V_{\mathbf{D}}]$$
$$r_{(m,k)} = [r_{(m,k)}(j), j \in V_{\mathbf{D}}]$$



(a) Pattern (query) network



(b) Factor graph



(c) Subset of messages for original message passing formulation

Message Computations

APTIMA
Human-Centered Engineering

- Updates at variable nodes (*summation*):

$$\mu_m(i) \propto -C_{mi} + \sum_{\substack{l: \\ (l,m)\in E_{\mathbf{M}}}} f_{(l,m)}(i) + \sum_{\substack{l: \\ (m,l)\in E_{\mathbf{M}}}} r_{(m,l)\to m}(i)$$

- Updates at factor nodes (*maximization*):

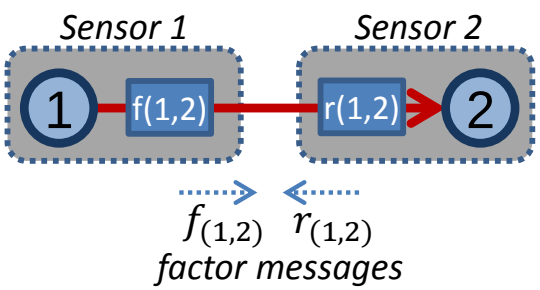
$$f_{(m,k)}(j) \propto \max_{\boldsymbol{i}} \left(-C_{mk;ij} + \mu_m(i) - r_{(m,k)}(i)\right)$$

$$r_{(m,k)}(j) \propto \max_{\boldsymbol{i}} \left(-C_{mk;ji} + \mu_k(i) - f_{(m,k)}(i)\right)$$

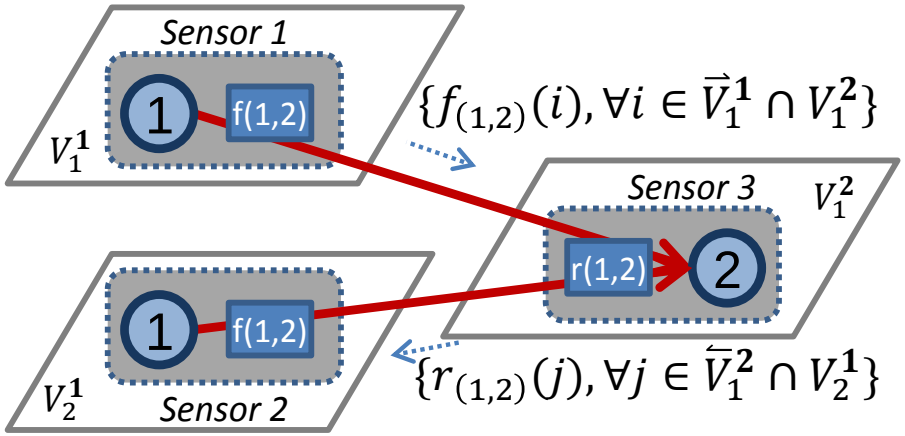
- Computational complexity
  - # iterations ~ diameter of query graph
  - # operations per iteration: $O(\max\{|V_{\mathbf{M}}|, |E_{\mathbf{M}}|\} \times \max\{|V_{\mathbf{D}}|, |E_{\mathbf{D}}|\})$

Distributed Implementation

- Can distribute to parallel processing system
  - Variable nodes = EEI inferences = assigned to analysis resource(s)
  - Factor nodes = EEI dependencies = assigned to resources with responsibility for **"from"** node
- Explicitly encodes
  - **Internal analysis:** find matches to assigned EEIs
  - **Collaboration messages:** factor messages
  - **Fusion** of external messages and internal analysis via BP updates



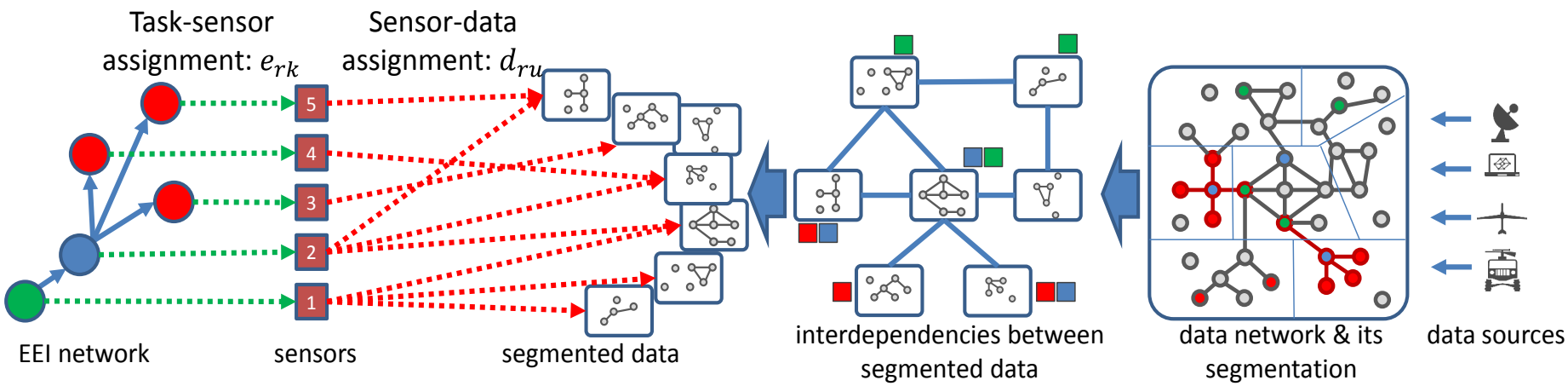
(a) Subset of messages for baseline distributed SLBP



(b) Subset of "forward" messages for extending distributed SLBP employing data decomposition

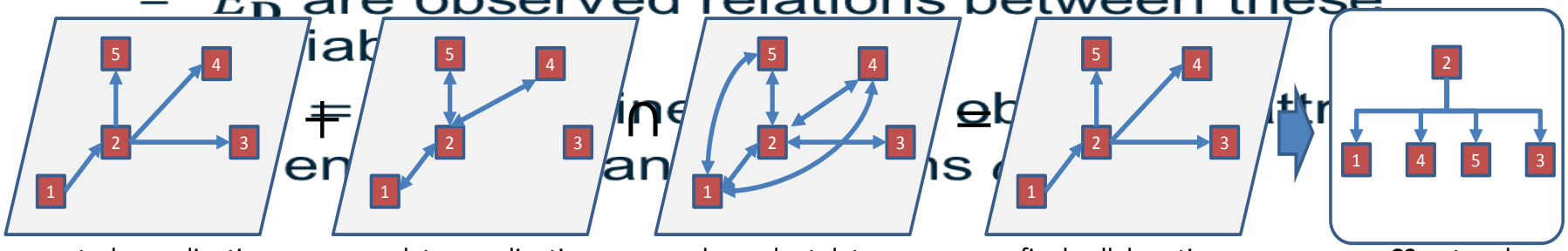
C2 Design Model-1

- *Data assignment* variable $d_{ru} = 1$, if sensor agent $r \in V_{\mathbf{R}}$ is allocated to process data subset $V_{\mathbf{D},u}$
- *EEI assignment* variable $e_{rk} = 1$, if sensor agent $r \in V_{\mathbf{R}}$ is allocated to respond to EEI $k \in V_{\mathbf{M}}$



C2 Design Model-2

- Query
  - EEIs = "query graph" $\mathbf{M} = (V_{\mathbf{M}}, E_{\mathbf{M}}, A_{\mathbf{M}})$
  - $V_{\mathbf{M}}$ are EEIs (nodes)
  - $E_{\mathbf{M}}$ are dependencies (links) between EEIs (distance, time, flow, similarity, etc.)
  - $A_{\mathbf{M}} = \{a^{\mathbf{M}}_{km}\}$ define expected observation attributes of EEIs $a^{\mathbf{M}}_{kk}$ and their relations $a^{\mathbf{M}}_{km}$
- Data
  - Attributed graph $\mathbf{D} = (V_{\mathbf{D}}, E_{\mathbf{D}}, A_{\mathbf{D}})$
  - $V_{\mathbf{D}} = \{1, \ldots, N\}$ (where $N \gg M$) are data variables (entities, locations, tracks, events, etc.)
  - $E_{\mathbf{D}}$ are observed relations between these



sensor-task coordination requirements | same data coordination requirements | dependent data coordination requirements | final collaboration network | C2 network

C2 Design Model-3

- Need several variables to define C2 organization, including estimates of:
  - Error of the agent's analytics
  - Computation workload
  - Communication workload
- Any error estimations require "fast" estimate of value of data in relation to EEI in the query
- Workload estimations require hypothesizing compressions that could be achieved

C2 Design Model-4

- Objectives of C2 design
  - minimize analysis errors $\rho[r]$,
  - balance the analysis workloads among agents $w^A[r]$, and
  - reduce the communication requirements penalty $p^C[r,l]$
- Overall **cost of task-agent assignment**

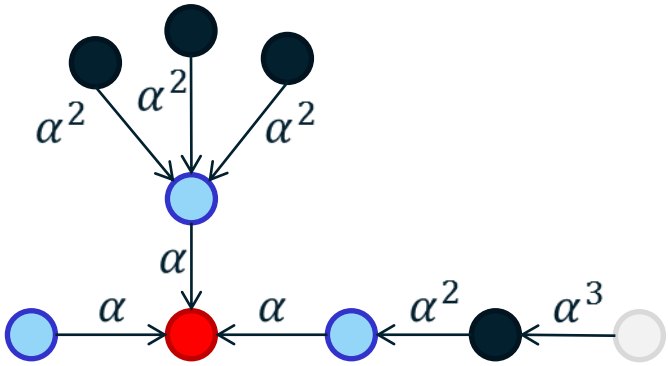
$$cost(\{e_{rm}\}) = \omega^{\rho} \underbrace{\sum_{r,m} e_{rm} \cdot \rho_m[r]}_{analysis\ error} + \omega^{A} \underbrace{\left(\sum_{r,m} e_{rm} \cdot w_m^A[r]\right)^2}_{analysis\ workload\ balance} +$$

$$\omega^{C} \underbrace{\sum_{\substack{r,l \\ (m,k)\in E_{\mathbf{M}}}} e_{rm} \cdot e_{lk} \cdot \frac{w^M(r,l,m,k)}{b[r,l]-w^M(r,l,m,k)}}_{communication\ penalty}$$

weights $\omega^{\rho}, \omega^{A}, \omega^{C}$ correspond to relative importance (trade-off) between analysis error, workload balance, and communication penalty

Variables: Analysis Error

- Approximating the "value" (relevance) of data using information propagation model:
  - For each data node, compute $\tilde{a}_{ii}^{\mathbf{D}} = I(a_{ii}^{\mathbf{D}}) + \sum_{l=1}^{h} \alpha^l \sum_{g(i,j)=l} I(a_{jj}^{\mathbf{D}})$, where α is a weight term, $I(a)$ is an indicator vector with values = 1 when input vector values are non-zero, $g(i,j)$ defines a number of links on a shortest path between i and j, and h is a neighborhood radius used to compute $\tilde{a}_{ii}^{\mathbf{D}}$.
  - For each EEI, compute $\tilde{a}_{mm}^{\mathbf{M}}$ in a query graph similar to above
  - For each EEI node $m \in V_{\mathbf{M}}$, and the data nodes $i \in V_{\mathbf{D}}$, we compute the propagation mismatches: $\tilde{C}_{mi} = \sum_u \theta(\tilde{a}_{ii}^{\mathbf{D}}[u], \tilde{a}_{mm}^{\mathbf{M}}[u])$.
    - Here, the function $\theta(a,b)$ is a positive difference function ($\theta(a,b) = a - b$, if $a > b$, and 0 otherwise)
  - Analysis error (inverse of efficiency) of performing an analysis task for EEI $m \in V_{\mathbf{M}}$ at agent $r \in V_{\mathbf{R}}$: $\rho_m[r] = \min_{i \in V_{\mathbf{D}(r)}} \tilde{C}_{mi}$
  - Overall agent's error: $\rho[r] = \sum_m e_{rm} \cdot \rho_m[r]$



Variables: Workload

APTIMA
Human-Centered Engineering

- Workload estimates:
  - # computations resulting from assigning task m to agent r: $w_m^A[r] = degree(m) \times \left(\left|V_{\mathbf{D}(r)}\right| + \left|E_{\mathbf{D}(r)}\right|\right)$, where $degree(m)$ is the degree of the node, i.e. the number of incoming and outgoing links in query graph for node m
  - the **analysis workload** $w^A[r]$ of the agent r is computed as the total number of computations at this agent: $w^A[r] = \sum_{m \in V_{\mathbf{M}}} e_{rm} \cdot w_m^A[r] = \left(\left|V_{\mathbf{D}(r)}\right| + \left|E_{\mathbf{D}(r)}\right|\right) \times \sum_{m \in V_{\mathbf{M}}} e_{rm} \cdot degree(m)$

Variables: Communication Load

APTIMA
Human-Centered Engineering

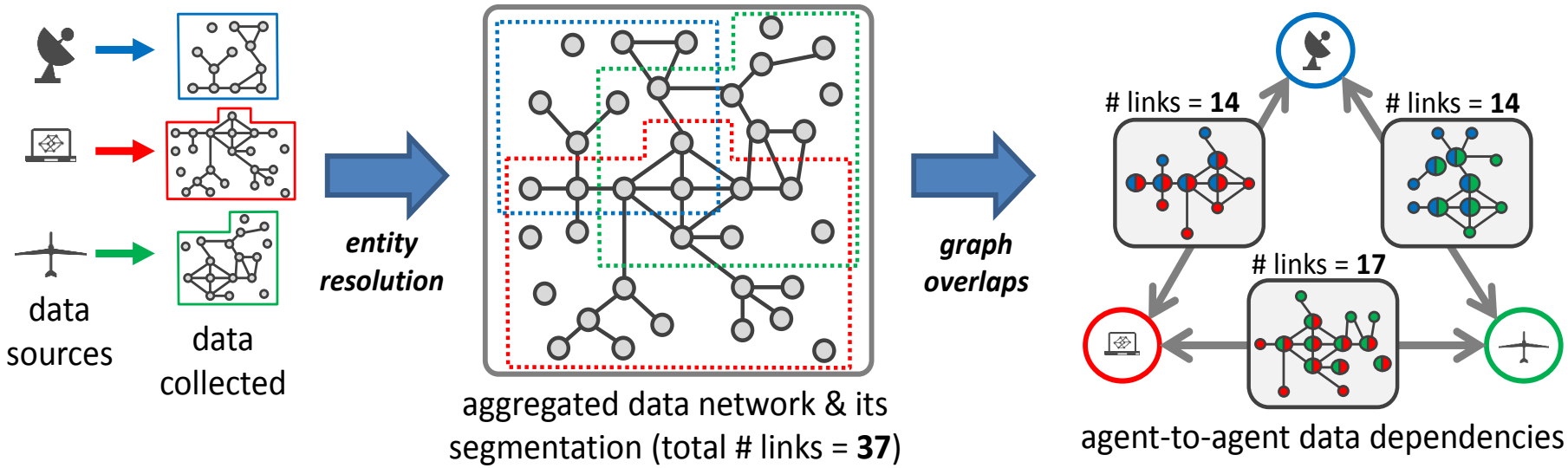
- Communication estimates:
  - **# messages communicated between any two agents** analyzing two interdependent EEIs can be computed based on the number of links in *data dependency graph*, which is an overlap of the data subgraphs accessible by two agents r and l: $o(r,l) = \left|\vec{V}_{\mathbf{D}(r)} \cap V_{\mathbf{D}(l)}\right| + \left|\overleftarrow{V}_{\mathbf{D}(l)} \cap V_{\mathbf{D}(r)}\right|$
  - estimate the **weight of the compressed message** passed from agent r to l due to the EEI link (m,k) using the entropy of the propagation attribute mismatch:

$$w^M(r,l,m,k) = o(r,l)\left(\begin{array}{l} \sum_{i \in V_{\mathbf{D}(l)} \cap \left(V_{\mathbf{D}(r)} \cup \overleftarrow{V}_{\mathbf{D}(r)}\right)} \dfrac{\left(\vec{C}_{mk;i} + \log \sum_j e^{-\overleftarrow{C}_{mk;i}}\right) e^{-\overleftarrow{C}_{mk;i}}}{\sum_j e^{-\overleftarrow{C}_{mk;i}}} \\ + \sum_{i \in V_{\mathbf{D}(l)} \cap \left(V_{\mathbf{D}(r)} \cup \overleftarrow{V}_{\mathbf{D}(r)}\right)} \dfrac{\left(\vec{C}_{mk;i} + \log \sum_j e^{-\overleftarrow{C}_{mk;i}}\right) e^{-\overleftarrow{C}_{mk;i}}}{\sum_j e^{-\overleftarrow{C}_{mk;i}}} \end{array}\right)$$

    , where we define $\overleftarrow{C}_{mk;j} = \min_{i:\,(i,j) \in E_{\mathbf{D}}} \tilde{C}_{mk;ij}$, $\vec{C}_{mk;i} = \min_{j:\,(i,j) \in E_{\mathbf{D}}} \tilde{C}_{mk;ij}$, $\tilde{C}_{mk;ij} = \sum_u \theta\left(\tilde{a}^{\mathbf{D}}_{ij}[u], \tilde{a}^{\mathbf{M}}_{mk}[u]\right)$, and attributes $\tilde{a}^{\mathbf{D}}_{ij}, \tilde{a}^{\mathbf{M}}_{mk}$ are link-based information propagation attributes
  - the **communication requirements** $w^C[r,l]$ from agent r to agent l, are computed as: $w^C[r,l] = \sum_{(m,k) \in E_{\mathbf{M}}} e_{rm} \cdot e_{lk} \cdot w^M(r,l,m,k)$
  - the **communication penalty** per agent-to-agent link (r,l) $p^C[r,l] = \sum_{(m,k) \in E_{\mathbf{M}}} e_{rm} \cdot e_{lk} \cdot \frac{w^M(r,l,m,k)}{b[r,l] - w^M(r,l,m,k)}$

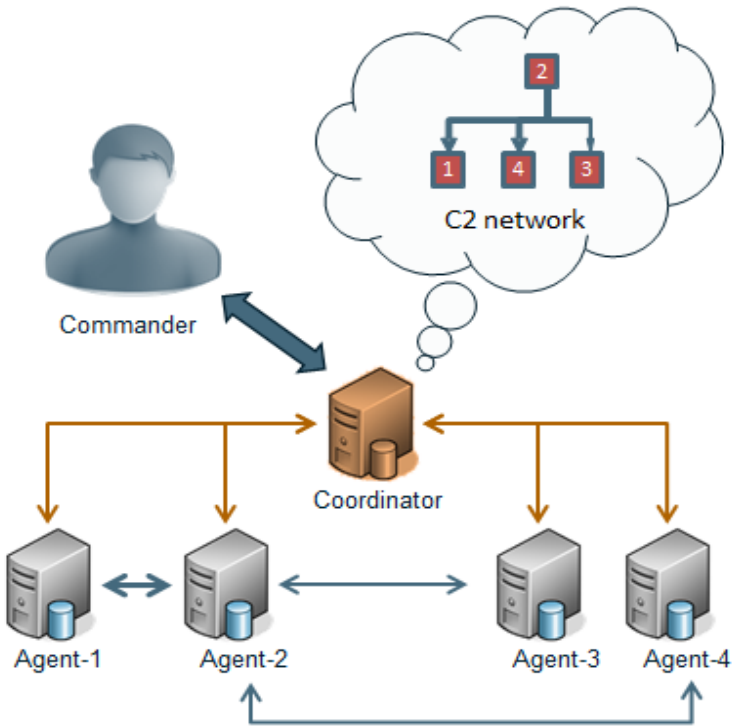
Example of C2 Agent Dependencies

- In general the agent-to-agent dependencies could be quite complex, and in the worst-case composed of a very large number of variables which are infeasible to communicate in contested environments
- **Objectives**: balance workload, maximize search accuracy, make efficient communication

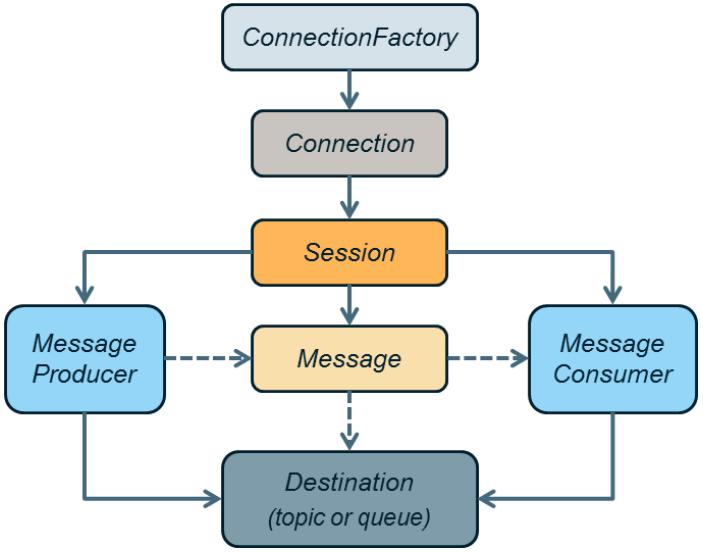


Implementation-1

- Java Messaging Service
  - Currently using Apache ActiveMQ - in-memory JMS provider



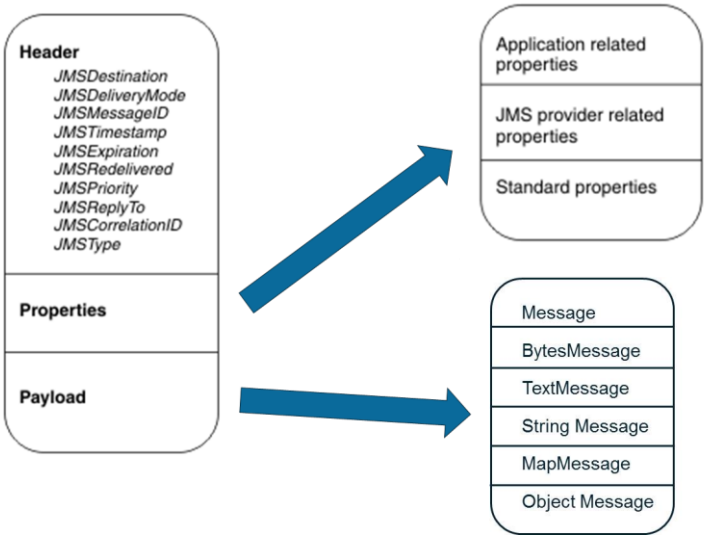
(a) Architecture



(b) JMS functional components

Implementation-2

- Message structure:



(a) Message structure

Property Name	type	default value	description
JMSDestination	javax.jms.Destination	set by the producer	Destination used by the producer
JMSReplyTo	javax.jms.Destination	null	user defined
JMSType	String	empty	user defined
JMSDeliveryMode	int	DeliveryMode.PERSISTENT	indicator if messages should be persisted
JMSPriority	int	4	value from 0-9
JMSMessageID	String	unique	unique identifier for the message
JMSTimestamp	long	time message sent	time in milliseconds
JMSCorrelationID	String	null	user defined
JMSExpiration	long	0	time in milliseconds to expire the message - 0 means never expire
JMSRedelivered	boolean	false	true if the message is being resent to the consumer

(b) Field/header properties

Message: The base class. This message type is used for event notification, and does not have a payload.

BytesMessage:The payload is stored as an array of bytes.

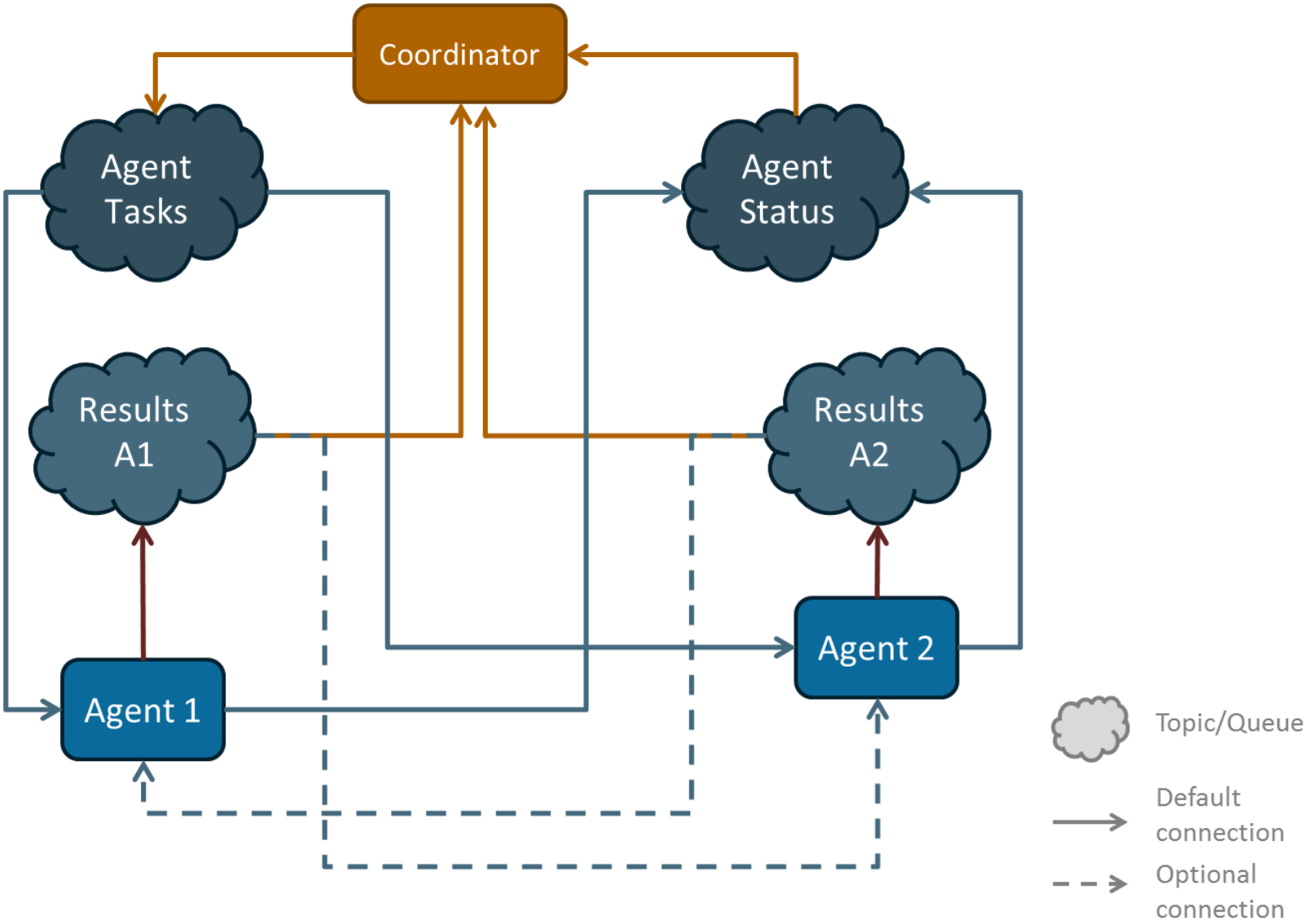
TextMessage: Data is stored as a string.

StreamMessage: A Stream message is a sequence of primitive Java types.

MapMessage: The payload of a MapMessage is stored as a set of name-value pairs. The name is defined as a string and the value is typed. The MapMessage is useful for delivering keyed data that can change from one message to the next.

ObjectMessage: The Object message carries a serializable Object as its payload. It is useful for exchanging objects.

Implementation-3



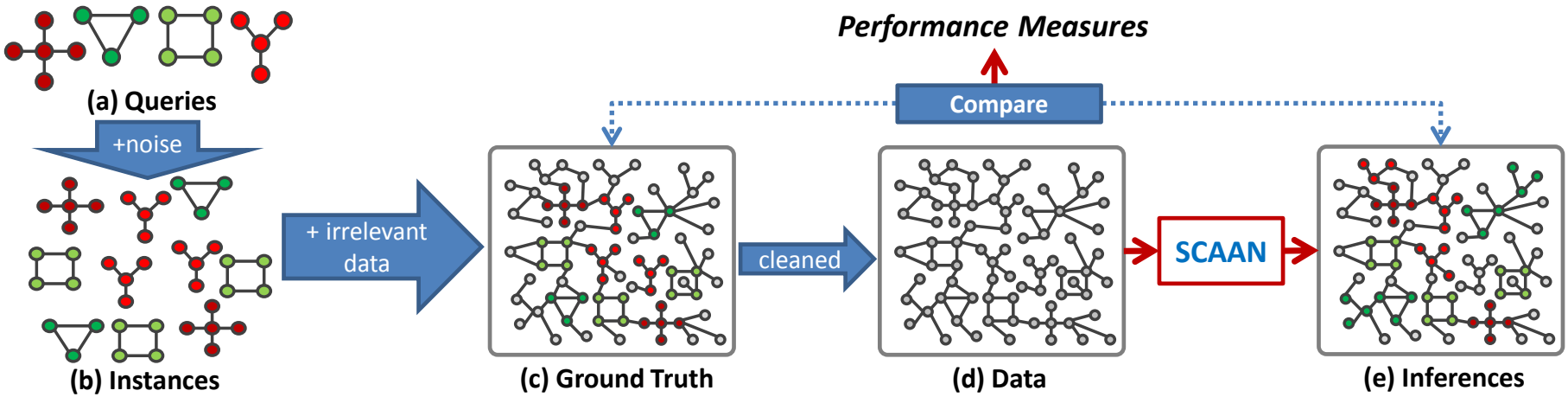
Experimental Design & Hypothesis

APTIMA
Human-Centered Engineering

- Compare:
  - Distributed vs centralized implementation
  - Analyze performance at different noise levels

- Hypotheses:
  - Distributed implementation is close in accuracy to centralized solution
  - Distributed implementation enables linear reduction of processing time with increase of # of available agents
  - Distributed implementation can achieve higher solution and better load distribution in “heterogeneous” setting

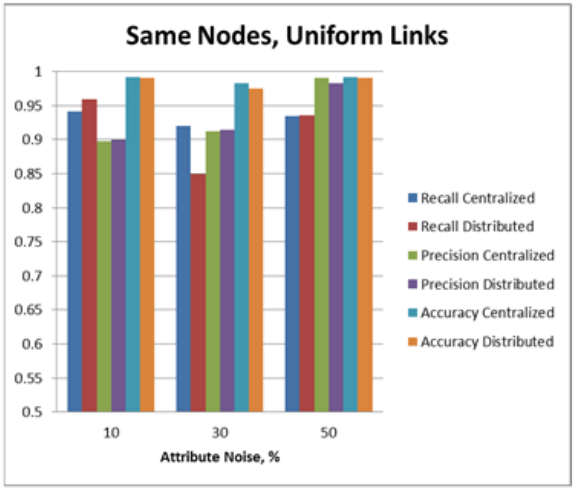
Experimental Setup

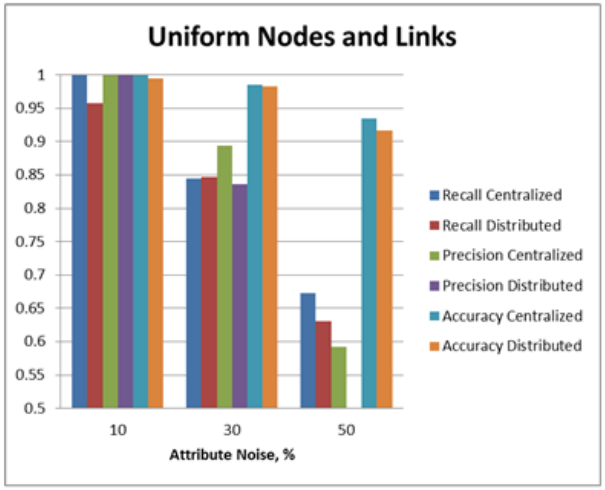


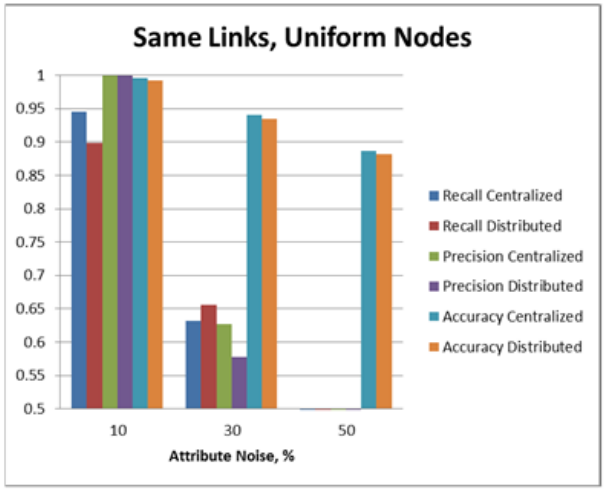
- 100 Monte-Carlo runs for three configuration of data node and relation ambiguity:
  - nodes are ambiguous (nodes have the same attribute values) while links have uniformly generated attributes
  - both nodes and links have uniformly generated attributes
  - links are ambiguous (links have the same attribute values) while nodes have uniformly generated attributes, and

Results

- Inference outcomes:



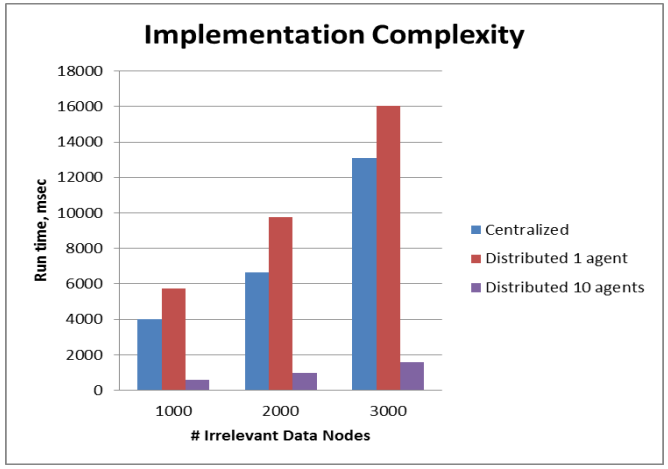




Nodes ambiguous → **Both varied** → **Links ambiguous**

Increase in relational ambiguity

- Computational time



Conclusions & Future Research

APTIMA
Human-Centered Engineering

- Many data analysis models, ranging from complex database queries to knowledge retrieval, employ graphical algorithms to perform joint inference and reasoning
- To achieve optimal scalability and accuracy, in “denied environments”, distributed data analysis solutions must incorporate collaborative data processing by a set of intelligent agents
- Such distribution is challenging due to uncertainties in workload and performance estimation, causing potential bottlenecks in synchronous collaborative data analysis process
- Our solution shows feasibility of decentralized data search
- Current research is focused on scalability and accuracy improvements for distributed implementation
  - Compression of messages, indexing graph data, prioritization of information access by agents

APTIMA®

Human-Centered Engineering