RDF graph synchronization for collaborative robotics - GAIA

ADAM FREJ

PATRYK TOMASZEWSKI

Goal of the system

Synchronise knowledge graphs between multiple agents in real-time, while each agent continuously appends new data, and while also handling issues typically present in wireless communication.

By:

Adding knowledge graph versioning

Implementing merging mechanism

Designing and implementing communication protocols for graph synchronisation

Our application

One role: RDFAgent

One subrole: MergeMasterAgent

Three interactions:

- Status
- Revision
- Revision-request

Environment

There exists a full knowledge graph of "truth at current time". This graph may change (addition/removal of triples) through time.

Each agent may, at some irregular interval, retrieves a triple (or information about removal of a triple) which is true at a given point of time. The same triple may be retrieved multiple times by different agents.

RDFAgent role

Role: RDFAgent

Description: The only agent type in the system. Stores internal graph, adds revisions and synchronizes them with other agents.

Protocols and Activities: HandleLocalRevisions, HandleExternalRevisions, HandleStatus, RequestExternalRevision, HandleRevisionRequest

Permissions: communicate with any agent, read and write local graph

MergeMasterAgent - subrole

Role: RDFAgent - MergeMaster

Description: A special subrole which any agent can perform. In every group one agent acts as merge master. It extends RDFAgent's responsibilities by performing merging in HandleExternalRevisions protocol.

Additional Protocols and Activities: HandleExternalRevisions – performs merging

Additional Permissions: perform and publish revisions merge

Interactions model

Protocol	Status	Revision	Revision-request
Initiator(s)	RDFAgent	RDFAgent	RDFAgent
Receiver(s)	All available agents	All grouped agents	RDFAgent
Responding Action	HandleStatus	HandleExternalRevisions	HandleRevisionRequest
Purpose / Parameters	Detect all available agents in the area (and form a group). Choose merge master. Determine other agents' tip revisions.	Propagate a given revision to other agents, which incorporate it into their local graphs.	Request a missing revision from a specific agent and complete a local graph.

Service model

Service	HandleLocalRevisions	HandleExternalRevisions	HandleStatus	RequestExternalRevision	HandleRevisionRequest
Inputs	Local graph	External revision	External status	Status | parent revisions	Missing revision metadata
Outputs	New revision	Combined revision	Internal status	Missing revision metadata	Revision
Pre-condition	The RDFAgent is initialized	The RDFAgent is initialized and connected to other agents	The RDFAgent is initialized	The RDFAgent is initialized and connected to other agents	The RDFAgent is initialized and connected to other agents
Post-condition	Random intervals	The RDFAgent received external revision	Periodic intervals (send status) | [receive status]	The RDFAgent detects another agent with unknown revisions	The RDFAgent received revision request

Service model

Service	Merge	Rebase
Inputs	2 new concurrent revisions	New local revision
Outputs	Combined revision	Combined revision
Pre-condition	The RDFAgent is initialized and acts as merge master	The RDFAgent is initialized and connected to other agents
Post-condition	The agent received revisions to merge	The agent created local revision before merging a new external revision

Acquaintance model

All connected (able to communicate) agents form a group

All grouped agents communicate (send revisions to synchronize a single graph)

Each group assigns one merge master

All group members know the merge master (and vice versa)

	RDFAgent	MergeMasterAgent
RDFAgent	I	I, A
MergeMasterAgent	I, A	-

Thank you for attention

Feel free to ask questions