

REA to relational model mapping rules

Algorithm 1 Agent Mapping Rules

```
1: for all Super Agents do
2:   create Agent Type Table with the name <Agent.Name>Type;
3:   - add column TypeName of type VARCHAR and make it the primary key;
4:   - add all Type Properties as columns with the specified type;
5:   - insert the name of each related Sub Agent into the table;
6:   create Agent Table with the name <Agent.Name>;
7:   - add all Object Properties as columns with the specified type;
8:   - make the primary key object property the primary key column;
9:   - create foreign key to the Agent Type Table;
10: end for
11: for all Sub Agents do
12:   if Sub Agent has Type Properties then
13:     create Agent Type Table with the name <Agent.Name>Type;
14:     - add column TypeName of type VARCHAR and make it the primary key;
15:     - make column TypeName a foreign key to its Super Agent Type Table;
16:     - add all Type Properties as columns with the specified type;
17:   end if
18:   if Sub Agent has Object Properties or generate all sub agent tables is true then
19:     create Agent Table with the name <Agent.Name>;
20:     - add all Object Properties as columns with the specified type;
21:     - create same primary key as in Super Agent and make it a foreign key to it;
22:   end if
23: end for
```

Algorithm 2 Resource Mapping Rules

```
1: for all Super Resources do
2:   create Resource Type Table with the name <Resource.Name>Type;
3:   - add column TypeName of type VARCHAR and make it the primary key;
4:   - add all Type Properties as columns with the specified type;
5:   - insert the name of each related Sub Resource into the table;
6:   if Super Resource is a Bulk Resource then
7:     - add column QoH with data type QoHDataType;
8:   end if
9:   if Super Resource has Object Properties then
10:    create Resource Table with the name <Resource.Name>;
11:    - add all Object Properties as columns with the specified type;
12:    - make the primary key object property the primary key column;
13:    - create foreign key to the Resource Type Table;
14:  end if
15: end for
16: for all Sub Resources do
17:   if Sub Resource has Type Properties then
18:    create Resource Type Table with the name <Resource.Name>Type;
19:    - add column TypeName of type VARCHAR and make it the primary key;
20:    - make column TypeName a foreign key to its Super Resource Type Table;
21:    - add all Type Properties as columns with the specified type;
22:   end if
23:   if Sub Resource has Object Properties or generate all sub resource tables is true then
24:    create Resource Table with the name <Resource.Name>;
25:    - add all Object Properties as columns with the specified type;
26:    - create same primary key as in Super Resource and make it a foreign key to it;
27:   end if
28: end for
```

Algorithm 3 Labor Resource Mapping Rules

```
1: for Labor Resource do
2:   create Labor Resource Type Table with the name RB.Labor;
3:   - add column TypeName of type VARCHAR and make it the primary key;
4:   - add all Type Properties as columns with the specified type;
5:   - add column QoHInMinutes with data type INTEGER;
6:   - add column Super Inside Agent Primary Key with the corresponding data type to reference
   the inside agent, who the minutes of labor belong to;
7: end for
```

Algorithm 4 Operational Mapping Rules

```
1: create Duality Table with the name <PlanningModel.Name>DualityTransfer/Transformation
2: - table name either Tranfer or Transformation according to model type
3: - add DualityId as primary key column
4: - add foreign key ContractId to the ContractTable
5: for all Events do
6:   create an Event Table
7:   - add all Object Properties as columns with the specified type
8:   - make the primary key object property the primary key column
9:   - add foreign key to the Duality Table and the Event Type Table
10:  - add foreign key to the Commitment Table which this event fulfills
11:  for all Connected Agents (participations) do
12:    if Single Agent then
13:      foreign key to Agent Table and add minutes used if is inside agent
14:    else if Multiple Agent then
15:      create a EventParticipateAgents Table
16:      - foreign key to Event Table and Agent Table
17:      - add minutes used column if is inside agent
18:    end if
19:  end for
20:  for all Connected Resources (stock-flows) do
21:    if Single Resource then
22:      foreign key to Resource Table and add minutes used if isUsed
23:    else if Multiple Resource then
24:      create a EventStockFlowResources Table
25:      - foreign key to Event Table and Resource Table
26:      - add minutes used column if isUsed
27:    else if Single Bulk Resource then
28:      foreign key to Resource Type Table and add minutes used if isUsed
29:      add quantity (of quantity type) column
30:    else if Multiple Bulk Resources then
31:      create a EventStockFlowResourceTypes Table
32:      - add Id as primary key column
33:      - foreign key to Event Table and Resource Type Table
34:      - add quantity (of quantity type) column
35:      - add minutes used column if isUsed
36:    end if
37:  end for
38: end for
```

Algorithm 5 Operational Stockflow and Participation Relationship Mapping Rules

```
1: for all Stockflow properties do
2:   if Single Resource connected then
3:     add the Stockflow Property as a column with the specified type to the existing event table;
4:   else if Multiple Resource connected then
5:     add the Stockflow Property as a column with the specified type to the existing stockflow
6:     table;
7:   end if
8: end for
9: for all Participate properties do
10:  if Single Agent connected then
11:    add the Participate Property as a column with the specified type to the existing event
12:    table;
13:  else if Multiple Agent connected then
14:    add the Participate Property as a column with the specified type to the existing participates
15:    table;
16:  end if
17: end for
```

Algorithm 6 Planning Mapping Rules

```
1: create Reciprocity Table with the name <PlanningModel.Name>ReciprocityContract/Schedule
2: - table name either Contract or Schedule according to model type
3: - add ContractId as primary key column
4: - add foreign key to the two committing Agents in Agent Table
5: for all Events do
6:   create an Event Type Table with TypeName of type VARCHAR as primary key
7:   - add all Type Properties as columns with the specified type
8:   create a Commitment Table
9:   - add all Commitment Properties as columns with the specified type
10:  - make the primary key commitment property the primary key column
11:  - add foreign key to the Reciprocity Table and the Event Type Table
12:  for all Connected Agents (specify/reserve participations) do
13:    if Single Agent then
14:      foreign key to Agent Table and add minutes scheduled if is inside agent
15:    else if Multiple Agent then
16:      create a CommitmentReservesAgents Table
17:      - foreign key to Commitment Table and Agent Table
18:      - add minutes scheduled column if is inside agent
19:    else if Single Agent Type then
20:      foreign key to Agent Type Table and add minutes scheduled if inside agent
21:    else if Multiple Agent Type then
22:      create a CommitmentSpecifiesAgentTypes Table
23:      - add Id as primary key column
24:      - foreign key to Commitment Table and Agent Type Table
25:      - add minutes scheduled column if is inside agent
26:    end if
27:  end for
28:  for all Connected Resources (specify/reserve stock-flows) do
29:    if Single Resource then
30:      foreign key to Resource Table and add minutes scheduled if isUsed
31:    else if Multiple Resource then
32:      create a CommitmentReservesResources Table
33:      - foreign key to Commitment Table and Resource Table
34:      - add minutes scheduled column if isUsed
35:    else if Single Resource Type then
36:      foreign key to Resource Type Table and add minutes scheduled if isUsed
37:      add quantity (of quantity type) column if is Bulk Resource
38:    else if Multiple Resource Type then
39:      create a CommitmentSpecifiesResourceTypes Table
40:      - add Id as primary key column
41:      - foreign key to Commitment Table and Resource Type Table
42:      - add quantity (of quantity type) column
43:      - add minutes scheduled column if isUsed
44:    end if
45:  end for
46: end for
```

Algorithm 7 Planning Stockflow and Participation Relationship Mapping Rules

```
1: for all Resource (type) stockflow relationships do
2:   if Policy properties exist then
3:     create an Event Type Policy Resource (Type) Table
4:     - foreign key/primary key to Event Type Table and Resource (Type) Table
5:     for all Policy properties do
6:       - add the stockflow Policy Property as a column with the specified type;
7:     end for
8:   end if
9:   for all Reserve properties do
10:    if Single Resource (Type) connected then
11:      add the stockflow Reserve Property as a column with the specified type to the existing
      commitment table;
12:    else if Multiple Resource (Types) connected then
13:      add the stockflow Reserve Property as a column with the specified type to the existing
      commitment-reserves/specifies-resource(type) table;
14:    end if
15:  end for
16: end for
17: for all Participate relationships do
18:   if Policy properties exist then
19:     create an Event Type Policy Agent (Type) Table
20:     - foreign key/primary key to Event Type Table and Agent (Type) Table
21:     for all Policy properties do
22:       - add the participate Policy Property as a column with the specified type;
23:     end for
24:   end if
25:   for all Reserve properties do
26:    if Single Agent (Type) connected then
27:      add the participate Reserve Property as a column with the specified type to the existing
      commitment table;
28:    else if Multiple Agent (Type) connected then
29:      add the participate Reserve Property as a column with the specified type to the existing
      commitment-reserve/specifies-agent(type) table;
30:    end if
31:  end for
32: end for
```
