Generalizing Multi-Context Systems for Reactive Stream Reasoning Applications

Stefan Ellmauthaler

Computer Science Institute
Leipzig University
Germany

ICCSW 2013
September 27th 2013

1Funded by the DFG (project FOR 1513)
Outline

1 Motivation

2 Background

3 Preference-based Iterative Managed Multi-Context Systems

4 Reactive Bridge Rules

5 Conclusion & Future Work
Assisted Living (AL)
An Application for Artificial Intelligence

The Basic Idea

- Enhance an apartment with an AI which monitors the activities of daily living of the inhabitants.
Assisted Living (AL)
An Application for Artificial Intelligence

The Basic Idea

- Enhance an apartment with an AI which monitors the activities of daily living of the inhabitants.
- Coordinate services by outside health care providers.
The Basic Idea

- Enhance an apartment with an AI which monitors the activities of daily living of the inhabitants.
- Coordinate services by outside health care providers.
- Provide supervision and assistance to ensure the inhabitants
  - health,
  - safety, and
  - well-being.
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
- Bob is on the toilet
- Bob is sleeping

Unwanted situation - what to do now?
- Wake up Bob
- Disable the stove
Assisted Living
An Example of AL in Action
Assisted Living
An Example of AL in Action
Assisted Living
An Example of AL in Action

What shall the AI believe?

- The stove is active
- Bob is cooking
- Bob is on the toilet
- Bob is sleeping

Unwanted situation - what to do now?

- Wake up Bob
- Disable the stove

Stefan Ellmauthaler
Generalizing MCS for Reactive Stream Reasoning
ICCSW '13, London
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active

Stefan Ellmauthaler
Generalizing MCS for Reactive Stream Reasoning
ICCSW ’13, London
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
- Bob is on the toilet
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
- Bob is on the toilet
- Bob is sleeping
Assisted Living
An Example of AL in Action

What shall the AI believe?
- The stove is active
- Bob is cooking
- Bob is on the toilet
- Bob is sleeping
- Unwanted situation - what to do now?
Assisted Living

An Example of AL in Action

What shall the AI believe?

- The stove is active
- Bob is cooking
- Bob is on the toilet
- Bob is sleeping
- Unwanted situation - what to do now?
  - Wake up Bob
  - Disable the stove
Realization of this Vision
A first step and some considerations

AL-Environment
- Sensors
- Gadgets to communicate/(re)act
- Reasoning units

Requirements
- Communication between the components
- Continuous evaluation of the situation
- Intelligent reasoning about intentions and beliefs
Realization of this Vision
A first step and some considerations

**AL-Environment**
- Sensors
- Gadgets to communicate/(re)act
- Reasoning units

**Requirements**
- Communication between the components
- Continuous evaluation of the situation
- Intelligent reasoning about intentions and beliefs
Realization of this Vision
A first step and some considerations

AL-Environment
- Sensors
- Gadgets to communicate/(re)act
- Reasoning units

Requirements
- Communication between the components
- Continuous evaluation of the situation
- Intelligent reasoning about intentions and beliefs

Existing Concepts
- (managed) Multi-Context Systems [1]
Realization of this Vision
A first step and some considerations

Requirements
- Communication between the components
- Continuous evaluation of the situation
- Intelligent reasoning about intentions and beliefs

Existing Concepts
- (managed) Multi-Context Systems [1]
- Stream Reasoning concepts
  - oclingo [2]
  - C-SPARQL [4]
(managed) Multi-Context Systems (mMCS)

Definition

A managed Multi-Context System $M$ is a collection $(C_1, \ldots, C_n)$ of managed contexts where, for $1 \leq i \leq n$, each managed context $C_i$ is a quintuple $C_i = (LS_i, kb_i, br_i, OP_i, mng_i)$ such that

- $LS_i = (BS_{LS_i}, KB_{LS_i}, ACC_{LS_i})$ is a logic suite,
- $kb_i \in KB_{LS_i}$ is a knowledge base,
- $OP_i$ is a management base,
- $br_i$ is a set of bridge rules for $C_i$, with the form
  
  $$\text{op}_i \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), \not(c_{j+1} : p_{j+1}), \ldots, \not(c_m : p_m).$$

  such that $\text{op}_i \in F_{LS_i}^{OP_i}$ and for all $1 \leq k \leq m$ there exists a context $c_k \in (C_1, \ldots, C_n)$ such that $p_k \in S \in BS_{LS_{c_k}}$, and
- $mng_i$ is a management function over $LS_i$ and $OP_i$. 

Stefan Ellmauthaler  
Generalizing MCS for Reactive Stream Reasoning  
ICCSW '13, London
Let $M = (C_1, \ldots, C_n)$ be an mMCS. A belief state $S = (S_1, \ldots, S_n)$ is an equilibrium of $M$ iff for every $1 \leq i \leq n$ there exists some $(kb'_i, ACC_{LS_i}) \in mng_i(app_i(S), kb_i)$ such that $S_i \in ACC_{LS_i}(kb'_i)$. 
Each context may use a different formalism e.g.:
- Theorem Prover
- Datalog
- ASP
- SQL
- DL
- ...

bridge rule:
\[ op_3 \leftarrow (C_1 : l_1), (C_1 : l_2), \text{not}(C_2 : l_1) \]
### Basic Concepts

- Utilize iterative and stream reasoning approach from potassco [2, 3]
- Specialized contexts for different tasks

### Context types

- Observing Contexts
- Reasoning Contexts
- Control Contexts
  - sliding windows
  - inconsistency handling policies
  - semantics and reasoning modes
  - determine actions
  - decide meta-reasoning aspects
Preference-based Iterative Managed Multi-Context Systems (pimMCS)

bridge rule:

\[ \text{op}_3 \leftarrow (C_1 : l_1), (C_1 : l_2), \text{not}(C_2 : l_1) \]
Preference-based Iterative Managed Multi-Context Systems (pimMCS)

bridge rule:

$\text{op}_3 \leftarrow (C_1 : l_1), (C_1 : l_2), \text{not}(C_2 : l_1)$

$\text{Obs}_1 = (\text{Obs}_0, \text{Obs}_1, \ldots)$

$R(\text{Obs}) = \text{Kb}_0, \text{Eq}_0, \text{Kb}_1, \text{Eq}_1, \ldots$
Preference-based Iterative Managed Multi-Context Systems (pimMCS)

\[ \text{Obs} = (\text{Obs}^0, \text{Obs}^1, \ldots) \]

\[ \text{bridge rule:} \]

\[ \text{op}_3 \leftarrow (C_1 : l_1), (C_1 : l_2), \text{not}(C_2 : l_1) \]

\[ \text{Obs}^i \]

\[ \text{Obs}^i \]

\[ \text{Obs}^i \]
Preference-based Iterative Managed Multi-Context Systems (pimMCS)

\[ \text{Obs} = (\text{Obs}^0, \text{Obs}^1, \ldots) \]

\[ R(\text{Obs}) = \text{Kb}^0, \text{Eq}^0, \text{Kb}^1, \text{Eq}^1, \ldots \]
Preference-based Iterative Managed Multi-Context Systems (pimMCS)

\[ \text{bridge rule:} \]

\[ \text{op}_3 \leftarrow (C_1 : l_1), (C_1 : l_2), \text{not}(C_2 : l_1) \]

\[ \text{Obs} = (\text{Obs}^0, \text{Obs}^1, \ldots) \]

\[ R(\text{Obs}) = \text{Kb}^0, \text{Eq}^0, \text{Kb}^1, \text{Eq}^1, \ldots \]
Some flaws of pimMCS

- If there is no global equilibrium, no actions between contexts
- Computation of one global equilibrium is expensive [1]
If there is no global equilibrium, no actions between contexts
Computation of one global equilibrium is expensive [1]

⇒ fast reaction to events is highly unlikely
## Concept Idea
- Use bridge rules on local belief sets instead of global equilibria
- All contexts have input streams
- Manipulate the input stream of other contexts

## Comparison to pimMCS
- Contexts do not have to wait for the global equilibria
- No agreement necessary
- Communication in case of emergencies is more immediate
- Inconsistency handling needs to be done via stream handling
Definition

A Reactive Bridge Rule (RBR) \( r \) for a context \( C_i \) of a collection of \( n \) contexts is a rule of the form

\[
t, j : h \leftarrow b_1, \ldots, b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m
\]

where

- \( t \in \{b, c\} \) specifies whether the literals need to be evaluated bravely or cautiously,
- \( j \leq n \) specifies which context will be provided with additional information,
- \( h \) is information which may be added to the input stream of \( C_j \), and
- for \( l \leq m \), \( b_l \) is a literal.
Reactive Bridge Rules

Definition

Let $r$ be an RBR of a context $C_i$, $ACC_{LS_i} \in ACC_{LS_i}$ be a selected semantics, and $S = \{S_1, \ldots, S_j\}$ be the belief sets of $C_i$ at step $t$, such that $S = ACC_{LS_i}(kb^t_i)$, where $kb^t_i$ is the knowledge base of context $C_i$ at step $t$.

- If $r$ is a cautious RBR, it is satisfied if
  $$\forall B \in S (b^+(r) \subseteq B \wedge b^-(r) \cap B = \emptyset).$$
- If $r$ is a brave RBR, it is satisfied if
  $$\exists B \in S (b^+(r) \subseteq B \wedge b^-(r) \cap B = \emptyset).$$

If a rule $r$ is satisfied, then $h$ will be added to the input stream of the context $C_j$ at step $t + 1$. 

Conclusion

We have introduced

- **pimMCS** to compute equilibria on stream based MCS
- **RBRs** to modify input streams of other contexts
Conclusion & Future Work

Conclusion

We have introduced

- **pimMCS** to compute equilibria on stream based MCS
- **RBRs** to modify input streams of other contexts

In addition there is

- a combination of pimMCS and RBRs
  - reactive managed Multi-Context Systems (rmMCS)
  - computes runs with equilibria like pimMCS
  - free capacities used for additional belief sets
  - RBRs may fire during the computation of the equilibria
Conclusion & Future Work

Conclusion
We have introduced
- **pimMCS** to compute equilibria on stream based MCS
- **RBRs** to modify input streams of other contexts

In addition there is
- a combination of pimMCS and RBRs
  - reactive managed Multi-Context Systems (rmMCS)
  - computes runs with equilibria like pimMCS
  - free capacities used for additional belief sets
  - RBRs may fire during the computation of the equilibria

Further Work

- Restrictions to Contexts
- Side effects of rmMCS
- Instantiation
- Implementation
- Reactive “extensions” for one-shot formalisms
Thank you!

The pictures used in this talk are taken from [5, 6]


References II


Appendix
Preference-based Iterative Managed Multi-Context Systems

Definition

Let $M$ be a managed MCS with contexts $C = (C_1, \ldots, C_n)$ ($C_1, \ldots, C_k$ are observer contexts), where $C_i \in C$ is a quintuple $C_i = (LS_i, kb_i, br_i, OP_i, mng_i, pref_i)$. Let $Obs = (Obs^0, Obs^1, \ldots)$ be a sequence of observations, that is, for $j \geq 0$, $Obs^j = (Obs_i^j)_{i \leq k}$, where $Obs_i^j$ is the new (sensor) information for context $i$ at step $j$, which is formalized as sets of formulas.

A run $R$ of $M$ induced by $Obs$ is a sequence

$$R = Kb^0, Eq^0, Kb^1, Eq^1, \ldots$$

where
A run $R$ of $M$ induced by $Obs$ is a sequence

$$R = Kb^0, Eq^0, Kb^1, Eq^1, \ldots$$

where

- $Kb^0 = (Kb_i^0)_{i \leq n}$ is the collection of initial knowledge bases, $Eq^0$ an equilibrium of $Kb^0$,
- for $j \geq 1$ and $i \leq n$, $Kb_i^j$ is the knowledge base of context $C_i$ produced by the context’s management function for the computation of $Eq^{j-1}$, and $Kb^j = (Kb_i^j)_{i \leq n}$,
- for $j \geq 1$, $Eq^j$ is an equilibrium for the knowledge bases

$$(Kb_0^j \cup Obs_0^j, \ldots, Kb_k^j \cup Obs_k^j, Kb_{k+1}^j, \ldots, Kb_n^j).$$

$(C, Obs, pref)$ is called a **preference-based iterative managed Multi-Context System**.