Multi-Context Systems: Integrating Heterogeneous Knowledge Bases

Gerhard Brewka

Computer Science Institute
University of Leipzig
brewka@informatik.uni-leipzig.de

joint work with Thomas Eiter, Michael Fink, Antonius Weinzierl

1. Background: The Problem

- Assume Klinikum Dortmund has
  - a patient database (e.g. Oracle),
  - an ontology of diseases (using description logic),
  - an ontology of the human body (using OWL),
  - an expert system describing the effects of different medications (using disjunctive logic programming).
- How to integrate the knowledge represented in such a variety of diverse formalisms?
- Translate everything into a single all-purpose formalism?
- Which one? Who does it?
- Need principled way of integrating knowledge expressed in different formats/languages/logics.

1. Background: The Problem

- Assume Klinikum Dortmund has
  - a patient database (e.g. Oracle),
  - an ontology of diseases (using description logic),
  - an ontology of the human body (using OWL),
  - an expert system describing the effects of different medications (using disjunctive logic programming).
- How to integrate the knowledge represented in such a variety of diverse formalisms?
- Translate everything into a single all-purpose formalism?
- Which one? Who does it?
- Need principled way of integrating knowledge expressed in different formats/languages/logics.

1. Background: The Problem

- Assume Klinikum Dortmund has
  - a patient database (e.g. Oracle),
  - an ontology of diseases (using description logic),
  - an ontology of the human body (using OWL),
  - an expert system describing the effects of different medications (using disjunctive logic programming).
- How to integrate the knowledge represented in such a variety of diverse formalisms?
- Translate everything into a single all-purpose formalism?
- Which one? Who does it?
- Need principled way of integrating knowledge expressed in different formats/languages/logics.

Our Answer

Multi-Context Systems

- Leave the different units separate; call them *contexts* (for historical reasons).
- Describe declaratively the flow of information among contexts that is necessary for the particular application.
- Use nonmonotonic rules for this purpose; heads specify what information needs to be added, body refers to other contexts.
- Define “reasonable” belief states (equilibria) for whole system.
- Note: arbitrary cycles among contexts allowed.

Our Answer

Multi-Context Systems

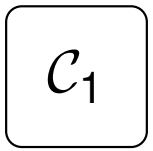
- Leave the different units separate; call them *contexts* (for historical reasons).
- Describe declaratively the flow of information among contexts that is necessary for the particular application.
- Use nonmonotonic rules for this purpose; heads specify what information needs to be added, body refers to other contexts.
- Define “reasonable” belief states (equilibria) for whole system.
- Note: arbitrary cycles among contexts allowed.

Our Answer

Multi-Context Systems

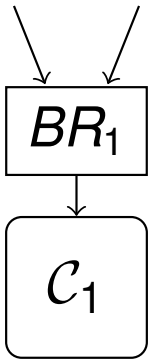
- Leave the different units separate; call them *contexts* (for historical reasons).
- Describe declaratively the flow of information among contexts that is necessary for the particular application.
- Use nonmonotonic rules for this purpose; heads specify what information needs to be added, body refers to other contexts.
- Define “reasonable” belief states (equilibria) for whole system.
- Note: arbitrary cycles among contexts allowed.

The Short Story



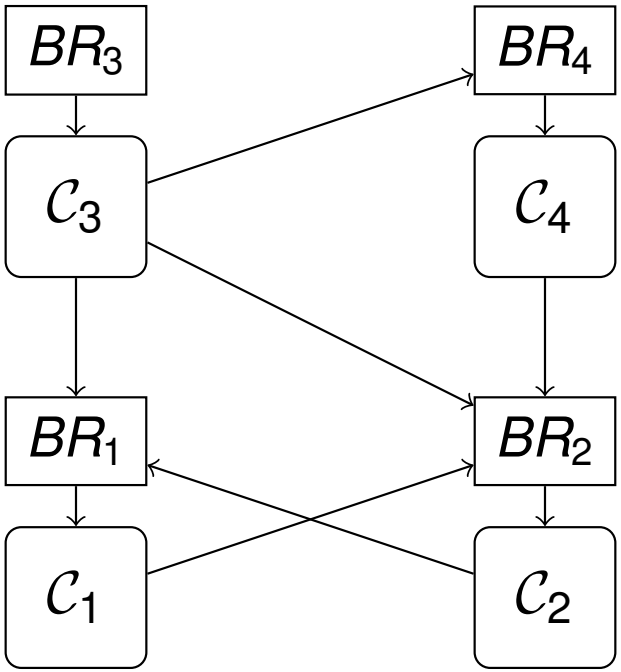
An isolated context.

The Short Story



An context equipped with bridge rules,
can “listen" to other contexts.

The Short Story



A multi-context system.

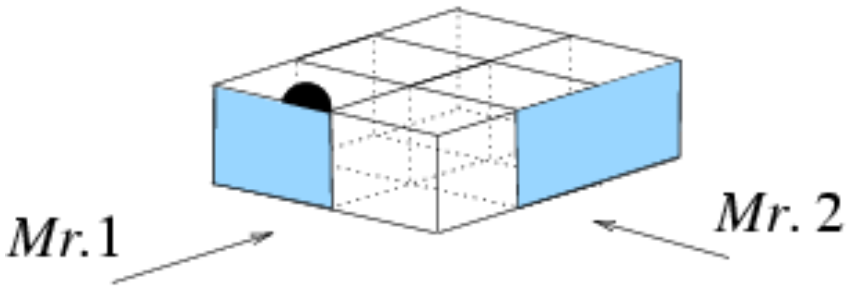
What Bridge Rules Can Do For You

- Flow of information can be specified declaratively $\Rightarrow$ control of reasoning
- Information not only passed on as is, but can be modified:
  - translation into the language/format of another context
  - abstraction
  - information selection and hiding
  - conclusions based on absence of information
  - preferences among parent contexts
  - majorities, voting ...

Contexts in AI

- First investigated by John McCarthy (1987), without definition
- Intuitively, a context describes a particular viewpoint, perspective, granularity, person/agent/database ...
- Here: (almost/somewhat) independent unit of reasoning
- Aspects of multi-context systems:
  - **Locality:** different languages, reasoning methods, logics
  - **Compatibility:** information flow between contexts
- Provide a particular form of information integration

Example: Magic Box



Brief History of MCS

- Monotonic MCS developed in the 90s (Giunchiglia, Serafini et al.)
- Integrate monotonic inference systems
- Attempts to make bridge rules nonmonotonic (Roelofsen/Serafini IJCAI 2005); (B./Roelofsen/Serafini IJCAI 2007)
- Resulting system homogeneous: reasoners of same type (namely logic programs or Reiter's default logic)
- *Heterogeneous* nonmonotonic MCS capturing *monotonic and nonmonotonic* contexts defined by B./Eiter (AAAI 2007).
- Managed MCS B./Eiter/Fink/Weinzierl (IJCAI 2011).

Outline

1. Background and Motivation (done)
2. Nonmonotonic MCS (the basic framework)
   - Logics and Contexts
   - Acceptable Belief States and Equilibria
3. Managed MCS (a substantial generalization)
   - Operations on Knowledge Bases
   - The Generalization
4. Revision-based MCS (a useful instantiation)
   - Inconsistencies in MCS
   - Preference-Based Conflict Handling
5. Conclusions

2. Nonmonotonic MCS: “Logics”

Want to capture the “typical” KR logics, including nonmonotonic logics with multiple acceptable belief sets (e.g., Reiter’s Default Logic).

Logic

A logic L is a tuple

$$L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$$

- $\mathbf{KB}_L$ is a set of well-formed knowledge bases, each being a set (of formulas)
- $\mathbf{BS}_L$ is a set of possible belief sets, each being a set (of formulas)
- $\mathbf{ACC}_L : \mathbf{KB}_L \to 2^{\mathbf{BS}_L}$ assigns to each knowledge base a set of acceptable belief sets

Example Logics Over Signature Σ

Propositional logic

- **KB**: the sets of prop. Σ-formulas
- **BS**: the deductively closed sets of prop. Σ-formulas
- **ACC**(kb): $Th(kb)$

Default logic

- **KB**: the default theories over Σ
- **BS**: the deductively closed sets of Σ-formulas
- **ACC**(kb): the extensions of kb

Normal LPs under answer set semantics

- **KB**: the logic programs over Σ
- **BS**: the sets of atoms of Σ
- **ACC**(kb): the answer sets of kb

Multi-Context Systems

Bridge Rules

$$s \leftarrow (r_1 : p_1), \ldots, (r_j : p_j), \textbf{not}\ (r_{j+1} : p_{j+1}), \ldots, \textbf{not}\ (r_m : p_m)$$

Multi-Context System

An MCS $M = (C_1, \ldots, C_n)$ consists of contexts

$$C_i = (L_i, kb_i, br_i),\ i \in \{1, \ldots, n\},$$

where

- each L_i is a logic,
- each $kb_i \in \mathbf{KB}_i$ is a L_i-knowledge base, and
- each br_i is a set of L_i-bridge rules over M.

2 sources of nonmonotonicity: *context logics* and *bridge rules*.

Example

Consider the multi-context system $M = (C_1, C_2)$, where the contexts are different views of a paper by the authors.

- C_1:
  - L_1 = Classical Logic
  - $kb_1 = \{\, unhappy \supset revision \,\}$
  - $br_1 = \{\, unhappy \leftarrow (2 : work) \,\}$
- C_2:
  - L_2 = Reiter's Default Logic
  - $kb_2 = \{\, good : accepted / accepted \,\}$
  - $br_2 = \{\, work \leftarrow (1 : revision),$
    $good \leftarrow \textbf{not}\ (1 : unhappy) \,\}$

Acceptable Belief States

- **Belief state:** sequence of belief sets, one for each context
- **Fundamental Question:** *Which belief states are acceptable?*
- **Intuition:** belief states must be in *equilibrium*: belief set selected for C_i must be acceptable for C_i's knowledge base *augmented by the heads of C_i's applicable bridge rules.*

Equilibrium

A belief state $S = (S_1, \ldots, S_n)$ of M is an equilibrium iff for $i \in \{1, \ldots, n\}$

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{head(r) \mid r \in br_i \text{ is applicable in } S\}).$$

Example (ctd)

Reconsider multi-context system $M = (C_1, C_2)$:

- $kb_1 = \{ \mathit{unhappy} \supset \mathit{revision} \}$ (Classical Logic)
- $kb_2 = \{ \mathit{good} : \mathit{accepted} / \mathit{accepted} \}$ (Default Logic)
- $br_1 = \{ \mathit{unhappy} \leftarrow (2 : \mathit{work}) \}$
- $br_2 = \{ \mathit{work} \leftarrow (1 : \mathit{revision}),$
  $\mathit{good} \leftarrow \textbf{not}\ (1 : \mathit{unhappy}) \}$

M has two equilibria:

- $E_1 = (\mathit{Th}(\{\mathit{unhappy}, \mathit{revision}\}), \mathit{Th}(\{\mathit{work}\}))$ and
- $E_2 = (\mathit{Th}(\{\mathit{unhappy} \supset \mathit{revision}\}), \mathit{Th}(\{\mathit{good}, \mathit{accepted}\}))$

Groundedness

- Definition of equilibria does not exclude self-justifying cycles
- Ok if formalisms (like AEL) are involved which also have them
- Can be excluded (sometimes) if nonmonotonic formalisms have groundedness check
- In this case generalize Gelfond/Lifschitz reduct for logic programs:
  - Guess belief state $S = (S_1, \ldots, S_n)$
  - Use S to reduce bridge rules as in GL-reduct
  - If C_i is nonmonotonic, use S_i to reduce its KB
  - Compute smallest belief state of resulting monotonic MCS and compare with S

Further remarks

Alternative semantics:

- also developed well-founded semantics for MCS

Complexity:

- depends on complexity of context logics
- L has kernel reasoning in Δ_k^p if
  - belief sets have polynomial representation, and
  - deciding whether K represents some belief set of kb is in Δ_k^p
- if context logics have kernel reasoning in Δ_k^p then
  - deciding existence of equilibrium is in Σ_{k+1}^p
  - brave reasoning in Σ_{k+1}^p, cautious reasoning in Π_{k+1}^p
- complexity of context logics that of standard propositional NMR approaches $\Rightarrow$ complexity does not increase

Further remarks

Alternative semantics:

- also developed well-founded semantics for MCS

Complexity:

- depends on complexity of context logics
- L has kernel reasoning in Δ_k^p if
  - belief sets have polynomial representation, and
  - deciding whether K represents some belief set of kb is in Δ_k^p
- if context logics have kernel reasoning in Δ_k^p then
  - deciding existence of equilibrium is in Σ_{k+1}^p
  - brave reasoning in Σ_{k+1}^p, cautious reasoning in Π_{k+1}^p
- complexity of context logics that of standard propositional NMR approaches $\Rightarrow$ complexity does not increase

Some “Instances” of MCS

- Parameterized Logic Programming; R. Goncalves, J.J. Alferes, JELIA 2010
  - Logic programs with atoms generalized to formulas in particular monotonic logic (parameter logic)
  - Corresponds to MCS with single context, KB empty
  - Bridge rules refer back to same context
  - Now PLP stable models coincide with our equilibria
- Base Logics in Argumentation; A. Hunter, COMMA-10
  - Introduces so-called bilogics composed of 2 base logics
  - Corresponds to MCS with 2 contexts, one for each base logic
  - Bridge rules add formulas whenever they are believed in other context
  - Consequence in bilogic corresponds to equilibrium
  - Caveat: infinitary MCS necessary

What has been done since 2007?

- Implementation:
  - Centralized based on HEX programs.
  - Decentralized.
- Inconsistency handling:
  - Identify bridge rules responsible for non-existence of equilibria.
  - Diagnosis and explanation.
  - Preference based assessment.
- Argumentation Context Systems:
  - Focus on a particular type of reasoners: Dung frameworks.
  - Information not only added, but also be deleted, changed, revised.
  - Achieved by adding a mediator to each context; handles inconsistencies, modifies underlying context.
- Current topics: partial knowledge, information aggregation, dynamic MCS, managed MCS.

3. Managed MCS

- MCS neglect 2 important aspects:
  - provide only basic mechanisms for cases where parent information is conflicting (e.g. encoding preference in rules),
  - allow to *add information* only, no other operations (e.g. updating/revising the underlying KB).
- mMCS solve these problems.
- Basic idea: context specific operations in heads of bridge rules.
- Leads to very general and powerful framework.

Definitions, informally and (somewhat) simplifying

- Each logic comes with a set of operations O.
- Operational statement: operation applied to a formula (e.g. insert(p), delete(p), revise(p), ...).
- Bridge rules: as before, but heads are operational statements.
- Management function: $mng : 2^{Opst} \times KB \rightarrow 2^{KB}$, produces collection of KBs out of set of operational statements and KB.
- Managed context:
  - logic, KB (as before),
  - bridge rules (now over O),
  - management function (new).
- mMCS: collection of managed contexts.

Equilibria

- Belief state $S = (S_1, \ldots S_n)$: a belief set for each context.
- Equilibrium: belief set chosen for each context must be acceptable for one of the KBs obtained by applying the management function to the heads of applicable bridge rules and the context's KB.
- More formally, for all contexts $C_i = (L_i, kb_i, br_i, mng_i)$:
  - let S_i be the belief set chosen for C_i,
  - let Op_i be the heads of bridge rules in br_i applicable in S,
  - then $S_i \in ACC_i(kb')$ for some $kb' \in mng_i(Op_i, kb_i)$.
- mMCS powerful tool for describing the influence contexts can have on each other.

4. Inconsistency Management

Different forms of inconsistency can arise in mMCS:

1. *Nonexistence of equilibria*: investigated in [Eiter et al. 10] for MCS.
2. *Local inconsistency:* inconsistent belief sets in equilibria; presupposes adequate notion of consistency.
3. *Operator inconsistency:* operations in the heads of applicable bridge rules are conflicting, e.g. $add(p)$ and $delete(p)$.

Handling inconsistencies of type 2 and 3 is one of the motivations that led to the development of mMCS.

Involves aspect of decision making.

An Instantiation of mMCS: Revision-based MCS

- Assume single operation *inc* (include).
- Want to guarantee consistency of belief sets in equilibria.
- What can go wrong? Formulas to be included
  1. may be inconsistent with each other,
  2. may be inconsistent with context KB.
- For 1 introduce preferences among bridge rules.
- For 2 assume consistency preserving base revision operator

$$rev : KB \times KB \to KB$$

that is, $rev(kb_1, kb_2)$ consistent whenever kb_2 is.

An Instantiation of mMCS: Revision-based MCS

- Assume single operation *inc* (include).
- Want to guarantee consistency of belief sets in equilibria.
- What can go wrong? Formulas to be included
  1. may be inconsistent with each other,
  2. may be inconsistent with context KB.
- For 1 introduce preferences among bridge rules.
- For 2 assume consistency preserving base revision operator

$$rev : KB \times KB \to KB$$

that is, $rev(kb_1, kb_2)$ consistent whenever kb_2 is.

An Instantiation of mMCS: Revision-based MCS

- Assume single operation *inc* (include).
- Want to guarantee consistency of belief sets in equilibria.
- What can go wrong? Formulas to be included
  1. may be inconsistent with each other,
  2. may be inconsistent with context KB.
- For 1 introduce preferences among bridge rules.
- For 2 assume consistency preserving base revision operator

$$rev : KB \times KB \rightarrow KB$$

that is, $rev(kb_1, kb_2)$ consistent whenever kb_2 is.

Revision-based MCS, ctd.

- Represent total preorder on bridge rules by using indices $inc_1, inc_2, \ldots$ (lower index = higher priority).
- Preferred set: pick maxi-consistent subset of inc_1-formulas, extend it maxi-consistently with inc_2-formulas etc.
- Define the management function as follows: for

$$S = \{inc_1(p_{1,1}), \ldots, inc_1(p_{1,m}), \ldots, inc_k(p_{1,m}), \ldots, inc_k(p_{k,n})\}$$

let:

$kb' \in mng(S, kb)$ iff $kb' \in rev(kb, F)$ for a preferred set F of S.

- Each belief set in each equilibrium apparently consistent.

Example

C context based on propositional logic, its KB is

$$\{July \rightarrow \neg Rain, Rain \rightarrow Umbrella, July\}$$

2 parent contexts; C_1 believes $Rain$, C_2 believes $\neg Rain$. Bridge rules

$$\{inc_1([\neg]Rain) \leftarrow 1{:}[\neg]Rain;\ inc_2([\neg]Rain) \leftarrow 2{:}[\neg]Rain\}$$

$K' \in rev(K, F)$ iff $K' = S \cup F$ for maximal $S \subseteq K$ consistent with F.

Single preferred set (as C_1 preferred to C_2): $\{Rain\}$

Acceptable belief sets for C:

$$Th(\{Rain, July \rightarrow \neg Rain, Rain \rightarrow Umbrella\})$$
$$Th(\{Rain, July, Rain \rightarrow Umbrella\})$$

Example

C context based on propositional logic, its KB is

$$\{July \rightarrow \neg Rain, Rain \rightarrow Umbrella, July\}$$

2 parent contexts; C_1 believes $Rain$, C_2 believes $\neg Rain$. Bridge rules

$$\{inc_1([\neg]Rain) \leftarrow 1{:}[\neg]Rain;\ inc_2([\neg]Rain) \leftarrow 2{:}[\neg]Rain\}$$

$K' \in rev(K, F)$ iff $K' = S \cup F$ for maximal $S \subseteq K$ consistent with F.

Single preferred set (as C_1 preferred to C_2): $\{Rain\}$

Acceptable belief sets for C:

$$Th(\{Rain, July \rightarrow \neg Rain, Rain \rightarrow Umbrella\})$$
$$Th(\{Rain, July, Rain \rightarrow Umbrella\})$$

Example

C context based on propositional logic, its KB is

$$\{July \rightarrow \neg Rain, Rain \rightarrow Umbrella, July\}$$

2 parent contexts; C_1 believes $Rain$, C_2 believes $\neg Rain$. Bridge rules

$$\{inc_1([\neg]Rain) \leftarrow 1{:}[\neg]Rain;\ inc_2([\neg]Rain) \leftarrow 2{:}[\neg]Rain\}$$

$K' \in rev(K, F)$ iff $K' = S \cup F$ for maximal $S \subseteq K$ consistent with F.

Single preferred set (as C_1 preferred to C_2): $\{Rain\}$

Acceptable belief sets for C:

$$Th(\{Rain, July \rightarrow \neg Rain, Rain \rightarrow Umbrella\})$$
$$Th(\{Rain, July, Rain \rightarrow Umbrella\})$$

Example

C context based on propositional logic, its KB is

$$\{July \rightarrow \neg Rain, Rain \rightarrow Umbrella, July\}$$

2 parent contexts; C_1 believes $Rain$, C_2 believes $\neg Rain$. Bridge rules

$$\{inc_1([\neg]Rain) \leftarrow 1{:}[\neg]Rain;\ inc_2([\neg]Rain) \leftarrow 2{:}[\neg]Rain\}$$

$K' \in rev(K, F)$ iff $K' = S \cup F$ for maximal $S \subseteq K$ consistent with F.

Single preferred set (as C_1 preferred to C_2): $\{Rain\}$

Acceptable belief sets for C:

$$Th(\{Rain, July \rightarrow \neg Rain, Rain \rightarrow Umbrella\})$$
$$Th(\{Rain, July, Rain \rightarrow Umbrella\})$$

Example

C context based on propositional logic, its KB is

$$\{July \rightarrow \neg Rain, Rain \rightarrow Umbrella, July\}$$

2 parent contexts; C_1 believes $Rain$, C_2 believes $\neg Rain$. Bridge rules

$$\{inc_1([\neg]Rain) \leftarrow 1{:}[\neg]Rain;\ inc_2([\neg]Rain) \leftarrow 2{:}[\neg]Rain\}$$

$K' \in rev(K, F)$ iff $K' = S \cup F$ for maximal $S \subseteq K$ consistent with F.

Single preferred set (as C_1 preferred to C_2): $\{Rain\}$

Acceptable belief sets for C:

$$Th(\{Rain, July \rightarrow \neg Rain, Rain \rightarrow Umbrella\})$$
$$Th(\{Rain, July, Rain \rightarrow Umbrella\})$$

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Basic systems allow monotonic/nonmonotonic systems to be integrated
- Restriction to extending knowledge.
- Managed MCS introduce arbitrary operations on KBs.
- Focussed on (preference based) inconsistency handling in MCS .
- mMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

Open Issues

- Further generalizations
  - More general bridge rules, e.g. (ordered) disjunctive
  - Aggregates in rule bodies (e.g. computation of sum of salaries, average salaries and the like)
  - More interesting preferences/decision making methods
- Reactive MCS where contexts receive continuous streams of data
- Instances of the framework suitable for particular applications

THANK YOU!

Open Issues

- Further generalizations
  - More general bridge rules, e.g. (ordered) disjunctive
  - Aggregates in rule bodies (e.g. computation of sum of salaries, average salaries and the like)
  - More interesting preferences/decision making methods
- Reactive MCS where contexts receive continuous streams of data
- Instances of the framework suitable for particular applications

THANK YOU!

Open Issues

- Further generalizations
  - More general bridge rules, e.g. (ordered) disjunctive
  - Aggregates in rule bodies (e.g. computation of sum of salaries, average salaries and the like)
  - More interesting preferences/decision making methods
- Reactive MCS where contexts receive continuous streams of data
- Instances of the framework suitable for particular applications

THANK YOU!

Open Issues

- Further generalizations
  - More general bridge rules, e.g. (ordered) disjunctive
  - Aggregates in rule bodies (e.g. computation of sum of salaries, average salaries and the like)
  - More interesting preferences/decision making methods
- Reactive MCS where contexts receive continuous streams of data
- Instances of the framework suitable for particular applications

THANK YOU!