Nonmonotonic Multi-Context Systems: State of the Art and Future Challenges

Gerhard Brewka

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

joint work with Thomas Eiter

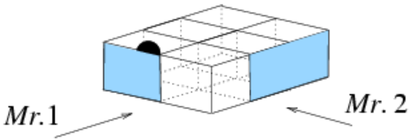
1. Motivation

- Larger and larger bodies of knowledge being formalized
- Sheer size of, say, medical ontologies requires methods for structuring and modularizing KBs
- Wealth of existing logical tools to model ontologies, actions, interactions, dynamic processes, forms of human reasoning, ...
- Single all-purpose formalism not in sight: necessary to integrate several formalisms into a single system
- Often done in an ad hoc way for particular pair of formalisms (e.g. rules and ontologies)
- **Can we do this in a more principled way?**

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



Outline

1. Motivation (done)
2. Nonmonotonic MCS
    - Background
    - Logics and Contexts
    - Acceptable Belief States
3. Argumentation Context Systems
    - Background
    - Context Dependent Argumentation
    - Mediators
    - The Framework and Acceptable Argumentation States
4. Combining MCS and ACS: Outlook
    - Making Logics Context Dependent
    - Mediators and Framework
5. Conclusions

2. Multi-Context Systems

Historical Background

- Monotonic multi-context systems developed by Giunchiglia, Serafini et al. in the 90s
- Integrate different monotonic inference systems
- Information flow modeled using bridge rules
- First attempts to make bridge rules nonmonotonic by Roelofsen/Serafini (2005) and Brewka/Roelofsen/Serafini (Contextual Default Logic, 2007)
- Resulting system homogeneous: reasoners of same type (namely logic programs or Reiter's default logic)

Our Goals

- Generalize existing approaches
- Define a *heterogeneous* multi-context framework accommodating both *monotonic and nonmonotonic* contexts
- Should be capable of integrating logics like description logics, modal logics, default logics, logic programs, etc.

"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)
- $\mathbf{ACC}_L : \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ assigns to each knowledge base a set of acceptable belief sets

L is called *monotonic*, if (1) $|\mathbf{ACC}_L(kb)| = 1$ and (2) $kb \subseteq kb'$, $\mathbf{ACC}_L(kb) = \{S\}$, and $\mathbf{ACC}_L(kb') = \{S'\}$ implies $S \subseteq S'$.

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

- As in monotonic MCS, information integration via bridge rules
- As in Contextual Default Logic, bridge rules (and logics used) can be nonmonotonic
- Unlike in Contextual Default Logic, arbitrary logics can be used

Bridge Rules

$L = L_1, \ldots, L_n$ a collection of logics.

L_k-bridge rule over L ($1 \leq k \leq n$):

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

where (1) every $kb \in \mathbf{KB}_k$ fulfills $kb \cup \{s\} \in \mathbf{KB}_k$, (2) each $r_k \in \{1, \ldots, n\}$, and (3) each p_k is in some belief set of L_{r_k}.

Multi-Context Systems, ctd.

Multi-Context System

A Multi-Context System

$$M = (C_1, \dots, C_n)$$

consists of contexts

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

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's logics.

M can be nonmonotonic because *one of its context logics* is AND/OR because a context has *nonmonotonic 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
  - $kb_1 = \{ \mathit{unhappy} \supset \mathit{revision} \}$
  - $br_1 = \{ \mathit{unhappy} \leftarrow (2 : \mathit{work}) \}$
- C_2:
  - L_2 = Reiter's Default Logic
  - $kb_2 = \{ \mathit{good} : \mathit{accepted} / \mathit{accepted} \}$
  - $br_2 = \{ \mathit{work} \leftarrow (1 : \mathit{revision}),$
    $\mathit{good} \leftarrow \mathbf{not}\ (1 : \mathit{unhappy}) \}$

Acceptable Belief States

- **Belief state:** sequence of belief sets, one for each context
- **Fundamental Question:** *Which belief states are acceptable?*
- Must be based on the knowledge base of a context AND the information accepted in other contexts (if there are appropriate bridge rules)
- **Intuition:** belief states must be in *equilibrium*:

> The selected belief set for each context C_i must be among the acceptable belief sets for C_i's knowledge base *together with the heads of C_i's applicable bridge rules.*

Acceptable Belief States, ctd.

Applicable Bridge Rules

Let $M = (C_1, \ldots, C_n)$. The bridge rule

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

is applicable in belief state $S = (S_1, \ldots, S_n)$ iff (1) $p_i \in S_{r_i}$ $(1 \leq i \leq j)$, and (2) $p_k \notin S_{r_k}$ $(j+1 \leq k \leq m)$.

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 $M = (C_1, C_2)$:

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

M has two equilibria:

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

Groundedness

- Problem: **self-justifying beliefs**
- Present e.g. in Autoepistemic Logic:

$$L\,rich \supset rich$$

- Other nonmonotonic formalisms are "grounded," e.g.
  - Reiter's Default Logic,
  - logic programs with Answer Set Semantics (Gelfond & Lifschitz, 91),
  - ...
- Equilibria of MCSs are possibly ungrounded, e.g. E_1; may be wanted or not
- Groundedness can be achieved by restriction to special class of nonmonotonic formalisms
- Generalization of Gelfond/Lifschitz reduct applied to belief state

Motivation

- Nonmonotonic MCS neglect 2 important aspects:
  - What if information provided by different contexts is conflicting?
  - What if a context does not only add information?
- ACS provide an answer to these questions.
- Focus on a particular type of local reasoners: argumentation frameworks.
- Goals achieved by introducing mediators.

Argumentation Context Systems: Background

- Work based on Dung's widely used abstract argumentation frameworks (AFs).
- Abstract approach: arguments un-analyzed, attacks represented in digraph; can be instantiated in many different ways.
- Argument accepted unless attacked by an accepted argument.
- Semantics single out appropriate accepted sets of arguments:
  - *Grounded extension*: accept unattacked args, eliminate args attacked by accepted args, continue until fixpoint reached.
  - *Preferred extension*: maximal conflict free set which attacks each of its attackers.
  - *Stable extension*: conflict-free set of arguments which attacks each excluded argument.
- (Value based) preferences captured: modify original AF.

Limitations

- No distinction between arguments, meta-arguments, sources of arguments etc.
- Our interest: additional structure and modularity
- Benefits:
  - A handle on complexity and diversity
  - A natural account of multi-agent argumentation
  - Explicit means to model meta-argumentation

Motivating Example: Conference Reviewing

Consider model of the paper review process for a conference

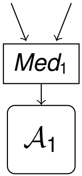
- Hierarchy consisting of PC chair, area chairs, reviewers, authors.
- PC chair determines review criteria.
- Area chairs make sure reviewers make fair judgements and eliminate unjustified arguments from reviews.
- Authors give feedback on reviews. Information flow thus cyclic.
- Reviewers exchange arguments in peer-to-peer discussion.
- Area chairs generate a consistent recommendation.
- PC chair takes recommendations as input for final decision.

Need a flexible framework allowing for cyclic structures encompassing different information integration methods.

The Short Story

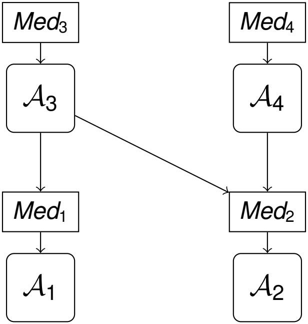
A (lonely) Dung style argumentation framework.

The Short Story



An argumentation module equipped with a mediator, can “listen" to other modules and “talk" to $\mathcal{A}_1$: sets an argumentation context using a context definition language; handles inconsistency.

The Short Story



An argumentation context system.

More Background

Inconsistency Handling

Use 4 methods for picking consistent subset of $(F_1, \ldots, F_n)$, F_i set of formulas (details irrelevant)

	Preference based	Majority based
Credulous	$sub_{\succ}$	maj
Skeptical	$sub_{sk,\succ}$	maj_{sk}

Bridge Rules

Only rules referring to single other module needed
$\Rightarrow$ bridge rules ordinary logic programming rules:

$$s \leftarrow p_1, \ldots, p_j, \textbf{not } p_{j+1}, \ldots, \textbf{not } p_m \tag{1}$$

head s a context expression (to be defined), body atoms arguments p_i from a parent argumentation framework.

Context Based Argumentation

First step: a language for representing context:

a, b args; v, v' values; $r \in \{skep, cred\}$; $s \in \{grnd, pref, stab\}$

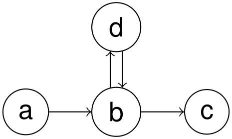
$\mathbf{arg}(a)$ / $\overline{\mathbf{arg}}(a)$	a is a valid (invalid) argument
$\mathbf{att}(a, b)$ / $\overline{\mathbf{att}}(a, b)$	(a, b) is a valid (invalid) attack
$a > b$	a is strictly preferred to b
$\mathbf{val}(a, v)$	the value of a is v
$v > v'$	value v is strictly better than v'
$\mathbf{mode}(r)$	the reasoning mode is r
$\mathbf{sem}(s)$	the chosen semantics is s

Context C: set of context expressions.

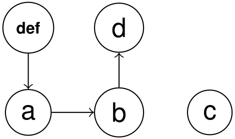
Contexts as Modifiers

What are extensions of AF $\mathcal{A}$ under context C?

C transforms $\mathcal{A}$ to $\mathcal{A}^C$ by (in)validating args and attacks appropriately using new argument **def**:



Let $C = \{\overline{\mathbf{arg}}(a), \mathbf{val}(b, v_1), \mathbf{val}(d, v_2), v_1 > v_2, c > b\}$. $\mathcal{A}^C$ is:



Acceptable Extensions

- Transformation handles statements except **mode** and **sem**.
- These are captured in the following definition:

Acceptable C-extension

Let $\textbf{sem}(s) \in C$. $S \subseteq AR$ is an *acceptable C-extension for $\mathcal{A}$*, if either

1. $\textbf{mode}(\mathit{skep}) \in C$ and $S \cup \{\textbf{def}\}$ is the intersection of all s-extensions of $\mathcal{A}^C$, or
2. $\textbf{mode}(\mathit{cred}) \in C$ and $S \cup \{\textbf{def}\}$ is an s-extension of $\mathcal{A}^C$.

Proposition: Definitions “do the right thing"

Mediators

- Context information may come from parent modules
- Need to “translate" abstract arguments to context statements $\Rightarrow$ use bridge rules
- Also need to guarantee consistency $\Rightarrow$ use consistency method, potentially preferences on parents

Mediator

$\mathcal{A}_1$ and $\mathcal{A}_2, \ldots, \mathcal{A}_k$ AFs. A *mediator for* $\mathcal{A}_1$ *based on* $\mathcal{A}_2, \ldots, \mathcal{A}_k$ is

$$Med = (E_1, R_2, \ldots, R_k, choice)$$

where

- E_1 is a set of context statements for $\mathcal{A}_1$;
- R_i $(2 \leq i \leq k)$ is a set of *bridge rules for* $\mathcal{A}_1$ *based on* $\mathcal{A}_i$;
- $choice \in \{ sub_{\succ}, sub_{sk,\succ}, maj, maj_{sk}\}$, where $\succ$ is a strict partial order on $\{1, \ldots, k\}$.

Mediators, ctd.

Mediator determines consistent context based on

- arguments accepted by parents and
- chosen consistency method.

Acceptable context

Let $Med = (E_1, R_2, \ldots, R_k, choice)$ be a mediator for $\mathcal{A}_1$ based on $\mathcal{A}_2, \ldots, \mathcal{A}_k$. A context C for $\mathcal{A}_1$ is *acceptable wrt. sets of arguments* $S_2, \ldots, S_k$ *of* $\mathcal{A}_2, \ldots, \mathcal{A}_k$, if C is a *choice*-preferred set for $(E_1, R_2(S_2), \ldots, R_k(S_k))$.

Here $R_i(S_i)$ are the context statements derivable from S_i under R_i:

$$\{h \mid h \leftarrow a_1, ..., a_j, \textbf{not } b_1, ..., \textbf{not } b_n \in R_i,\ \textit{each } a_i \in S_i,\ \textit{each } b_m \notin S_i\}$$

The Framework

- Put the pieces together
  - Take collection of context based argument systems
  - Add mediator to each of them
  - Connect them in an arbitrary graph
  - Use mediator to generate consistent context

(Argumentation) Module

Pair $\mathcal{M} = (\mathcal{A}, \mathit{Med})$, where $\mathcal{A}$ is an AF and *Med* a mediator for $\mathcal{A}$ based on some AFs $\mathcal{A}_1, \ldots, \mathcal{A}_k$.

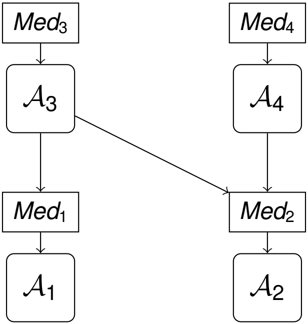
Argumentation context system

Set $\mathcal{F} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ of modules $\mathcal{M}_i = (\mathcal{A}_i, \mathit{Med}_i)$ such that each Med_i is based only on AFs $\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_k}$, where $i_j \in \{1, \ldots, n\}$ *(self-containedness)*.

The Module Graph

Module graph

Digraph $G(\mathcal{F}) = (\mathcal{F}, E)$ where $\mathcal{M}_j \rightarrow \mathcal{M}_i$ in E iff $\mathcal{A}_j$ is among the $\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_k}$ Med_i is based on.



An argumentation context system

Acceptable States

- For each module, pick accepted set of arguments and context
- Must fit together: chosen arguments acceptable given context, chosen context acceptable given chosen arguments of parents

Acceptable state

State $\mathcal{S}$ *of* $\mathcal{F}$: maps each $\mathcal{M}_i = (\mathcal{A}_i, Med_i)$ to $\mathcal{S}(\mathcal{M}_i) = (Acc_i, C_i)$, Acc_i a set of arguments of $\mathcal{A}_i$, C_i a context for $\mathcal{A}_i$.

$\mathcal{S}$ *acceptable*, if

- each Acc_i is an acceptable C_i-extension for $\mathcal{A}_i$, and
- each C_i is an acceptable context for Med_i wrt. all Acc_j for which $G(\mathcal{F})$ has an arc $\mathcal{M}_j \rightarrow \mathcal{M}_i$.

Some Results

- Existence of acceptable states
  - Not guaranteed, even without stable semantics and default negation
  - Guaranteed if $\mathcal{F}$ hierarchic and **sem**(*stab*) does not occur in any mediator.
- Complexity
  - Reasoning tasks related to acceptable states intractable in general.
  - Deciding whether $\mathcal{ACS}$ $\mathcal{F}$ has some acceptable state Σ_3^p-complete.
  - Has lower complexity depending on the various parameters and graph structure.
  - $\mathcal{F}$ hierarchic, modules use grounded semantics and either $sub_{\succ}$ or *maj* $\Rightarrow$ acceptable state computable in polynomial time.
  - Complexity of *C*-extensions dominated by underlying argumentation framework.

4. Generalizing MCS and ACS: An Outlook

- Advantage of MCS: cover large variety of logics
- Advantage of ACS: mediators
  1. include consistency mechanisms integrating conflicting views
  2. allow for KB updates which are more general than just adding premises
  3. can even select the adequate semantics
- Want best of both worlds: Mediator-based MCS

4. Generalizing MCS and ACS: An Outlook

- Advantage of MCS: cover large variety of logics
- Advantage of ACS: mediators
  1. include consistency mechanisms integrating conflicting views
  2. allow for KB updates which are more general than just adding premises
  3. can even select the adequate semantics
- Want best of both worlds: Mediator-based MCS

4. Generalizing MCS and ACS: An Outlook

- Advantage of MCS: cover large variety of logics
- Advantage of ACS: mediators
  1. include consistency mechanisms integrating conflicting views
  2. allow for KB updates which are more general than just adding premises
  3. can even select the adequate semantics
- Want best of both worlds: Mediator-based MCS

MMCS: Context Formalisms

- Need updatable logics.
- Need parameterized semantics.

Context formalism

A context formalism L is a tuple

$$L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{Sem}_L = \{\mathbf{ACC}^i_L\}, \mathbf{U}_L, \mathbf{upd}_L\}$$

- $\mathbf{KB}_L$ and $\mathbf{BS}_L$ as before.
- $\mathbf{Sem}_L$ a set of possible semantics, each $\mathbf{ACC}^i_L : \mathbf{KB}_L \to 2^{\mathbf{BS}_L}$ assigns to a KB a set of acceptable belief sets.
- $\mathbf{U}_L$ a context language with adequate notion of consistency.
- $\mathbf{upd}_L : \mathbf{KB}_L \times 2^{\mathbf{U}_L} \to \mathbf{KB}_L \times \mathbf{Sem}_L$ assigns to a KB and a set of context formulas an updated KB and a semantics.

MMCS: The Rest

- *Acceptable belief set*: E acceptable for KB under context C: $E \in \mathbf{ACC}^i(KB')$ where $\mathbf{upd}(KB, C) = (KB', \mathbf{ACC}^i)$.
- *Mediator*: as in ACS, bridge rules with heads taken from $\mathbf{U}_L$ and bodies elements of belief sets of parents.
- *MMCS*: as in ACS, modules consisting of a KB of particular formalism and corresponding mediator connecting to parents.
- *Acceptable state*: context and belief set for each module such that
  - belief set acceptable under chosen context,
  - context acceptable given belief sets of parents.

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!

5. Conclusions

- Account of recent/ongoing work on multi-context systems.
- Part I: heterogeneous nonmonotonic systems.
- Part II: generalized updates and consistency mechanisms, focus on argumentation.
- Part III: try to capture best of both worlds.
- MCS special case (cum grano salis): updates extensions, no consistency handling
- ACS special case: all formalisms Dung AFs
- MMCS very general and flexible; cover wide range of applications involving multi-agent meta-reasoning.

THANK YOU!