

Propositional Fusion Rules

Anthony Hunter and Rupert Summerton

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

Abstract. In previous papers, we have presented a logic-based framework for merging structured news reports [14, 16, 15]. Structured news reports are XML documents, where the text entries are restricted to individual words or simple phrases, such as names and domain-specific terminology, and numbers and units. We assume structured news reports do not require natural language processing. In this paper, we present propositional fusion rules as a way of implementing logic-based fusion for structured news reports. Fusion rules are a form of scripting language that define how structured news reports should be merged. The antecedent of a fusion rule is a call to investigate the information in the structured news reports and the background knowledge, and the consequent of a fusion rule is a formula specifying an action to be undertaken to form a merged report. It is expected that a set of fusion rules is defined for any given application. We give the syntax and mode of execution for fusion rules, and explain how the resulting actions give a merged report.

1 Introduction

Structured news reports are XML documents, where the text entries are restricted to individual words or simple phrases (such as names and domain-specific terminology), dates, numbers and units. We assume that structured news reports do not require natural language processing. In addition, each tag provides semantic information about the text entries, and a structured news report is intended to have some semantic coherence. To illustrate, news reports on corporate acquisitions can be represented as structured news reports using tags including `buyer`, `seller`, `acquisition`, `value`, and `date`. Structured news reports can be obtained from information extraction systems (e.g. [8]).

In order to merge structured news reports, we need to take account of the contents. Different kinds of content need to be merged in different ways as illustrated in Example 1. There are many further examples we could consider, each with particular features that indicate how the merged report should be formed.

Example 1. Consider the following two conflicting weather reports which are for the same day and same country.

<pre> <report> <source> TV1 </source> <date> 19/3/02 </date> <country> UK </country> <today> showers </today> <windspeed> 10 kph </windspeed> <tomorrow> sun </tomorrow> <regionalreport> <region> South East </region> <maxtemp> 20C </maxtemp> </regionalreport> </report> </pre>	<pre> <report> <source> TV3 </source> <date> 19 March 2002 </date> <country> UK </country> <today> inclement </today> <windspeed> 15 kph </windspeed> <tomorrow> rain </tomorrow> <regionalreport> <region> North West </region> <maxtemp> 18C </maxtemp> </regionalreport> </report> </pre>
---	--

We can merge them so the source is TV1 and TV3, and the weather for today is showers and inclement, and the weather for tomorrow is sun or rain. Also we may wish to take each subtree that is rooted at regionalreport in the input, and put them both in the merged report.

```

<report>
  <source> TV1 and TV3 </source>
  <date> 19.03.02 </date>
  <country> UK </country>
  <today> showers and inclement </today>
  <windspeed> 10 – 15 kph </windspeed>
  <tomorrow> sun or rain </tomorrow>
  <regionalreport>
    <region> South East </region>
    <maxtemp> 20C </maxtemp>
  </regionalreport>
  <regionalreport>
    <region> North West </region>
    <maxtemp> 18C </maxtemp>
  </regionalreport>
</report>

```

An alternative way of merging these reports may be possible if we have a preference for one source over the other. Suppose we have a preference for TV3 in the case of conflict, then we may prefer the textentry rain for the tag tomorrow.

In our approach to merging structured news reports, we draw on domain knowledge to help produce merged reports. The approach is based on fusion rules defined in a logical meta-language. These rules are of the form $\alpha \Rightarrow \beta$, expressing that if α holds, then β is made to hold. So we consider α as a condition to check the information in the structured reports and in the background information, and we consider β as an action to undertake to construct the merged report.

To merge a set of structured news reports, we start with the background knowledge and the information in the news reports to be merged, and attempt to apply all the fusion rules to this information. The application of the fusion rules is then a monotonic process that builds up a set of actions that define how the merged structured news report to be output should be constructed.

2 Structured news reports

We use XML to represent structured news reports. So each structured news report is an XML document, but not vice versa, as defined below. This restriction means that we can easily represent each structured news report by a ground term in classical logic.

Definition 1. Structured news report: *If ϕ is a tagname (i.e an element name), and ψ is a textentry, then $\langle\phi\rangle\psi\langle/\phi\rangle$ is a structured news report. If ϕ is a tagname and $\sigma_1, \dots, \sigma_n$ are structured news reports, then $\langle\phi\rangle\sigma_1\dots\sigma_n\langle/\phi\rangle$ is a structured news report.*

Clearly each structured news report is isomorphic to a tree with the non-leaf nodes being the tagnames and the leaf nodes being the textentries.

Definition 2. Tree: *If $\langle\phi\rangle\psi\langle/\phi\rangle$ is a structured news report, then there is an isomorphic tree that has the root ϕ and a child ψ where ψ is a leaf. If $\langle\phi\rangle\sigma_1\dots\sigma_n\langle/\phi\rangle$ is a structured news report, then there is an isomorphic tree where (1) the root is ϕ and (2) by recursion there is an isomorphic tree ρ_i for each $\sigma_i \in \{\sigma_1, \dots, \sigma_n\}$ where the root of ρ_i is a child of ϕ .*

This isomorphism allows us to give a definition for a branch of a structured news report.

Definition 3. Branch: *Let σ be a structured news report and let ρ be a tree that is isomorphic to σ . A sequence of tagnames $\phi_1/..\phi_n$ is a branch of ρ iff (1) ϕ_1 is the root of ρ and (2) for each i , if $1 \leq i < n$, then ϕ_i is the parent of ϕ_{i+1} . Note, the child of ϕ_n is not necessarily a leaf node. By extension, $\phi_1/..\phi_n$ is a branch of σ iff $\phi_1/..\phi_n$ is a branch of ρ*

When we refer to a subtree (of a structured news report), we mean a subtree formed from the tree representation of the structured news report, where the root of the subtree is a tagname and the leaves are textentries. We formalize this as follows.

Definition 4. Subtree: *Let σ be a structured news report and let ρ be a tree that is isomorphic to σ . A tree ρ' is a subtree of ρ iff (1) the set of nodes in ρ' is a subset of the set of nodes in ρ , and (2) for each node ϕ_i in ρ' , if ϕ_i is the parent of ϕ_j in ρ , then ϕ_j is in ρ' and ϕ_i is the parent of ϕ_j in ρ' . By extension, if σ' is a structured news report, and ρ' is isomorphic to σ' , then we describe σ' as a subtree of σ .*

Each structured news report is also isomorphic with a ground term (of classical logic) where each tagname is a function symbol and each textentry is a constant symbol.

Definition 5. News term: Each structured news report is isomorphic with a ground term (of classical logic) called a news term. This isomorphism is defined inductively as follows: (1) If $\langle\phi\rangle\psi\langle/\phi\rangle$ is a structured news report, where ψ is a textentry, then $\phi(\psi)$ is a news term that is isomorphic with $\langle\phi\rangle\psi\langle/\phi\rangle$; and (2) If $\langle\phi\rangle\psi_1..\psi_n\langle/\phi\rangle$ is a structured news report, and ψ'_1 is a news term that is isomorphic with ψ_1 , ..., and ψ'_n is a news term that is isomorphic with ψ_n , then $\phi(\psi'_1, \dots, \psi'_n)$ is a news term that is isomorphic with $\langle\phi\rangle\psi_1..\psi_n\langle/\phi\rangle$.

Via this isomorphic relationship, we can refer to a branch of a news term by using the branch of the isomorphic structured news report, and we can refer to a subtree of a news term by using the subtree of the isomorphic structured news report.

Definition 6. Let σ be a structured news report and let π be a news term that is isomorphic to σ . By extending Definition 3, $\phi_1/../\phi_n$ is a branch of π iff $\phi_1/../\phi_n$ is a branch of σ . Let σ' be a structured news report and let π' be a news term such that π' is isomorphic to σ' . By extending Definition 4, π' is a subtree of π iff σ' is a subtree of σ .

We now define two functions that allow us to obtain subtrees and textentries from news terms.

Definition 7. Let π be a news term, let π' be a subtree of π , and let ψ be a textentry. If $\phi_1/../\phi_n$ is a branch of π , and the root of π' is ϕ_n , then let $\text{Subtree}(\phi_1/../\phi_n, \pi) = \pi'$, otherwise let $\text{Subtree}(\phi_1/../\phi_n, \pi) = \text{null}$. If $\phi_1/../\phi_n$ is a branch of π , and ψ is the child of ϕ_n , then let $\text{Textentry}(\phi_1/../\phi_n, \pi) = \psi$, otherwise $\text{Textentry}(\phi_1/../\phi_n, \pi) = \text{null}$.

Example 2. Consider the following structured news report.

```

<auctionreport>
  <buyer><firstname>John</firstname><surname>Smith</surname></buyer>
  <property>Lot37</property>
</auctionreport>

```

This can be represented by the following news term:

```
auctionreport(buyer(firstname(John), surname(Smith)), property(Lot37))
```

In this news term, `auctionreport/buyer/firstname` is a branch. If the news term is denoted by π , we have

```

Subtree(auctionreport/buyer,  $\pi$ ) = buyer(firstname(John), surname(Smith))
Subtree(auctionreport/buyer/firstname,  $\pi$ ) = firstname(John)
Textentry(auctionreport/buyer/firstname,  $\pi$ ) = John

```

Definition 8. A skeleton is of the form $\phi(\psi_1, \dots, \psi_n)$ where ϕ is a tagname and ψ_1, \dots, ψ_n are skeletons. If ψ_i is a tagname, then it is a skeleton. A skeleton $\phi(\psi_1, \dots, \psi_n)$ can be regarded as a tree.

A skeleton is equivalent to a structured news report without text entries. It is the underlying structure without the content.

3 Fusion rules

In this paper, we restrict consideration to merging sets of structured news reports of a fixed cardinality. So a set of fusion rules is specified for an application to take exactly n structured news reports as input to produce a merged report as output.

Definition 9. We assume an arbitrary naming of the input reports with names from a set of report names $\{\text{in}_1, \dots, \text{in}_n\}$. The set of **report names** is denoted \mathcal{N} .

Definition 10. Let $\phi_1/../\phi_n$ be a branch, and let $\mu \in \mathcal{N}$ be a report name. A **subtree variable** is denoted $\mu/|\phi_1/.../\phi_n$, and a **textentry variable** is denoted $\mu/|\phi_1/.../\phi_n\#$. A **schema variable** is either a subtree variable or a textentry variable. Let the set of schema variables be denoted \mathcal{S} .

In the following definition, we augment a definition for a classical logic language (where function symbols are not nested) with notation for schema variables which are just placeholders to be instantiated with news terms before logical reasoning.

Definition 11. Let \mathcal{C} be a set of constant symbols, let \mathcal{S} be a set of schema variables, and \mathcal{F} be a set of function symbols. The set of ground terms \mathcal{G} is $\mathcal{C} \cup \{f(c_1, \dots, c_k) \mid f \in \mathcal{F} \text{ and } c_1, \dots, c_k \in \mathcal{C}\}$. The set of terms \mathcal{T} is $\mathcal{C} \cup \{f(d_1, \dots, d_k) \mid f \in \mathcal{F} \text{ and } d_1, \dots, d_k \in \mathcal{C} \cup \mathcal{S}\}$. Let \mathcal{P} be a set of predicate symbols. The set of atoms \mathcal{A} is $\{p(t_1, \dots, t_k) \mid p \in \mathcal{P} \text{ and } t_1, \dots, t_k \in \mathcal{T}\}$. The set of literals \mathcal{L} is $\mathcal{A} \cup \{\neg\gamma \mid \gamma \in \mathcal{A}\}$. The set of ground atoms \mathcal{B} is $\{p(g_1, \dots, g_k) \mid p \in \mathcal{P} \text{ and } g_1, \dots, g_k \in \mathcal{G}\}$. The set of ground literals \mathcal{M} is $\mathcal{B} \cup \{\neg\gamma \mid \gamma \in \mathcal{B}\}$.

Definition 12. Let $\phi(\psi_1, \dots, \psi_n)$ be a news term. The **subterms** of this news term are given by the function Subterms as follows.

$$\text{Subterms}(\phi(\psi_1, \dots, \psi_n)) = \{\phi(\psi_1, \dots, \psi_n)\} \cup \text{Subterms}(\psi_1) \cup \dots \cup \text{Subterms}(\psi_n)$$

For a set of news terms Φ , let $\text{Subterms}(\Phi) = \bigcup_{\phi(\psi_1, \dots, \psi_n) \in \Phi} \text{Subterms}(\phi(\psi_1, \dots, \psi_n))$

We assume that for all possible news terms π , we have $\text{Subterms}(\pi) \subset \mathcal{G}$. We also assume that if $\phi_1/../\phi_n$ is a branch, then it is a constant symbol, called a **branch constant**, and it is in \mathcal{G} . Similarly, if $\phi(\psi_1, \dots, \psi_n)$ is a skeleton, then it is a constant symbol, called a **skeleton constant**, and it is in \mathcal{G} .

Definition 13. A **propositional fusion rule** is of the following form where $\alpha_1, \dots, \alpha_n \in \mathcal{L}$ and $\beta \in \mathcal{A}$.

$$\alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \beta$$

We call $\alpha_1, \dots, \alpha_n$ the **condition literals** and β the **action atom**.

We regard a fusion rule that incorporates schema variables, as a scheme for one or more classical propositional formulae. These propositional formulae are obtained by grounding schema variables as we explain below. We discuss condition literals in Section 3.1 and action literals in Section 3.2.

Example 3. The following is a propositional fusion rule for Example 1.

$$\begin{aligned} & \neg \text{Synonymous}(\text{in}_1//\text{report}/\text{today}\#, \text{in}_2//\text{report}/\text{today}\#) \\ & \wedge \text{Coherent}(\text{in}_1//\text{report}/\text{today}\#, \text{in}_2//\text{report}/\text{today}\#) \\ & \Rightarrow \text{AddText}(\text{Conjunction}(\text{in}_1//\text{report}/\text{today}\#, \\ & \qquad \qquad \qquad \text{in}_2//\text{report}/\text{today}\#), \text{report}/\text{today}) \end{aligned}$$

Definition 14. Let Φ be a set of structured news reports to be merged. Let λ be an assignment (bijection) from the report names \mathcal{N} to Φ . τ is a **valid grounding for a subtree variable** $\mu//\phi_1/\dots/\phi_k$ iff $\tau \in \text{Subterms}(\lambda(\mu))$ and $\text{Subtree}(\phi_1/\dots/\phi_k, \lambda(\mu)) = \tau$. τ is a **valid grounding for a textentry variable** $\mu//\phi_1/\dots/\phi_k\#$ iff $\tau \in \text{Subterms}(\lambda(\mu))$ and $\text{Textentry}(\phi_1/\dots/\phi_k, \lambda(\mu)) = \tau$.

Example 4. Consider the rule in Example 3. For $\text{in}_1//\text{report}/\text{today}\#$, the valid grounding is $\text{Textentry}(\text{report}/\text{today}, \lambda(\text{in}_1))$. This is evaluated to `showers` if we let $\lambda(\text{in}_1)$ refer to the top left structured news report in Example 1. Similarly, the valid grounding for variable $\text{in}_2//\text{report}/\text{today}\#$ is $\text{Textentry}(\text{report}/\text{today}, \lambda(\text{in}_2))$, which is evaluated to `inclement` if we let $\lambda(\text{in}_2)$ refer to the top right structured news report in Example 1.

Definition 15. A **ground fusion rule** is a propositional fusion rule with every schema variable replaced by a valid grounding. Let $\text{Ground}(\delta, \Phi)$ be the set of all ground fusion rules formed from the propositional fusion rule δ where each schema variable in δ is systematically replaced by a valid grounding from $\text{Subterms}(\Phi)$.

Example 5. The ground fusion rule obtained with the fusion rule given in Example 3 with the news reports in Example 1 is the following:

$$\begin{aligned} & \neg \text{Synonymous}(\text{showers}, \text{inclement}) \\ & \wedge \text{Coherent}(\text{showers}, \text{inclement}) \\ & \Rightarrow \text{AddText}(\text{Conjunction}(\text{showers}, \text{inclement}), \text{report}/\text{today}) \end{aligned}$$

Proposition 1. If γ is a ground fusion rule, then γ is a formula of propositional classical logic.

This result means that we have a clear and simple characterization of propositional fusion rules as schema for classical propositional formulae such that once we have the grounded versions of them, reasoning with them is straightforward. As we discuss next, fusion rules provide a bridge between structured news reports and logical reasoning with background knowledge.

3.1 Condition literals

The condition literals in fusion rules relate the contents of structured news reports to the background knowledge. There are many possible condition literals that we could define that relate one or more features from one or more structured news reports to the background knowledge. To illustrate, these literals may include the following kinds: `SameDate(T, T')` where T and T' are news terms with equal date; `SameSource(T, T')`

where T and T' are news terms that refer to the same source; $\text{SameCity}(T, T')$ where T and T' are news terms that refer to the same city; $\text{Synonymous}(T, T')$ where T and T' are news terms that are synonyms; and $\text{Coherent}(T, T')$ where T and T' are news terms that are coherent.

Example 6. Some examples of condition literals (when ground) may include the following.

```

SameDate(date(14Nov01), date(14.11.01))
SameDate(date(day(14), month(11), year(01)), date(14.11.01))
SameCity(city(Mumbai), city(Bombay))
Coherent(snow, sleet)
Coherent(sun, sunny)
Coherent(showers, inclement)
¬Coherent(sun, rain)
¬Coherent(sun, snow)
¬Synonymous(showers, rain)

```

The condition literals are evaluated by querying background knowledge. In the simplest case, the background knowledge may be just a set of ground atoms that hold. However, we would expect the background knowledge would include classical quantified formulae that can be handled using automated reasoning. In any case, the background knowledge is defined by a knowledge engineer building a fusion system for an application.

3.2 Action atoms

Action atoms specify the structure and content for a merged report. In a ground fusion rule $\alpha' \Rightarrow \beta'$, if the ground literals in the antecedent α' hold, then the merged report should meet the specification represented by the ground atom β' . We look at this in more detail in the next section. Here we consider the syntax for action literals.

Each action atom is a member of \mathcal{A} . These incorporate terms based on action functions that take one or more news terms as arguments and return a news term. There are many possibilities for action functions including the following where X and Y are grounded with textentries: $\text{Interval}(X, Y)$ returns an interval $X - Y$ as a textentry; $\text{Conjunction}(X, Y)$ returns a textentry X and Y ; and $\text{Disjunction}(X, Y)$ returns a textentry X or Y . We assume action functions are interpreted in the underlying implementation and return the appropriate textentries for evaluating the action atom.

Example 7. The following ground function, on the left of the = symbol, are rewritten to the news terms on the right:

```

Interval(18C, 25C) = 18 - 25C
Conjunction(TV1, TV3) = TV1 and TV3
Disjunction(sun, rain) = sun or rain

```

We now define a basic set of action atoms. A number of further definitions for action atoms are possible.

Definition 16. *The action atoms are literals that include the following specifying how the merged report should be constructed.*

1. `Initialize($\phi(\psi_1, \dots, \psi_n)$)` where $\phi(\psi_1, \dots, \psi_n)$ is a skeleton constant. The intended action is to start the construction of the merged structured news report with the basic structure being defined by $\phi(\psi_1, \dots, \psi_n)$. The root of the merged report is ϕ .
2. `AddText($T, \phi_1/../\phi_n$)` where T is a textentry, and $\phi_1/../\phi_n$ is a branch constant. The intended action is to add the textentry T as the child to the tagname ϕ_n in the merged report on the branch $\phi_1/../\phi_n$.
3. `AddTree($T, \phi_1/../\phi_n$)` where T is a news term, and $\phi_1/../\phi_n$ is a branch constant. The intended action is to add T to the merged report so that the tagname for the root of T has the parent ϕ_n on the branch $\phi_1/../\phi_n$.

The action atoms are specifications that are intended to be made to hold by producing a merged report that satisfies the specification.

Example 8. Consider the action literal in the consequent of Example 3.

`AddText(Conjunction(showers, inclement), report/today)`

The term `Conjunction(showers, inclement)` is rewritten by the system to the term `showers and inclement`, and so the action literal is now the following:

`AddText(showers and inclement, report/today)`

This specifies that the textentry should be `showers and inclement` in the merged report for tagname `today` on the branch `report/today`, as obtained in Example 1

4 Rule execution

In order to use a set of fusion rules, we need to be able to execute them with background knowledge and a set of structured news reports.

Definition 17. *A fusion call is a triple (Δ, Γ, Φ) where Γ is a set of fusion rules, Δ is a background knowledgebase (a set of classical first-order formulae), and Φ is a set of structured news reports.*

To merge some reports, the fusion rules are ground with the structured news reports, and then a form of modus ponens is exhaustively applied together with the background knowledge.

Definition 18. *Let (Δ, Γ, Φ) be a fusion call.*

$$\text{Actions}(\Delta, \Gamma, \Phi) = \{\beta' \mid \alpha' \Rightarrow \beta' \in \text{Ground}(\delta, \Phi) \text{ and } \delta \in \Gamma \text{ and } \Delta \vdash \alpha'\}$$

where \vdash is the classical consequence relation.

Example 9. A fusion call with an appropriate set of fusion rules and the pair of news reports given in the top of Example 1 together with appropriate background knowledge can give the following action atoms:

```

Initialize(report(source, date, country, today, windspeed, tomorrow))
AddText(Conjunction(TV1, TV3), report/source)
AddText(19.03.02, report/date)
AddText(UK, report/country)
AddText(Conjunction(showers, inclement), report/today)
AddText(Interval(18Kph, 15Kph), report/windspeed)
AddText(Disjunction(sun, rain), report/tomorrow)
AddTree(regionalreport(region(SouthEast), maxtemp(20C)), report)
AddTree(regionalreport(region(NorthWest), maxtemp(18C)), report)

```

These action atoms specify the merged report given in the bottom of Example 1.

Proposition 2. *Let (Δ, Γ, Φ) be a fusion call. $\text{Actions}(\Delta, \Gamma, \Phi)$ is a finite set iff Γ is a finite set of fusion rules.*

Given a fusion call (Δ, Γ, Φ) , the set $\text{Actions}(\Delta, \Gamma, \Phi)$ is input to an algorithm for constructing a merged report. The minimum we expect of a set of action atoms is that a merged report can be produced that meets the specification. We formalize this as follows.

Definition 19. *Let (Δ, Γ, Φ) be a fusion call and let σ be a structured news report.*

σ meets $\text{Actions}(\Delta, \Gamma, \Phi)$ iff $\forall \beta \in \text{Actions}(\Delta, \Gamma, \Phi)$ σ meets β

where σ is isomorphic with a news term π such that

σ meets $\text{Initialize}(\phi(\psi_1, \dots, \psi_n))$
iff each branch of $\sigma(\psi_1, \dots, \psi_n)$ is a branch of π

σ meets $\text{AddText}(T, \phi_1/../\phi_n)$
iff $\text{Textentry}(\phi_1/../\phi_n, \pi) = T$

σ meets $\text{AddTree}(T, \phi_1/../\phi_n)$
iff $\text{Subtree}(\phi_1/../\phi_n/\phi_{n+1}, \pi) = T$ and the root of T is ϕ_{n+1}

However, the meets relation is a little too relaxed in the sense that a report may meet an action sequence but may also include extra information that has not been specified.

Definition 20. *The **matches** relation is defined as follows where σ is a structured news report and (Δ, Γ, Φ) is a fusion call: σ matches $\text{Actions}(\Delta, \Gamma, \Phi)$ iff σ meets $\text{Actions}(\Delta, \Gamma, \Phi)$ and there is no σ' s.t. (σ' meets $\text{Actions}(\Delta, \Gamma, \Phi)$ and $\text{Subterms}(\sigma') \subset \text{Subterms}(\sigma)$).*

The matches relation identifies the minimal structured news report(s) that meet(s) the action sequence. In other words, it identifies the news reports that do not include any superfluous information.

Definition 21. A fusion call (Δ, Γ, Φ) is **complete** iff there is a ψ for which $\text{Initialize}(\psi)$ atom is in $\text{Actions}(\Delta, \Gamma, \Phi)$ and for all branches $\phi_1/../\phi_n$ of ψ there is an action atom $\text{AddText}(T, \phi_1/../\phi_n)$ in $\text{Actions}(\Delta, \Gamma, \Phi)$ for some T .

In other words, an action sequence is complete if it is not the case that the structured news report that results has missing textentries.

Definition 22. A fusion call (Δ, Γ, Φ) is **consistent** iff (1) there is exactly one $\text{Initialize}(\psi)$ atom in $\text{Actions}(\Delta, \Gamma, \Phi)$; and (2) if $\text{AddText}(T, \phi_1/../\phi_n)$ is in $\text{Actions}(\Delta, \Gamma, \Phi)$, then for all $\text{AddText}(T', \phi_1/../\phi_n)$ in $\text{Actions}(\Delta, \Gamma, \Phi)$ $T = T'$;

Proposition 3. A fusion call (Δ, Γ, Φ) is complete and consistent iff there is a structured news report π such that π meets $\text{Actions}(\Delta, \Gamma, \Phi)$.

Definition 23. A fusion system (Δ, Γ) is **well-behaved** iff for all Φ either (Δ, Γ, Φ) is a complete and consistent fusion call or there is no ψ such that the $\text{Initialize}(\psi)$ atom is in $\text{Actions}(\Delta, \Gamma, \Phi)$.

This means the fusion rules in a fusion system need to be engineered so that exactly one Initialize atom is obtained for any set of structured news reports Φ that are to be covered by the fusion system, and no Initialize atom is to be obtained for any set of structured news reports Φ that are not to be covered by the fusion system.

The set of action atoms that we have defined in this paper is only part of the range of possible action atoms. We have implemented others including $\text{ExtendTree}(T, \phi_1/../\phi_n)$ where T is a news term, and $\phi_1/../\phi_n$ is a branch constant, and the intended action is to extend the merged report with T so that the tagname for the root of T is ϕ_n on the branch $\phi_1/../\phi_n$. We intend to extend the range of implemented functions to include functions based on voting strategies so that for example if the majority of news reports input have a particular textentry on a particular branch, then the merged report will have that textentry on that branch.

5 Discussion

The definition for a fusion call suggests an implementation based on existing automated reasoning technology and on XML programming technology. Once information is in the form of XML documents, a number of technologies for managing and manipulating information in XML are available. We have developed a prototype implementation in Java for executing executing fusion rules that are marked up in `FusionRuleML` and constructing the merged reports [17]. Background knowledge is handled in a Prolog system and this is queried by the Java implementation.

Our logic-based approach differs from other logic-based approaches for handling inconsistent information such as belief revision theory (e.g. [11, 9, 18, 20]) and knowledgebase merging (e.g. [19, 1]). These proposals are too simplistic in certain respects for handling news reports. Each of them has one or more of the following weaknesses: (1) One-dimensional preference ordering over sources of information — for news reports we require finer-grained preference orderings; (2) Primacy of updates in belief revision — for news reports, the newest reports are not necessarily the best reports;

and (3) Weak merging based on a meet operator — this causes unnecessary loss of information. Furthermore, none of these proposals incorporate actions on inconsistency or context-dependent rules specifying the information that is to be incorporated in the merged information, nor do they offer a route for specifying how merged reports should be composed.

Merging information is also an important topic in database systems. A number of proposals have been made for approaches based in schema integration (e.g. [24]), the use of global schema (e.g. [12]), and conceptual modelling for information integration based on description logics [4, 3, 10, 23, 2]. These differ from our approach in that they do not seek an automated approach that uses domain knowledge for identifying and acting on inconsistencies. Heterogeneous and federated database systems are relevant, but they do not identify and act on inconsistency in a context-sensitive way [26, 22, 6], though there is increasing interest in bringing domain knowledge into the process (e.g. [5, 27]). Also relevant is revision programming, a logic-based framework for describing and enforcing database constraints [21].

Our approach also goes beyond other technologies for handling news reports. The approach of wrappers offers a practical way of defining how heterogeneous information can be merged (see for example [13, 7, 25]). However, there is little consideration of problems of conflicts arising between sources. Our approach therefore goes beyond these in terms of formalizing reasoning with inconsistent information and using this to analyse the nature of the news report and for formalizing how we can act on inconsistency.

Acknowledgements

The authors wish to thank Weiru Liu and the referees for helpful feedback on the paper.

References

1. C Baral, S Kraus, J Minker, and V Subrahmanian. Combining knowledgebases consisting of first-order theories. *Computational Intelligence*, 8:45–71, 1992.
2. S Bergamaschi, S Castano, M Vincini, and D Beneventano. Semantic integration of heterogeneous information sources. *Data and Knowledge Engineering*, 36:215–249, 2001.
3. D Calvanese, G De Giacomo, M Lenzerini, D Nardi, and R Rosati. Description logic framework for information integration. In *Proceedings of the 6th Conference on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13. Morgan Kaufmann, 1998.
4. D Calvanese, G De Giacomo, M Lenzerini, D Nardi, and R Rosati. Source integration in data warehousing. In *Proceedings of the 9th International Workshop on Database and Expert Systems (DEXA'98)*, pages 192–197. IEEE Computer Society Press, 1998.
5. L Cholvy. Reasoning with data provided by federated databases. *Journal of Intelligent Information Systems*, 10:49–80, 1998.
6. L Cholvy and S Moral. Merging databases: Problems and examples. *International Journal of Intelligent Systems*, 10:1193–1221, 2001.
7. W Cohen. A web-based information system that reasons with structured collections of text. In *Proceedings of Autonomous Agents'98*, 1998.
8. J Cowie and W Lehnert. Information extraction. *Communications of the ACM*, 39:81–91, 1996.

9. D Dubois and H Prade, editors. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 3. Kluwer, 1998.
10. E Franconi and U Sattler. A data warehouse conceptual data model for multidimensional aggregation. In S Gatzju, M Jeusfeld, M Staudt, and Y Vassiliou, editors, *Proceedings of the Workshop in Design and Management of Data Warehouses*, 1999.
11. P Gardenfors. *Knowledge in Flux*. MIT Press, 1988.
12. G Grahne and A Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)*, Lecture Notes in Computer Science. Springer, 1999.
13. J Hammer, H Garcia-Molina, S Nestorov, and R Yerneni. Template-based wrappers in the TSIMMIS system. In *Proceedings of ACM SIGMOD'97*. ACM, 1997.
14. A Hunter. Merging potentially inconsistent items of structured text. *Data and Knowledge Engineering*, 34:305–332, 2000.
15. A Hunter. Logical fusion rules for merging structured news reports. *Data and Knowledge Engineering*, 42:23–56, 2002.
16. A Hunter. Merging structured text using temporal knowledge. *Data and Knowledge Engineering*, 41:29–66, 2002.
17. A Hunter and R Summerton. FusionRuleML: Representing and executing fusion rules. Technical report, UCL Department of Computer Science, 2002.
18. H Katsuno and A Mendelzon. On the difference between updating a knowledgebase and revising it. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference (KR'91)*, pages 387–394. Morgan Kaufmann, 1991.
19. S Konieczny and R Pino Perez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 488–498. Morgan Kaufmann, 1998.
20. P Liberatore and M Schaerf. Arbitration (or how to merge knowledgebases). *IEEE Transactions on Knowledge and Data Engineering*, 10:76–90, 1998.
21. V Marek and M Truszczyński. Revision programming. *Theoretical Computer Science*, 190:241–277, 1998.
22. A Motro. Cooperative database systems. *International Journal of Intelligent Systems*, 11:717–732, 1996.
23. N Paton, R Stevens, P Baker, C Goble, S Bechhofer, and A Brass. Query processing in the TAMBIS bioinformatics source integration system. In *Proceedings of the 11th International Conference on Scientific and Statistical Databases*, 1999.
24. A Poulouvasilis and P McBrien. A general formal framework for schema transformation. *Data and Knowledge Engineering*, 28:47–71, 1998.
25. A Sahuguet and F Azavant. Building light-weight wrappers for legacy web data-sources using W4F. In *Proceedings of the International Conference on Very Large Databases (VLDB'99)*, 1999.
26. A Sheth and J Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22:183–236, 1990.
27. K Smith and L Obrst. Unpacking the semantics of source and usage to perform semantic reconciliation in large-scale information systems. In *ACM SIGMOD RECORD*, volume 28, pages 26–31, 1999.