

Introduction to Information Integration

Part I: Classical Approaches Basic Problem & Common Solution Approaches

Uwe Keller
uwe.keller@deri.org

Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria

SENSE Technical Project Meeting, DERI Innsbruck

3rd May 2006



Outline

- 1 Classical Information Integration
 - The Problem
 - Basic Solution
 - Abstract Model of Data Integration Systems
- 2 Defining a Data Integration System
 - Defining the Mapping
- 3 Further aspects of Information Integration



Outline

- 1 **Classical Information Integration**
 - **The Problem**
 - Basic Solution
 - Abstract Model of Data Integration Systems
- 2 Defining a Data Integration System
 - Defining the Mapping
- 3 Further aspects of Information Integration



Welcome in the Information Age

The best world to be imagined

**Consulting an “oracle” or using a single “almighty”
information source**



Characteristics of the ideal world

Single „omnipotent” source of information

- **No** heterogenities in data representation
- **No** distribution of data
- Source stores **complete information** (i.e. all available information) about all entities
- **No** redundancy & **No** inconsistency

Benefits

- Query answering becomes **relatively easy!**



But the real world is different!

Often, there is no such desired “oracle” for answering the questions



Characteristics of a real-world scenario

Most of the nice properties are gone!

- **Heterogeneties** in data representation (datamodel)
- Datasources have **different query processing capabilities**
- **Distribution** of data
- Datasources are **autonomous** (federated: **loss of control**)
- **Incompleteness** of Sources
- **Redundancy** between sources & **inconsistency**



Characteristics of a real-world scenario

If this so complicated, why does this scenario occur at all?

- Unexpected (or unforeseen) **need for combination** of different **existing** information **sources**
- Mergers of departments & organizations
- Establishing of loose cooperation between partners (for some period of time)



How does this relate to the SEnSE environment?

- Engineers use various **different specialized tools** (over years already) to design elements of a product. This can **not** be changed! Even stronger: the **SEnSE system** needs to be **transparent** to these legacy systems.

⇒ **Heterogeneous dataformats & datamodel**

- Engineering is a **group effort**, **distributed teams**, team members might belong to **different departments or even organizations**.
- Information about parts of a product are distributed over various sources. Not every partner has the same access to the sources.

⇒ **Autonomous and location-wise distributed sources**

- Each source gives only **part of the available (relevant) information** on such an element

⇒ **Combination of information needed**



How does this relate to the SEnSE environment?

- Engineers use various **different specialized tools** (over years already) to design elements of a product. This can **not** be changed! Even stronger: the **SEnSE system** needs to be **transparent** to these legacy systems.

⇒ **Heterogeneous dataformats & datamodel**

- Engineering is a **group** effort, **distributed teams**, team members might belong to **different departments or even organizations**.
- Information about parts of a product are distributed over various sources. Not every partner has the same access to the sources.

⇒ **Autonomous and location-wise distributed sources**

- Each source gives only **part of the available (relevant) information** on such an element

⇒ **Combination of information needed**



How does this relate to the SEnSE environment?

- Engineers use various **different specialized tools** (over years already) to design elements of a product. This can **not** be changed! Even stronger: the **SEnSE system** needs to be **transparent** to these legacy systems.

⇒ **Heterogeneous dataformats & datamodel**

- Engineering is a **group** effort, **distributed teams**, team members might belong to **different departments or even organizations**.
- Information about parts of a product are distributed over various sources. Not every partner has the same access to the sources.

⇒ **Autonomous and location-wise distributed sources**

- Each source gives only **part of the available (relevant) information** on such an element

⇒ **Combination of information needed**



Outline

- 1 **Classical Information Integration**
 - The Problem
 - **Basic Solution**
 - Abstract Model of Data Integration Systems
- 2 Defining a Data Integration System
 - Defining the Mapping
- 3 Further aspects of Information Integration



How to overcome the problem?

Computer Science or „The Art of Abstraction“: Pretend that the world would be ideal! (Virtualization of Resources)



Basic Solution: Elements

Intuitively

- Source schemas S_i describe the structure of the sources, where the **real** data are
- Global schema \mathcal{G} provides a reconcile, integrated, **virtual** view of the underlying sources (no actual data in materialized form!)
- Mapping \mathcal{M}_i represents **assertions** that establish connection between the elements of S_i and those in \mathcal{G}

Queries to \mathcal{I}

- posed in terms of the global schema \mathcal{G} over a query language \mathcal{L}_Q over alphabet \mathcal{A}_G
- Specification of which data to extract from the virtual database



Data integration

Some “definitions”

- [Len02] **Data integration** is the problem of **combining** data residing at **different sources**, and providing the user with a **unified view** of these data
- [Lev99] The **data integration problem** is to provide uniform access to multiple heterogeneous information sources available online (e.g. databases on the WWW). . . . The data integration problem is complicated by the fact that (1) sources contain closely related and overlapping data, (2) data is stored in multiple datamodels and schemas, and (3) data sources have different query processing capabilities.



Data integration

Some “definitions”

- [Len02] **Data integration** is the problem of **combining** data residing at **different sources**, and providing the user with a **unified view** of these data
- [Lev99] The **data integration problem** is to provide uniform access to multiple heterogeneous information sources available online (e.g. databases on the WWW). . . . The data integration problem is complicated by the fact that (1) sources contain closely related and overlapping data, (2) data is stored in multiple datamodels and schemas, and (3) data sources have different query processing capabilities.



Data integration systems

- Invent a **(virtual) common schema**
 - **User** poses queries against this **unified view** of the data
- ⇒ Heterogeneity and distribution are **transparent** to the user
- **Data integration system** computes the “right” queries for the single (relevant) sources to retrieve information bits that are needed to answer the question (**Query reformulation**)
 - **Correctness** of retrieved answers needed, **Completeness** desirable
- ⇒ **Requires** to analyse the connection between the data sources and the unified view (**Mappings M_i**)



Data integration systems

- Invent a **(virtual) common schema**
- **User** poses queries against this **unified view** of the data
- ⇒ Heterogeneity and distribution are **transparent** to the user
- **Data integration system** computes the “right” queries for the single (relevant) sources to retrieve information bits that are needed to answer the question (**Query reformulation**)
- **Correctness** of retrieved answers needed, **Completeness** desirable
- ⇒ **Requires** to analyse the connection between the data sources and the unified view (**Mappings M_i**)



Data integration systems

- Invent a **(virtual) common schema**
- **User** poses queries against this **unified view** of the data
- ⇒ Heterogeneity and distribution are **transparent** to the user
- **Data integration system** computes the “right” queries for the single (relevant) sources to retrieve information bits that are needed to answer the question (**Query reformulation**)
- **Correctness** of retrieved answers needed, **Completeness** desirable
- ⇒ **Requires** to analyse the connection between the data sources and the unified view (**Mappings M_i**)



Outline

- 1 **Classical Information Integration**
 - The Problem
 - Basic Solution
 - **Abstract Model of Data Integration Systems**
- 2 Defining a Data Integration System
 - Defining the Mapping
- 3 Further aspects of Information Integration



Data integration systems

An abstract model [Len02]

A **data integration system** \mathcal{I} is a triple $(\mathcal{G}, \mathcal{S}, \mathcal{M})$, where

- \mathcal{G} is a **global schema** in a language $\mathcal{L}_{\mathcal{G}}$ over an alphabet $\mathcal{A}_{\mathcal{G}}$. *The alphabet comprises a symbol for each (conceptual) element in \mathcal{G} (e.g. relation if \mathcal{G} is relational, class if \mathcal{G} is OO etc.)*
- \mathcal{S} is a **source schema** in a language $\mathcal{L}_{\mathcal{S}}$ over an alphabet $\mathcal{A}_{\mathcal{S}}$
- \mathcal{M} is the **mapping** between \mathcal{G} and \mathcal{S} , constitute by a set of assertions of the forms $q_{\mathcal{S}} \rightsquigarrow q_{\mathcal{G}}$ or $q_{\mathcal{G}} \rightsquigarrow q_{\mathcal{S}}$, where $q_{\mathcal{S}}$ and $q_{\mathcal{G}}$ are two queries of the same arity, resp. over \mathcal{S} and \mathcal{G} . Queries $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{M}, \mathcal{S}}$ and $q_{\mathcal{G}} \in \mathcal{L}_{\mathcal{M}, \mathcal{G}}$

Intuitively: $q_{\mathcal{S}} \rightsquigarrow q_{\mathcal{G}}$ specifies that concept represented by $q_{\mathcal{S}}$ over sources corresponds to the concept in the global schema represented by $q_{\mathcal{G}}$

Note: This definition considers only a **single** data source!



Data integration systems

Semantics of the abstract model [Len02]

What does the abstract model $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ mean ?

- Consider a **source database** \mathcal{D} for \mathcal{I} , i.e a database that conforms to local schema \mathcal{S}
- Based on \mathcal{D} we specify the **information content of the global schema** \mathcal{G} :
 - A **global database for** \mathcal{I} is any database \mathcal{B} for \mathcal{G}
 - Global DB \mathcal{B} for \mathcal{I} is **legal wrt. \mathcal{D}** , if (a) \mathcal{B} **conforms to \mathcal{G}** , i.e. satisfies all the constraints of \mathcal{G} , and (b) \mathcal{B} **satisfies the mapping \mathcal{M}** wrt. the specific local DB \mathcal{D}
 - Note: **Mappings** are interpreted as **declarative specs** above!
- In general, a **global DB \mathcal{B}** which is legal for \mathcal{I} wrt. \mathcal{D} is **not unique**
- Hence: Strong relation between data integration systems and databases with incomplete knowledge (or knowledge bases)



Data integration systems

Semantics of the abstract model [Len02]

What does a **query q posed against $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$** mean ?

- Query q of arity n and \mathcal{D} database, then $q^{\mathcal{D}}$ denotes the set of n -tuples in \mathcal{D} that satisfy q .
- Given a source database \mathcal{D} for \mathcal{I} , then **answer $q^{\mathcal{I}, \mathcal{D}}$** to query q in \mathcal{I} wrt. local DB \mathcal{D} is the set of n -tuples of objects s.t. $t \in \mathcal{B}$ for every global DB \mathcal{B} that is legal for \mathcal{I} wrt. the local DB \mathcal{D} .
- The set $q^{\mathcal{I}, \mathcal{D}}$ is called the set of **certain answers** to q in \mathcal{I} wrt. the local DB \mathcal{D} :

$$\text{certain}^{\mathcal{I}, \mathcal{D}}(q) = \{\vec{t} \in U^n / \forall \mathcal{B} \in \text{sem}^{\mathcal{D}}(\mathcal{I}) : \vec{t} \in q^{\mathcal{B}}\}$$

Query answering in DIS is **logical entailment checking!**



Data integration systems

Semantics of the abstract model [Len02]

What does a **query q** posed against $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ mean ?

Alternative semantics to queries: possible answers

$$\text{possible}^{\mathcal{I}, \mathcal{D}}(q) = \{\vec{t} \in U^n / \exists \mathcal{B} \in \text{sem}^{\mathcal{D}}(\mathcal{I}) : \vec{t} \in q^{\mathcal{B}}\}$$

Query answering in DIS under this semantics is a satisfiability problem!



Parameters of the model

Definition is **general enough to capture almost all approaches** in the literature!

Specific approach depends on

- Characteristics of the mapping (Mapping approach)
- Nature and expressive power of the various schema and query languages.

Example

- \mathcal{L}_G simple (only relation definitions) vs. allow for additional specification of various integrity constraints over symbols in \mathcal{A}_G .
- Type of \mathcal{L}_S : relational, semi-structured, object-oriented, ...



Outline

- 1 Classical Information Integration
 - The Problem
 - Basic Solution
 - Abstract Model of Data Integration Systems
- 2 Defining a Data Integration System
 - Defining the Mapping
- 3 Further aspects of Information Integration



How to define the mapping?

The agony of choice: there is **no exclusive approach!**

- **Local-as-view**: Define the single elements of a source schema \mathcal{S}_i in terms of the elements in the global schema \mathcal{G}
- **Global-as-view**: Define how the single elements in the global schema \mathcal{G} can be computed in terms of the local schemas \mathcal{S}_i
- **Local-and-Global-as-view**: Combination of LAV and GAV.

Each approach has its own advantages and drawbacks, and hence are **architectural decisions for a data integration system!**



Local-as-view

How it works

- \mathcal{M} associates to each element $s \in \mathcal{S}$ a query $q_{\mathcal{G}}$ over \mathcal{G} , i.e.
 $\mathcal{L}_{\mathcal{M}, \mathcal{S}} = \mathcal{A}_{\mathcal{S}}$
- Therefore, \mathcal{M} consists only of assertions of the form $s \rightsquigarrow q_{\mathcal{G}}$
- For each (conceptual) element $s \in \mathcal{S}$, there is a respective assertion $s \rightsquigarrow q_{\mathcal{G}}$

Basic Intention

Content of each source \mathcal{S} is characterized in terms of the unified view.
If \mathcal{G} is an ontology, this can be seen as explaining the semantics of the source elements in \mathcal{S})



Local-as-view

Example

Global Schema \mathcal{G} : *movie*(*Title*, *Year*, *Director*)
 european(*Director*)
 review(*Title*, *Critique*)

Source S_1 : *r*₁(*Title*, *Year*, *Director*) , since 1960, Europ. Direct.

Source S_2 : *r*₂(*Title*, *Critique*) , since 1990

Query q *Title and critique of movies in 1998*
 $\{(T, R) / \exists D : \textit{movie}(T, 1998, D) \wedge \textit{review}(T, R)\}$



Local-as-view

Example

\mathcal{G} : *movie*(*Title*, *Year*, *Director*)
european(*Director*)
review(*Title*, *Critique*)

LAV: Associated to source relations we have **views over global schema**

$S_1: r_1(T, Y, D) \rightsquigarrow \{(T, Y, D) / \text{movie}(T, Y, D) \wedge \text{european}(D) \wedge Y \geq 1960\}$

$S_2: r_2(T, C) \rightsquigarrow \{(T, R) / \text{movie}(T, Y, D) \wedge \text{review}(T, R) \wedge Y \geq 1990\}$

Query $q = \{(T, R) / \exists D : \text{movie}(T, 1998, D) \wedge \text{review}(T, R)\}$ is processed by some rewriting procedure (based on **inferences**) to rephrase the atoms of the global schema in terms of atoms over the local schemas. **Match Query Predicate with View Definitions** E.g. here:

$$q = \{(T, R) / \exists D : r_1(T, 1998, D) \wedge r_2(T, R)\}$$


Local-as-view

Distinguish **types of source element views** $s \rightsquigarrow q_G$

- **Sound views:** S_i provides with s any subset of the tuples satisfying the view q_G (**default type**)
- **Complete views:** S_i provides with s any superset of the tuples satisfying the view q_G (i.e. not necessarily relevant ones)
- **Exact views:** S_i provides with s exactly the tuples satisfying the view q_G

Sound views Given a source database \mathcal{D} , from the fact that a tuple \vec{t} is in $s^{\mathcal{D}}$ one can conclude that it satisfies the associated view over the global schema, while from the fact that \vec{t} is not in $s^{\mathcal{D}}$ one can not conclude that that it does not satisfy the corresponding view.

Formally, a global DB \mathcal{B} satisfies the assertion $s \rightsquigarrow q_G$ wrt \mathcal{D} if $s^{\mathcal{D}} \subseteq q_G^{\mathcal{B}}$. I.e. **consider $s \rightsquigarrow q_G$ logically as $\forall \vec{x} : s(\vec{x}) \rightarrow q_G(\vec{x})$**



Local-as-view

Complete views Given a source database \mathcal{D} , from the fact that a tuple \vec{t} is not in $s^{\mathcal{D}}$ one can conclude that it does not satisfies the associated view over the global schema, while from the fact that \vec{t} is in $s^{\mathcal{D}}$ one can not conclude that that it satisfies the corresponding view.

Formally, a global DB \mathcal{B} satisfies the assertion $s \rightsquigarrow q_G$ wrt \mathcal{D} if $s^{\mathcal{D}} \supseteq q_G^{\mathcal{B}}$. I.e. **consider $s \rightsquigarrow q_G$ logically as $\forall \vec{x} : s(\vec{x}) \leftarrow q_G(\vec{x})$**

Exact views Formally, a global DB \mathcal{B} satisfies the assertion $s \rightsquigarrow q_G$ wrt \mathcal{D} if $s^{\mathcal{D}} = q_G^{\mathcal{B}}$. I.e. **consider $s \rightsquigarrow q_G$ logically as $\forall \vec{x} : s(\vec{x}) \leftrightarrow q_G(\vec{x})$**



Local-as-view

Benefits and drawbacks

Benefits

- Favours **extensibility** of the data integration system: **adding new sources is easy**, since it means enriching the mapping \mathcal{M} wrt. the new source schemas only. Other mapping elements do not have to be touched at all!

Drawbacks

- **Assumes** that the **global schema is stable** (does not change). In case of changes to \mathcal{G} **all** mappings in \mathcal{M} have to be checked and perhaps reworked. Changing \mathcal{G} from time to time only feasible for a small number of sources (or more exactly small size of the mapping)



Global-as-view

How it works

- \mathcal{M} associates to each element $g \in \mathcal{G}$ a query q_S over S , i.e.
 $\mathcal{L}_{\mathcal{M},g} = \mathcal{A}_g$
- Therefore, \mathcal{M} consists only of assertions of the form $g \rightsquigarrow q_S$
- For each (conceptual) element $g \in \mathcal{G}$, there is a respective assertion $g \rightsquigarrow q_S$

Basic Intention

Content of each element g of the global schema \mathcal{G} is characterized in terms of a view q_f over the sources. In a sense, the mapping **explicitly tells** the system **how to use the sources** to retrieve the data.



Global-as-view

Example

\mathcal{G} : *movie*(*Title*, *Year*, *Director*)
european(*Director*)
review(*Title*, *Critique*)

GAV: Associated to elements in \mathcal{G} we have **views over local schemas**

$$\begin{aligned} \textit{movie}(T, Y, D) &\rightsquigarrow \{(T, Y, D) / r_1(T, Y, D)\} \\ \textit{european}(D) &\rightsquigarrow \{(D) / \exists T, Y : r_1(T, Y, D)\} \\ \textit{review}(\textit>Title, \textit>Critique) &\rightsquigarrow \{(T, R) / r_2(T, R)\} \end{aligned}$$

Query $q = \{(T, R) / \exists D : \textit{movie}(T, 1998, D) \wedge \textit{review}(T, R)\}$ is processed by unfolding, i.e. expanding each atom according to its associated definition in \mathcal{M} , to come up with a description in terms of local schemas. E.g. here:

$$q = \{(T, R) / \exists D : r_1(T, 1998, D) \wedge r_2(T, R)\}$$



Global-as-view

Sound, complete and exact mapping view definitions

Similar to LAV: Associate type to definitions of elements $g \in \mathcal{G}$

Sound views $g \rightsquigarrow q_S$ Formally, a global DB \mathcal{B} satisfies the assertion $g \rightsquigarrow q_S$ wrt \mathcal{D} if $g^{\mathcal{D}} \supseteq q_G^{\mathcal{B}}$. Logically, this can be represented as $\forall \vec{x} : g(\vec{x}) \rightarrow q_S(\vec{x})$

Complete views $g \rightsquigarrow q_S$ Formally, a global DB \mathcal{B} satisfies the assertion $g \rightsquigarrow q_S$ wrt \mathcal{D} if $g^{\mathcal{D}} \subseteq q_G^{\mathcal{B}}$. Logically, this can be represented as $\forall \vec{x} : g(\vec{x}) \leftarrow q_S(\vec{x})$

Exact views $g \rightsquigarrow q_S$ Formally, a global DB \mathcal{B} satisfies the assertion $g \rightsquigarrow q_S$ wrt \mathcal{D} if $g^{\mathcal{D}} = q_G^{\mathcal{B}}$. Logically, this can be represented as $\forall \vec{x} : g(\vec{x}) \leftrightarrow q_S(\vec{x})$



Global-as-view

Common observations

- Often **implicitly assumed** in GAV: **exact views**

If all views are exact and a first-order query language is used as $\mathcal{L}_{\mathcal{M},\mathcal{S}}$, a GAV integration system enjoys the **„single database property“**: the single global database \mathcal{B} obtained by associating each element g the set of tuples computed by q_S .

⇒ Motivates often that GAV is easier than LAV. However, **„single database property“** only applies in this restricted setting!

- Possibility to express constraints in \mathcal{G} greatly **enhances modelling power** of of GAV systems, especially when data model of \mathcal{G} is conceptual or semantically rich (ontology)
- Most of the current data integration systems (2002) followed the GAV approach



Global-as-view

Common observations

- Often **implicitly assumed** in GAV: **exact views**

If all views are exact and a first-order query language is used as $\mathcal{L}_{\mathcal{M},\mathcal{S}}$, a GAV integration system enjoys the **„single database property“**: the single global database \mathcal{B} obtained by associating each element g the set of tuples computed by q_S .

⇒ Motivates often that GAV is easier than LAV. However, **„single database property“** only applies in this restricted setting!

- Possibility to express constraints in \mathcal{G} greatly **enhances modelling power** of of GAV systems, especially when data model of \mathcal{G} is conceptual or semantically rich (ontology)
- Most of the current data integration systems (2002) followed the GAV approach



Global-as-view

Common observations

- Often **implicitly assumed** in GAV: **exact views**

If all views are exact and a first-order query language is used as $\mathcal{L}_{\mathcal{M},\mathcal{S}}$, a GAV integration system enjoys the **„single database property“**: the single global database \mathcal{B} obtained by associating each element g the set of tuples computed by q_S .

⇒ Motivates often that GAV is easier than LAV. However, **„single database property“** only applies in this restricted setting!

- Possibility to express constraints in \mathcal{G} greatly **enhances modelling power** of GAV systems, especially when data model of \mathcal{G} is conceptual or semantically rich (ontology)
- Most of the current data integration systems (2002) followed the GAV approach



Global-as-view

Common observations

- Often **implicitly assumed** in GAV: **exact views**

If all views are exact and a first-order query language is used as $\mathcal{L}_{\mathcal{M},\mathcal{S}}$, a GAV integration system enjoys the „**single database property**“: the single global database \mathcal{B} obtained by associating each element g the set of tuples computed by q_S .

⇒ Motivates often that GAV is easier than LAV. However, „**single database property**“ only applies in this restricted setting!

- Possibility to express constraints in \mathcal{G} greatly **enhances modelling power** of GAV systems, especially when data model of \mathcal{G} is conceptual or semantically rich (ontology)
- Most of the current data integration systems (2002) followed the GAV approach



Global-as-view

Benefits and drawbacks

Benefits

- Favours ease of query processing within the data integration system: **reformulation of queries is simple**, since it means query expansion by directly applying the mapping \mathcal{M} . No complicated form of reasoning is needed!

Drawbacks

- Assumes that the set of **considered sources (and source schemas)** is stable and does not change often. In case of changes to some source schema or addition of new source schemas \mathcal{S} **all** mappings in \mathcal{M} have to be checked and perhaps reworked. This is feasible only for small number of integrated sources and small global schemas.



Main problems in data integration

- 1 How to construct the global schema
- 2 (Automatic) source wrapping
- 3 How to discover mappings between the sources and the global schema
- 4 Limitations in the mechanisms for accessing the sources
- 5 Data extraction, cleaning and reconciliation
- 6 How to process updates expressed on the global schema, and updates expressed on the sources (“read/write” vs “read-only” data integration)
- 7 **The modeling problem: How to model the mappings between the sources and the global schema**
- 8 **The querying problem: How to answer queries expressed on the global schema**
- 9 Query optimization



Beyond the classical model

Relevance for SENSE !?

- Ontology Replace Schemas by semantically-richer structures (**Ontologies**)
- Peer-to-Peer Data Integration Systems: No **global schema** anylonger, bit a network of local schema mappings
- Data Exchange: **Materialization** using views to speed up retrieval
- So far considered only read data. What about **updating data**?
This is already very hard in the case of a single information source, providing partial views on it schema (**View-Update-Problem!**) Moreover, **autonomy of sources** is a major factor here!



References



Maurizio Lenzerini.

Data Integration: A Theoretical Perspective.

In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), June 3-5, Madison, Wisconsin, USA*, pages 233–246, 2002.



Alon Y. Levy.

Logic-Based Techniques in Data Integration.

In Jack Minker, editor, *Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14–16, 1999*, College Park, Maryland, 1999. Computer Science Department, University of Maryland.

