Belief propagation during data integration in a P2P network *

Piotr Szwed

AGH University of Science and Technology
pszwed@agh.edu.pl

Abstract. We examine properties of a peer to peer network comprising several agents that store various types of local data and exchange them through established communication channels. We propose a communication model applicable to a developed platform for data integration between various security agencies and we focus on analysis of consequences of established channels, e.g. an unintended information leakage or a presence of data silos that can be an impediment for cooperation. To detect such situations efficiently, we do not concentrate on exchanged data itself, but on a belief related to known classes of data. In the analyses we use a model, in which communications and belief states are expressed as matrix operations of linear algebra. We show that applying this model we can efficiently reason about the data that can potentially be exchanged between agents not linked directly and about the ranges, which can be reached by the data during communication flows.

Keywords: peer to peer network, data integration, belief revision, linear algebra

1 Introduction

We analyze properties of a peer to peer network comprising several agents that store various types of local data and exchange it through established communication channels.

The presented considerations stem from a practical problem related to specification and design of a platform enabling data integration based on secure exchange of information between various security and law enforcement agencies in Poland. The project is conducted within the Polish Platform for Homeland Security.

An operational concept of the system is presented in Fig. 1. Several organizations ($A_1 \dots A_n$) are responsible for collecting data and keeping them in local

* This is the draft version of the paper presented at the **Artificial Intelligence and Soft Computing - 13th International Conference, ICAISC 2014**, Zakopane, Poland, June 1-5, 2014. The paper was published in **Artificial Intelligence and Soft Computing – Lecture Notes in Computer Science**, Volume 8467, 2014 and is available at: http://link.springer.com/chapter/10.1007/978-3-319-07173-2_69

repositories. The information exchange between participants is subject to various restrictions having their origins in law regulations or bilateral contracts. Typically, they specify which data object(or its part) and in which situation can be provided for a given requester. In many cases getting access to data requires following a certain workflow in which one institution issues a formal request for information and obtains either positive or negative response.

The main goal of the designed integration platform is to automate the communication process, while respecting strictly the security and confidentiality requirements, as well as the defined rules for information exchange.

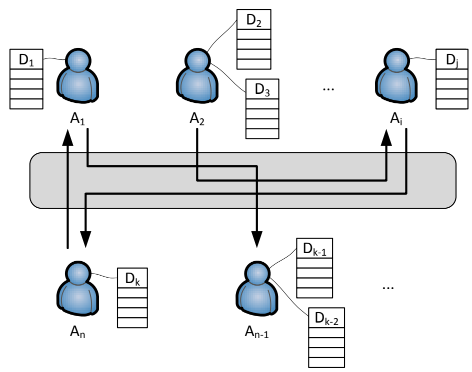

Fig. 1. An operational concept of a platform for secure exchange of information

Setting up the platform in a real environment brings up two problems. The first has rather technical implications. If an agent participates in an information exchange, appropriate interfaces, e.g. web services, should be implemented at its side. These interfaces reference data types that are either sent or received. Hence, an agent should be aware of types of objects that can reach him after flowing through the network (including readiness to accept incomplete records).

The second problem is related to the consequences of rules specifying information flows. A set of communication channels established between participants may result in unintended information leakage or, on the other side, create information silos or islands that can be an impediment for cooperation between security agencies.

To solve those problems efficiently, we do not focus on exchanged data itself, but on types of data (classes) that an agent is aware of. Moreover, it is assumed that all data types belong to a global schema (ontology) and potential problems related to definition of mappings between local ontologies and the global one can be at this stage ignored. Hence, a statement that an agent A_i *knows* a class D_j can be treated as a part of global belief state, which may be changed due to defined information flows. In the analyses we use a relatively simple, yet computationally efficient model, in which belief states and their updates are expressed as matrix operations of linear algebra. We show that this model allows for reasoning about the data, which can potentially be exchanged between agents not linked directly and about ranges the data may reach during the communication flows.

The paper is organized as follows: next Section 2 discusses various approaches to data integration with a special focus on application to crime and intelligence support, as well as on models for P2P integration approach. It is followed by Section 3, which discusses the model of communication and integration based on linear algebra. Formal tools enabling reasoning are defined in Section 4. An example of a communication system is discussed in Section 5. Section 6 provides concluding remarks.

2 Related works

Integration of data from heterogeneous data sources is an intensively researched topic stimulated by growing demand from various domains. They include business IT systems, which challenge the problem of interoperability between legacy systems after company mergers or acquisitions, bioinformatics [1], coordination of military systems [2], as well as crime and intelligence analysis [3].

The last domain encounters specific problems related to strict rules of data ownership and privacy, legal regulations pertaining to data exchange, as well as various impediments including lack of agreement between agencies responsible for collecting, storing and disseminating criminal intelligence [4]. In consequence, national or multinational security agencies often develop local repositories [5] and dedicated data integration and analysis tools, e.g. Coplink in USA [6] or recently LINK in Poland [7].

Basically, two approaches to the problem of data integration can be applied. The first assumes migration of data from heterogeneous sources to a central repository or a warehouse that can be queried referencing the terms in defined a common schema. Nevertheless, solutions based on such architecture often occurs too costly, moreover, they suffer from problems with data freshness and synchronization between local sources and the warehouse. In many situations they are also unfeasible and this is obviously the case for the considered application in the security domain.

The second approach consists in building a platform allowing to query the data in local repositories maintained by independent agents, e.g. company branches or institutions. Integration architectures within this approach fall into two categories: they are either centralized or peer to peer (P2P) [8].

A centralized architecture relies on a mediator service [9] providing a uniform interface to integrated data sources and referencing a global schema (or ontology). Within this setting, the most discussed architectural decision is related to the method of mapping between local and global schemas. It may follow either *Global as View* (GaV) or *Local as View* (LaV) approach [10]. In GaV every entity in a global schema is assigned with a set of mapping from local schemas. In LaV each local schema is treated as a view of the global one.

In a P2P architecture [11, 12] each peer represents an autonomous information system with a local schema and the data integration is usually achieved by defining separate mapping between pairs of agents. However, P2P systems may also use a global ontology approach [13].

Epistemic logic [14] is a formal language that can be used to describe state of communicating agents; it was used by Calvanese et al. [12] to define semantcis of P2P data integration systems. A multi-agent modal logic capable of representing communications among agents was proposed by Pacuit and Parikh [15]. Liau showed in [16] that belief reasoning, revision and fusion can be interpreted as operations of matrix algebra.

This paper owns the most to the work [17] by Tojo, who proposed a linear algebra model describing belief updates in a network of communicating agents. We adapted this model to enable reasoning about types of data being exchanged among agents under the assumption that their schema belongs to or can be mapped to a global ontology.

3 Model of communication system

We analyze a system comprised of a set of agents $A = \{A_1 \dots A_n\}$ linked by channels $c_1, \dots c_m$. Agents may store and exchange various data objects. The number of classes (types) of objects that can be used within the system is finite. Hence, they can be enumerated as $D_1, \dots D_k$. Let us denote $D = \{D_1, \dots D_k\}$

Each agent A_i can store objects belonging to a set of classes $D_{A_i} \subseteq D$. It may, however, expose only a part of its data. The restriction rules may concern both particular classes and particular objects. Moreover, they can be established individually for each bilateral communication within a pair of agents Nevertheless, in this work we are focused on modeling restrictions related to classes.

A communication channel is described as a tuple $c = (A_i, D_s, A_j, D_r)$, where A_i is a sender, D_s is a class of sent data, A_j is a receiver and D_r is a class of received data. Hence, a data object o of class D_s while being transmitted by a channel n can be transformed to an object o' belonging to D_r.

3.1 Classes

Let $\mathcal{A}$ is a global set of attributes (relations) and $\mathcal{V}$ a set of values. Let $v \colon \mathcal{A} \to 2^{\mathcal{V}}$ be a function that assigns to an attribute $a \in \mathcal{A}$ a set of values.

Definition 1. *A class is defined as a tuple* $D = (\mathcal{A}_c, v_c)$*, where* $\mathcal{A}_c \subset \mathcal{A}$ *and* $v_c \subset v$ *satisfies:* $\forall a \in \mathcal{A}_c \colon a \in \operatorname{dom} v_c \wedge v_c(a) \subset v(a)$.

Speaking informally, a class is defined by giving a set of its attributes $\mathcal{A}_c$ and possible attribute values.

Following Definition 1 an object o belonging to a class D can be interpreted as a valuation function $v_o \colon \mathcal{A} \to 2^{\mathcal{V}}$, satisfying: $\forall a \in \operatorname{dom} v_o \colon v_o(a) \subset v(a)$ and $\forall a \in \mathcal{A}_c \colon a \in \operatorname{dom} v_o \wedge v_o(a) \subset v_c(a)$.

Definition 2. *A class* $D_1 = (\mathcal{A}_1, v_1)$ *subsumes (is more general than)* $D_2 = (\mathcal{A}_2, v_2)$*, what is denoted by* $D_1 \sqsupseteq D_2$ *if* $\mathcal{A}_1 \subset \mathcal{A}_2$ *and* $\forall a \in \mathcal{A}_1 \colon v_1(a) \supset v_2(a)$*.*

A child class may introduce additional attributes or restrict values of attributes appearing in its supperclass. The definition allows to classify an object based on valuation of attributes.

To give some examples: $Person \sqsupseteq PersonWithAddress$, provided that $Person = (\{forename, surname, age\}, \{(forename \rightarrow string), (surname \rightarrow string), (age \rightarrow [0, \infty]\})$ and $PersonWithAddress = (\{forename, surname, address\}, \{(forename \rightarrow string), (surname \rightarrow string), (age \rightarrow [0, \infty]), (address \rightarrow string)\})$.

Another example is $Person \sqsupseteq Adolescent$,
where $Adolescent = (\{forename, surname, age\}, \{(forename \rightarrow string), (surname \rightarrow string), (age \rightarrow [12, 18]\})$.

3.2 Upcasting

If a condition $D_1 \sqsupseteq D_2$ holds, then an object o_2 of the class D_2 can be *upcast* to the class D_1.

Let us assume that o_2 is described by a valuation function v_2. The upcast object o_1 should satisfy: $v_1 = v_2 \setminus \{(a, v_2(a)) \colon a \in \mathcal{A}_2 \setminus \mathcal{A}_1\}$. Upcasting allows to view an object of a child class D_2 as belonging to its parent class D_1. The upcasting operation removes a number of attributes from the mapping v_2. It should be mentioned, that the sets of admissible attribute values, which are restricted in child classes, do not need to be changed while upcasting.

Let $D = \{D_1, \ldots, D_n\}$ be a set of classes, and $\sqsupseteq$ is a subsumption relation. The relation $\sqsupseteq$ is a transitive closure if $\forall (D_i, D_j), (D_j, D_k) \in \sqsupseteq \colon (D_i, D_k) \in \sqsupseteq$.

Technically, a closure is stored by $n \times n$ matrix of boolean values $U = [u_{ij}]$ called the *upcast matrix*. The value of an element u_{ij} is set to T if $C_i \sqsupseteq C_j$ and F otherwise. If H is a matrix showing direct taxonomic relations (direct subsumption), then $U = H^*$.

3.3 Definition of a communication system

Let $c = (A_i, D_s, A_j, D_r)$ be a communication channel between two agents. We limit our considerations to *upcasting channels*, i.e. channels satisfying $D_r \sqsupseteq D_s$. Such assumption can be justified as follows: while an object is sent through a channel it is not likely that its content will be extended, e.g. by setting additional attributes. Rather an opposite direction is to be taken. Some attributes may be hidden and removed due to legal restrictions related to information access.

To summarize the discussed concepts we give below the definition of a communication system.

Definition 3. *A communication system is defined as* $\Gamma = (A, D, \sqsupseteq, C)$*, where* A *is a set of agents,* D *a set of data types (classes),* $\sqsupseteq$ *is a subsumption relation and* $C \subset A \times D \times A \times D)$ *is a set of communication channels. It is assumed that all channels are upcasting, i.e. the following condition holds:* $\forall (A_i, D_m, A_j, D_n) \in C \colon D_n \sqsupseteq D_m$*.*

4 Reasoning

In this section we reformulate definition of the communication system in terms of linear algebra, as well as we provide formal tools enabling reasoning about its properties.

4.1 System state

The state of the system is described as an assignment of sets of classes to agents. We do not focus on the data items that are known to an agents, but rather on classes of objects which they store.

We assume that sets of classes D and agents A are ordered. System state is a $|D| \times |A|$ matrix $S = [s^i_j]$. Its element s^i_j is equal to T (true) if an agent A_j is aware of the existence of a class D_i.

4.2 Communication and belief propagation

The set of channels $C \subset A \times D \times A \times D$ is encoded as 4-dimensional matrix $E = [e^{lj}_{ki}]$ of size $|D| \times |A| \times |A| \times |D|$ containing boolean values T and F.

$$e^{lj}_{ki} = \begin{cases} T, & \text{if } (A_i, D_j, A_k, D_l) \in C \\ T, & \text{if } l = i \text{ and } k = j \\ F, & \text{otherwise} \end{cases} \tag{1}$$

It can be observed that elements at diagonals $l = i$ and $k = j$ are set to T. They play the role of identity matrix I, hence each matrix E can be decomposed into $E' + I$, where E' describes the true communications and I guarantees that agents preserve the information gained.

Belief propagation is described by a state equation (2), where $S(m)$ and $S(m+1)$ denote successive states and $\circ$ is an operator that takes on input a 4D and a 2D boolean matrix and yields a 2D matrix, whose elements are calculated according to formula (3). In each step agents propagate information on classes of stored data to their neighbors. They also keep information on classes they know.

Following the Einstein convention for tensors we omit conjunction and disjunction in the subsequent formulas defining matrix operators.

$$S(m+1) = E \circ S(m) \tag{2}$$

$$s^l_k(m+1) = \bigvee_i \bigwedge_j e^{lj}_{ki} s^i_j(m) \tag{3}$$

4.3 Reachable state

Applying the equation (2) multiple times we obtain a sequence of states.

Proposition 1. *The sequence of states* $\sigma = S(0), S(1), \ldots, S(n), \ldots,$ *where* $S(i+1) = E \circ S(i)$ *converges.*

Proof. From (1) the matrix E can be expressed as sum of $(E' + I)$, hence for any i: $S(i+1) = E' \circ S(i) + S(i)$, thus σ is nondecreasing. As each state S_i is bounded above by a matrix S_{max} having all elements equal to T (true), the sequence σ converges.

Consequences of Proposition 1 are the following: if we assume, what an agent knows, i.e. which types of data it stores, we may conclude how far this information can be propagated in the network. This allows for detecting information silos or islands of belief.

4.4 Closure of a communication graph

Let us define and operator $\otimes$ that multiplies two communication matrices E and G. The resulting matrix $F = E \otimes G$ id given by formula (4).

$$f^{kl}_{mn} = e^{kl}_{ij} g^{ji}_{mn} \tag{4}$$

Each i-th element $S(i)$ of the sequence σ can be expressed applying the operator $\otimes$ as $(E \otimes E \otimes \ldots E) \circ S(0)$, where E component appear i times. This can be denoted shortly as: $S(i) = E^i \circ S(0)$

Proposition 2. *The sequence* $\epsilon = E, E^2, \ldots E^i \ldots$ *converges.*

Proof. Observe that $E^i \otimes E$ can be expressed as $(E'' + I) \otimes (E' + I) = E'' \otimes E' + E'' + E' + I$, hence the sequence ϵ is nondecreasing, it is also bounded above, thus converges.

As a consequence of Proposition 2, the E^* matrix can be interpreted as a transitive closure of the communication graph. To give an example, if there exist two channels $c_{12} = (A_1, D_1, A_2, D_2)$ and $c_{23} = (A_2, D_2, A_3, D_3)$ represented as appropriate elements in E, the matrix E^* contains an element corresponding to the derived channel $c_{13} = (A_1, D_1, A_3, D_3)$ being a shortcut from A_1 to A_3. Such derived channels can be identified by examination of $E^* - E$.

4.5 Channel and class matching

Let us consider a situation where agents A_i and A_j are linked by a channel $n = (A_i, C_s, A_j, C_r)$ and the agent A_i is aware of a class C_{s2} satisfying $C_s \sqsupseteq C_{s2}$. The channel specification does not match directly the class C_{s2}, hence objects of this class cannot be transmitted trough the channel. However, they can be upcast to C_s (by removing extra attributes appearing in C_{s2}) and then sent.

Following this observation, we introduce additional component to the state equation (2), namely the upcast matrix U.

$$S(m+1) = E \circ (US(m)) \tag{5}$$

The formula (5) can be rewritten as (6), where $\odot$ operator is defined by (7).

$$S(m+1) = (E \odot U) \circ S(m) \tag{6}$$

$$f^{kl}_{mj} = e^{kl}_{mi} u^i_j \tag{7}$$

Let us observe that an upcast operation can be also applied on arrival of data through a channel, i.e. on left side of the state equation (2). With a set of introduced operators it can be defined as (8).

$$S(m+1) = ((I \odot U) \otimes E) \circ S(m) \tag{8}$$

From now we will omit operators in presented formulas assuming that appropriate operator can be selected based on types of operands. It should be noted that the data structures supporting E, S and U matrices and operators given by (3), (4) and (7) were implemented in a prototype software that was used to analyze the example presented in the next section.

5 Example

An example of a system comprising five agents linked by communication channels is given in Fig. 2. Agents $A_1 \dots A_5$ are marked as circles, whereas channels as rectangles. Each channel is attributed with two class names: the first is a class of objects that are sent, the second class of objects received. The hierarchy of classes referenced on the diagram is shown in Fig. 3. As it can be checked, all channels are upcasting channels, i.e. classes on output subsume classes on input.

The upcast matrix for the class hierarchy in Fig. 3 is given by (9). As we are not capable of presenting communication matrices, we will rather enumerate the channels they define.

$$U = \begin{pmatrix} T & T & F & F & F & F \\ F & T & F & F & F & F \\ F & F & T & T & T & T \\ F & F & F & T & T & T \\ F & F & F & F & T & F \\ F & F & F & F & F & T \end{pmatrix} \tag{9}$$

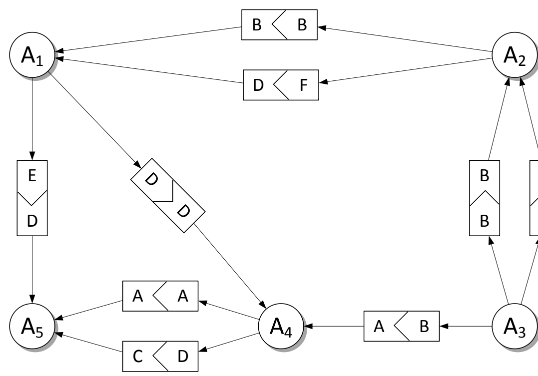

Fig. 2. Five agents $A_1 \dots A_5$ linked by communication channels

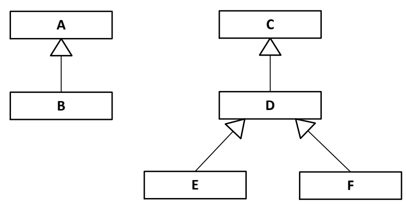

Fig. 3. Example of a class hierarchy

5.1 Closures of communication graph

In order to reason about possible communications we calculate closures of the communication graph. Table 1 gives the communication channels for E^*, E^*U, UE^* and $(EU)^*$. The initial setting defines 9 channels. The first column related to E^* gives 17 information flows, that are obtained by direct class matching (without upcasting). The second applies upcasting on the right side. The third column corresponds to UE^* (upcasting objects on arrival) and the fourth applies upcasting each time the objects are sent. For the considered example E^*U fits the best intuition, how the communication is performed. Calculation of the closure $(EU)^*$ yields channels that apply upcasting on arrival, e.g. $A_1 \to (D) \to (C) \to A_4$, whereas the initial channel specification is $A_1 \to (D) \to (D) \to A_4$. This can be explained by analysis of formula (10) showing the expanded chain of matrix multiplications for $(EU)^*$. It can be observed, that actually for each communication, apart the last one, upcasting on reception of data occurs.

$$(EU)^* = EUEU \dots E(UEU) \dots EU \tag{10}$$

Regardless of the closure applied, its analysis yields valuable information about possible information flows. Returning back to the problem origins, each implemented channel (described by E) is defined according to law regulations or bilateral contracts. In the case where E^* and E differs, e.g. the channel $A_3 \to$

Table 1. Commnunication and its closures

E^*	E^*U	UE^*	$(EU)^*$
1. $A_1 \to (D) \to (D) \to A_4$	1. $A_1 \to (D) \to (D) \to A_4$	1. $A_1 \to (D) \to (C) \to A_4$	1. $A_1 \to (D) \to (C) \to A_4$
2. $A_1 \to (D) \to (C) \to A_5$	2. $A_1 \to (D) \to (C) \to A_5$	2. $A_1 \to (D) \to (D) \to A_4$	2. $A_1 \to (D) \to (D) \to A_4$
3. $A_1 \to (E) \to (D) \to A_5$	3. $A_1 \to (E) \to (D) \to A_4$	3. $A_1 \to (D) \to (C) \to A_5$	3. $A_1 \to (D) \to (C) \to A_5$
4. $A_2 \to (B) \to (B) \to A_1$	4. $A_1 \to (E) \to (C) \to A_5$	4. $A_1 \to (E) \to (C) \to A_5$	4. $A_1 \to (E) \to (C) \to A_4$
5. $A_2 \to (F) \to (D) \to A_1$	5. $A_1 \to (E) \to (D) \to A_5$	5. $A_1 \to (E) \to (D) \to A_5$	5. $A_1 \to (E) \to (D) \to A_4$
6. $A_2 \to (F) \to (D) \to A_4$	6. $A_1 \to (F) \to (D) \to A_4$	6. $A_2 \to (B) \to (A) \to A_1$	6. $A_1 \to (E) \to (C) \to A_5$
7. $A_2 \to (F) \to (C) \to A_5$	7. $A_1 \to (F) \to (C) \to A_5$	7. $A_2 \to (B) \to (B) \to A_1$	7. $A_1 \to (E) \to (D) \to A_5$
8. $A_3 \to (B) \to (B) \to A_1$	8. $A_2 \to (B) \to (B) \to A_1$	8. $A_2 \to (F) \to (C) \to A_1$	8. $A_1 \to (F) \to (C) \to A_4$
9. $A_3 \to (B) \to (B) \to A_2$	9. $A_2 \to (F) \to (D) \to A_1$	9. $A_2 \to (F) \to (D) \to A_1$	9. $A_1 \to (F) \to (D) \to A_4$
10. $A_3 \to (B) \to (A) \to A_4$	10. $A_2 \to (F) \to (D) \to A_4$	10. $A_2 \to (F) \to (C) \to A_4$	10. $A_1 \to (F) \to (C) \to A_5$
11. $A_3 \to (B) \to (A) \to A_5$	11. $A_2 \to (F) \to (C) \to A_5$	11. $A_2 \to (F) \to (D) \to A_4$	11. $A_2 \to (B) \to (A) \to A_1$
12. $A_3 \to (F) \to (D) \to A_1$	12. $A_3 \to (B) \to (B) \to A_1$	12. $A_2 \to (F) \to (C) \to A_5$	12. $A_2 \to (B) \to (B) \to A_1$
13. $A_3 \to (F) \to (F) \to A_2$	13. $A_3 \to (B) \to (B) \to A_2$	13. $A_3 \to (B) \to (A) \to A_1$	13. $A_2 \to (F) \to (C) \to A_1$
14. $A_3 \to (F) \to (D) \to A_4$	14. $A_3 \to (B) \to (A) \to A_4$	14. $A_3 \to (B) \to (B) \to A_1$	14. $A_2 \to (F) \to (D) \to A_1$
15. $A_3 \to (F) \to (C) \to A_5$	15. $A_3 \to (B) \to (A) \to A_5$	15. $A_3 \to (B) \to (A) \to A_2$	15. $A_2 \to (F) \to (C) \to A_4$
16. $A_4 \to (A) \to (A) \to A_5$	16. $A_3 \to (F) \to (D) \to A_1$	16. $A_3 \to (B) \to (B) \to A_2$	16. $A_2 \to (F) \to (D) \to A_4$
17. $A_4 \to (D) \to (C) \to A_5$	17. $A_3 \to (F) \to (F) \to A_2$	17. $A_3 \to (B) \to (A) \to A_4$	17. $A_2 \to (F) \to (C) \to A_5$
	18. $A_3 \to (F) \to (D) \to A_4$	18. $A_3 \to (B) \to (A) \to A_5$	18. $A_3 \to (B) \to (A) \to A_1$
	19. $A_3 \to (F) \to (C) \to A_5$	19. $A_3 \to (F) \to (C) \to A_1$	19. $A_3 \to (B) \to (B) \to A_1$
	20. $A_4 \to (A) \to (A) \to A_5$	20. $A_3 \to (F) \to (D) \to A_1$	20. $A_3 \to (B) \to (A) \to A_2$
	21. $A_4 \to (B) \to (A) \to A_5$	21. $A_3 \to (F) \to (C) \to A_2$	21. $A_3 \to (B) \to (B) \to A_2$
	22. $A_4 \to (D) \to (C) \to A_5$	22. $A_3 \to (F) \to (D) \to A_2$	22. $A_3 \to (B) \to (A) \to A_4$
	23. $A_4 \to (E) \to (C) \to A_5$	23. $A_3 \to (F) \to (F) \to A_2$	23. $A_3 \to (B) \to (A) \to A_5$
	24. $A_4 \to (F) \to (C) \to A_5$	24. $A_3 \to (F) \to (C) \to A_4$	24. $A_3 \to (F) \to (C) \to A_1$
		25. $A_3 \to (F) \to (D) \to A_4$	25. $A_3 \to (F) \to (D) \to A_1$
		26. $A_3 \to (F) \to (C) \to A_5$	26. $A_3 \to (F) \to (C) \to A_2$
		27. $A_4 \to (A) \to (A) \to A_5$	27. $A_3 \to (F) \to (D) \to A_2$
		28. $A_4 \to (D) \to (C) \to A_5$	28. $A_3 \to (F) \to (F) \to A_2$
			29. $A_3 \to (F) \to (C) \to A_4$
			30. $A_3 \to (F) \to (D) \to A_4$
			31. $A_3 \to (F) \to (C) \to A_5$
			32. $A_4 \to (A) \to (A) \to A_5$
			33. $A_4 \to (B) \to (A) \to A_5$
			34. $A_4 \to (D) \to (C) \to A_5$
			35. $A_4 \to (E) \to (C) \to A_5$
			36. $A_4 \to (F) \to (C) \to A_5$

$(B) \to (B) \to A_1$ exists in E^* and not in E, a contract between A_3 and A_1 can be proposed to shorten the communication path.

An interesting problem that can be examined is a possible specification of *forbidden* communication channels. Such restriction may stem from legal regulations. For example, if a channel $A_3 \to (F) \to (C) \to A_4$ is forbidden (c.f. Fig. 2), its presence in $(EU)^*$ can be considered a possibility of an unintended information leakage violating current law regulations.

5.2 State reachability

Another property that can be examined is state reachability. For the analysis an initial state S_0 defining, which classes are known by the agents is required. Then, the state equation (2) or (5) can be applied multiple times giving information about the data types that the agents eventually should be aware of. An alternative method consists in calculating directly E^*US_0 or $(EU)^*S_0$.

Probably, the most valuable result can be obtained by analyzing a state that can be reached assuming that only one agent is aware of several classes (e.g. related to the data that are stored in its local database). Hence, the reachable state describes the range, which information originating from a given agent can reach.

Fig. 4 shows reachable states that can be computed applying $(EU)^*$. Classes are marked as rectangles and are connected by edges with agents, who know them. In the initial state S_0 agent A_3 knows classes from the set $\{E, F, B\}$. The initial assignment is marked with the continuous bold lines.

It can be observed, that A_3 does not share information of type E with anyone, i.e. behaves like a silo with respect to E, further, data objects of type F can reach only A_2 and for B the information is shared with A_1 and A_2.

The analysis of reachable states may indicate several clusters of agents, each of them assigned with a certain class D_i. Hence, no data object belonging to classes in D_i may leave the cluster, which constitutes in this way an island of belief. Presence of islands may indicate a serious obstacle for integration of activities of various security agencies.

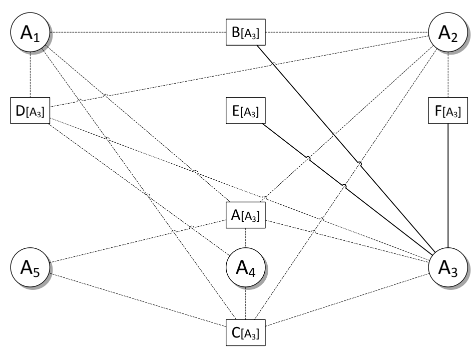

Fig. 4. Reachable state from A_3 for $(EU)^*$

6 Conclusions

The main contributions of this work are an extension of Tojo's model of belief update with the concept of upcasting operations and an idea of its application to reason about a P2P data integration platform within the security domain.

Introduction of upcasting channels was indispensable to model schema mapping and partial information hiding during data transmission. As a distinction is made between types of data that are sent and received, 4-dimensional tensors were used to model communications. A new element of the model is also the idea of applying upcasting operation before transmission (technically implemented by an upcast matrix) to match a channel specification.

Mapping the concept of upcasting back to the Tojo's belief revision model related to logic propositions, we may consider an atomic proposition p as a statement: an agent A knows (stores objects of) the class P. Hence, if a class Q subsumes P, then such knowledge can be specified by an axiom $p \Rightarrow q$, which should be globally satisfied by an underlying semantic model.

Although the formal tools used for analysis of communications between agents are relatively simple, they occurred surprisingly very efficient and computationally feasible. The size of the presented example is small, hence, it is possible to reason about its properties by hand. As a real use case for the developed platform, we may expect few dozen agents storing data belonging to about ten categories, that can be further divided into about fifty subclasses. Even if the expected structure of communication is rather sparse, we find that for systems of such size, a manual analysis of their properties would be virtually impossible. Hence, models and tools supporting automated analysis can be considered a very useful aid during the deployment and the validation.

Acknowledgments. This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. O ROB 0021 01/ID 21/2

References

1. Caragea, D., Pathak, J., Bao, J., Silvescu, A., Andorf, C., Dobbs, D., Honavar, V.: Information integration and knowledge acquisition from semantically heterogeneous biological data sources. In: Data Integration in the Life Sciences, Springer (2005) 175–190
2. Tolk, A., Muguira, J.A.: The levels of conceptual interoperability model. In: Proceedings of the 2003 Fall Simulation Interoperability Workshop. Volume 7., Citeseer (2003)
3. Chen, H., Wang, F.Y.: Guest editors' introduction: Artificial intelligence for homeland security. Intelligent Systems, IEEE **20**(5) (2005) 12–16
4. Parliamentary Joint Committee on Law Enforcement: Inquiry into the gathering and use of criminal intelligence. Commonwealth of Australia (May 2013)
5. Ruść, T., Orzechowski, T., Korus, P., Grega, M., Leszczuk, M., et. al.: Preliminary report on police and prosecutor repositories and access procedures. Deliverable D5.1, INDECT Consortium (June 2010)
6. Hauck, R., Atabakhsb, H., Ongvasith, P., Gupta, H., Chen, H.: Using Coplink to analyze criminal-justice data. Computer **35**(3) (2002) 30–37
7. Dajda, J., Debski, R., Kisiel-Dorohinicki, M., Pietak, K.: Multi-domain data integration for criminal intelligence. In Gruca, D.A., Czachórski, T., Kozielski, S., eds.: Man-Machine Interactions 3. Volume 242 of Advances in Intelligent Systems and Computing. Springer International Publishing (2014) 345–352
8. Cruz, I.F., Xiao, H.: Ontology driven data integration in heterogeneous networks. In Tolk, A., Jain, L., eds.: Complex Systems in Knowledge-based Environments: Theory, Models and Applications. Volume 168 of Studies in Computational Intelligence. Springer Berlin Heidelberg (2009) 75–98
9. Hull, R., Zhou, G.: A framework for supporting data integration using the materialized and virtual approaches. Volume 25. ACM (1996)
10. Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM (2002) 233–246
11. Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R.J., Mylopoulos, J.: The hyperion project: from data integration to data coordination. ACM SIGMOD Record **32**(3) (2003) 53–58
12. Calvanese, D., Damaggio, E., De Giacomo, G., Lenzerini, M., Rosati, R.: Semantic data integration in p2p systems. In: Databases, Information Systems, and Peer-to-Peer Computing. Springer (2004) 77–90

13. Cruz, I.F., Xiao, H., Hsu, F.: Peer-to-peer semantic integration of xml and rdf data sources. In: In Third International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2004. (2004)
14. Gochet, P., Gribomont, P.: Epistemic logic. In Gabbay, D.M., Woods, J., eds.: Logic and the Modalities in the Twentieth Century. Volume 7 of Handbook of the History of Logic. North-Holland (2006) 99 – 195
15. Pacuit, E., Parikh, R.: Reasoning about communication graphs. Interactive Logic **1** (2007) 135–157
16. Liau, C.J.: Belief reasoning, revision and fusion by matrix algebra. In Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W., eds.: Rough Sets and Current Trends in Computing. Volume 3066 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 133–142
17. Tojo, S.: Collective belief revision in linear algebra. In M. Ganzha, L. Maciaszek, M.P., ed.: Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, IEEE (2013) pages 175–178.