Paulo Alexandre Fangueiro Oliveira Maio

AN EXTENSIBLE ARGUMENTATION MODEL FOR ONTOLOGY MATCHING NEGOTIATION

Doutoramento em

Engenharia Electrotécnica e de Computadores

Orientador: Professor Nuno Alexandre Pinto da Silva

Co-orientador: Professor José Carlos Silva Cardoso

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

VILA REAL, 2012

An Extensible Argumentation Model for Ontology Matching Negotiation

a thesis submitted for the Degree of Doctor of Philosophy by

Paulo Alexandre Fangueiro Oliveira Maio

supervised by

Professor Nuno Alexandre Pinto da Silva

Professor José Carlos Silva Cardoso

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

VILA REAL, PORTUGAL

2012

ACKNOWLEDGMENTS

I would like to thank Professor Nuno Silva for his supervision throughout my PhD. He has shown great patience and pragmatic sense, which have been very important at the hardest times. It has been an enormous privilege to be his first PhD student.

I would also like to thank Professor José Cardoso for the opportunities he gave me by supervising my PhD within the scope of the Universidade de Trás-os-Montes e Alto Douro, and for the availability he showed during these years.

I am very grateful to the Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão (GECAD) for logistical and financial support, along with the opportunity to research in such a dynamic research group.

Special thanks to Jorge Santos and Maria João Viamonte for their revision of this thesis.

Thanks also to Hélio Martins for his inexhaustible and clever sense of humor.

I am also very grateful to Doctor Owen Gilson for his English revisions of my publications in international conferences and especially for the revision of this thesis.

Thanks also to Jorge Coelho and Besik Dundua for their pertinent counsels and readiness over these years.

I am also very grateful to the Instituto Superior de Engenharia do Porto and Instituto Politécnico do Porto for the financial and organizational support they provided me.

Thanks also to my colleagues of Departamento de Engenharia Informática and especially those with whom I have the pleasure to teach, for their support and efforts during these years.

I must also thank to Fundação para a Ciência e Tecnologia for the support given to my PhD through the EDGAR, COALESCE and AEMOS projects and to the Portuguese program Quadro de Referência Estratégico Nacional 2007-2013 for their financial support to my participation in international conferences.

Thanks also to the Serviços Técnicos de Informática Logistica e Comunicações (STILC), and in particular to Eng. Marques dos Santos, for the pleasant and rewarding years I worked there and for the understanding shown when I decided to follow new professional directions. I would not be the same professional without this experience.

I am especially grateful to the person who firstly and timely challenged me to undertake doctoral work, for always having belief in me and in my abilities and continuously supporting me and encouraging me throughout this difficult and long journey. I hope I never disappoint you: so far, and in the future.

Special thanks to Toni, Sofia, Eduardo and Teresa for their friendship, sincerity and intelligence. They are like family to me and give me great moments of fun and common sense discussions. I wish them all the best.

Last but definitely not the least, the very special thanks go to my parents and Alexandra for their blessings, love, affection, and interest in my work. Their moral support, patience and timely advice have been instrumental to my life and, in particular, to the completion of this thesis. All the confidence they always place in me and all the joy they always give me when I see and think of each of them makes me a better person.

ABSTRACT

With the advent of new technological and socio-organizational paradigms raised by the globalization phenomenon, information and communication systems are facing unprecedented levels of distribution, heterogeneity and evolution. In this context, many applications/scenarios see the ontology matching process as an appropriate approach to overcome such heterogeneity since it is able to define an alignment between two ontologies at the conceptual level, which in turn is further exploited to enhance interoperability between applications and/or systems.

Due to the subjective nature of ontologies, context, preferences, interests and alignment requirements, different systems have contradictory and inconsistent perspectives about the alignment. Consequently, conflicts arise between systems about the best alignment to use.

To address such conflicts, ontology matching negotiation arises as promising approach enabling systems to establish a consensual alignment between their ontologies. There are two kinds of ontology matching negotiation approaches: (i) relaxation-based approaches and (ii) argument-based approaches.

The first contribution of this thesis is the identification and description of several limitations in the state-of-the-art. These limitations drove the research efforts described in this thesis in order to overcome them.

This thesis focus on researching argumentation-based (semi-) automatic negotiation approaches and methodologies that exploit existing ontology matching tools in order to permit heterogeneous applications/systems to overcome potentially existing divergences in establishing a consensual ontology alignment, and consequently facilitating their interoperability.

Despite the focus of the thesis, several of the presented contributions go beyond the ontology matching negotiation domain, such that they are also applicable in a diversity of domains including e-commerce, legal reasoning and decision making. In this particular, two generic contributions are highlighted.

The first generic contribution is the proposal of an iterative and incremental argument-based negotiation process that promotes the adoption of an explicit, formal and extensible specification of a shared argumentation model between argument-based negotiating agents. Argumentation is adopted by the negotiating agents as a modeling formalism for theoretical and practical reasoning and, therefore, governing the internal and external agents' behavior.

The second contribution concerns the specification of a general argumentation framework that (i) comprehends a conceptualization layer to capture the semantics of the argumentation data employed in a specific context and (ii) provides modularity and extensibility features that simplify its adoption by argumentation systems. Yet, it also adopts a general and intuitive argument structure which is largely accepted by the argumentation community.

Finally, a third contribution concerns the application of the previous generic contributions to the ontology matching negotiation domain. The novel argument-based negotiation approach exploits and profits from these generic contributions to overcome (most of) the limitations previously identified in the state-of-the-art on argument-based ontology matching negotiation.

Accordingly, the dichotomy and the symbiosis between the generic contributions and ontology matching negotiation contributions are intrinsic and an essential part of this thesis.

RESUMO ALARGADO

O processo de Mapeamento de Ontologias consiste em definir ao nível conceptual um conjunto de correspondências semânticas entre entidades de duas ontologias distintas: cada uma descrevendo a informação capturada num determinado sistema de informação e/ou comunicação. O resultado deste processo denomina-se alinhamento, e é posteriormente explorado de diferentes formas para superar a heterogeneidade existente entre os sistemas e permitir a sua interoperabilidade, nomeadamente em: transformação de dados [Sheth and Larson 1990; Bernstein and Rahm 2000], integração de informação [Halevy et al. 2005; Wache et al. 2001; Draper, Halevy, and Weld 2001], visualização de informação [Gilson et al. 2008], comunicação entre agentes em sistemas multiagente [Eijk et al. 2001; Wiesman, Roos, and Vogt 2001; Bailin and Truszkowski 2003], partilha de informação em sistema ponto-a-ponto[1] [Zaihrayeu 2006], resposta a inquéritos[2] [Lopez, Motta, and Uren 2006; Mena et al. 1996] ou navegação na web semântiac [Sabou, Lopez, and Motta 2006]. Contudo, devido à natureza subjectiva das ontologias, ao contexto, às preferências, interesses e requisitos de mapeamento, diferentes sistemas têm perspetivas contraditórias e inconsistentes sobre as correspondências, o que conduz ao surgimento de conflitos entre os sistemas sobre o alinhamento a adotar.

O trabalho realizado tem por objectivo pesquisar abordagens e metodologias (semi-) automáticas de negociação baseadas em argumentos que explorem as ferramentas de mapeamento de ontologias já existentes e conduzam sistemas heterogêneos a superar conflitos na obtenção de um alinhamento consensual entre as suas ontologias.

[1] Do inglês "peer-to-peer".
[2] Do inglês "query answering".

Com vista à resolução de conflitos e à obtenção de melhores acordos entre os sistemas negociadores, esta tese defende a necessidade e os benefícios da adoção de uma especificação explícita, formal, extensível e partilhada de um modelo de argumentação.

Genericamente, o trabalhado descrito nesta tese está dividido em sete partes, que correspondem ao longo do documento a outros tantos capítulos:

1. Conhecimentos Requeridos;
2. Estado da Arte;
3. Processo de Negociação baseado em Argumentos;
4. TLAF: Estrutura de Argumentação em Três Camadas;
5. Negociação do Alinhamento;
6. Extensibilidade e Modularidade na TLAF;
7. Experiências.

As secções seguintes descrevem sumariamente o conteúdo de cada uma destas partes, com especial enfoco no trabalho desenvolvido e nos resultados mais relevantes. Posteriormente, na última secção apresenta-se uma síntese do trabalho desenvolvido e alguns indicadores da relevância da investigação realizada.

1. Conhecimentos Requeridos

O trabalho descrito ao longo da tese explora e relaciona principalmente três áreas de conhecimento: (i) agentes e sistemas baseados em agentes, (ii) mapeamento de ontologias e (iii) argumentação e sistemas baseados em argumentação. Os conceitos e princípios fundamentais de cada uma destas áreas são sumariamente apresentados.

Um agente é uma entidade que observa e atua num ambiente em representação de pessoas e organizações com o propósito de concretizar os seus objetivos [Wooldridge and Jennings 1995; Wooldridge 2000]. Um sistema multiagente consiste num conjunto de agentes autónomos que interagem com outros agentes de forma semelhante ao que as pessoas e as organizações fazem no dia-a-dia. Neste contexto, a adoção de técnicas de argumentação podem ser exploradas pelo agente:

- Internamente: para determinar (i) em que acreditar (raciocíonio teórico) e (ii) o que fazer (raciocínio prático);
- Externamente: para interação com os outros agentes. Nomeadamente em diálogos envolvendo negociação, deliberação e persuasão [Walton and Krabbe 1995].

O processo de mapeamento de ontologias pode ser visto como uma função que dado um par de ontologias a mapear, um conjunto de parâmetros e um conjunto de recursos, retorna um conjunto de correspondências entre as entidades das ontologias a mapear [Euzenat and Shvaiko 2007]. Este deve ter em consideração os diversos tipos de heterogeneidade que podem existir simultâneamente: (i) sintática, (ii) terminológica, (iii) conceptual e (iv) pragmática. Os algoritmos de mapeamento são normalmente classificados como: (i) simples se apenas exploram um critério de mapeamento ou (ii) complexos se exploram mais do que um critério de mapeamento [Rahm and Bernstein 2001]. Os critérios de mapeamento têm em consideração diversas dimensões como a granularidade (e.g. ao nível do elemento, ao nível da estrutura) e a interpretação da informação de entrada (e.g. interpretação sintática, lexical) [Shvaiko and Euzenat 2005]. Tipicamente, um sistema de mapeamento de ontologias explora vários algoritmos simples com competências diferentes e complementares [Euzenat and Shvaiko 2007]. Estes recorrem a vários métodos de filtragem [Ehrig and Sure 2004; Do and Rahm 2002; Melnik, Rahm, and Bernstein 2003] e de agregação [Ji, Haase, and Qi 2008] dos alinhamentos resultantes de forma a fornecerem como saída um único alinhamento. A seleção de um sistema de mapeamento deve ter em consideração tanto as ontologias a mapear como o problema a ser resolvido pelo alinhamento resultante [Euzenat and Shvaiko 2007].

As estruturas[3] de argumentação abstratas são formalismos adequados para representar vários cenários diferentes, sem comprometimento com qualquer domínio de aplicação. Nestas estruturas (e.g. AF [Dung 1995], BAF [Cayrol and Lagasquie-Schiex 2005a], VAF [Bench-Capon 2003]) um argumento é algo que pode atacar/suportar ou ser atacado/suportado por um outro argumento. A ausência de uma estrutura e semântica interna do argumento torna estes formalismos adequados ao estudo de propriedades independentes relevantes para qualquer contexto de argumentação, nomeadamente a aceitabilidade de argumentos [Baroni and Giacomin 2009]. Contudo, esta abstração também representa uma limitação de expressividade para a sua adoção direta num contexto de aplicação específico. Assim, normalmente os sistemas de argumentação adotam um destes formalismos abstratos e estendem-no de modo a obter um formalismo menos abstrato que especifique:

- A construção de argumentos e a sua estrutura interna;
- As condições em que um argumento ataca e/ou suporta outro argumento;
- As condições em que um argumento derrota outro.

Apesar desta situação ser recorrente, as estruturas de argumentação abstratas não contemplam qualquer funcionalidade que facilite e reja o processo de extensão da mesma.

[3] Do inglês “framework”.

Relativamente à aceitabilidade de argumentos, uma *extensão preferida*[4] é um subconjunto do conjunto de argumentos conhecidos que representa uma posição consistente e defensável contra qualquer ataque, e não pode ser estendido (aumentado) sem introduzir um conflito. Para o mesmo conjunto de argumentos é possível existirem múltiplas extensões preferidas.

2. Estado da Arte

Na literatura relativa à obtenção de um alinhamento consensual entre agentes existem dois tipos de abordagens diferentes.

A primeira abordagem adota uma negociação baseada em mecanismos de relaxação [Silva, Maio, and Rocha 2005; Maio et al. 2006]. Nesta abordagem, cada agente classifica as diversas correspondências em obrigatórias, propostas, negociáveis, rejeitadas e eliminadas. Posteriormente, as exigências sobre as correspondências são relaxadas interna e privadamente por cada agente na expectativa do seu oponente proceder da mesma forma. A obtenção de um acordo sobre o alinhamento a usar resulta da avaliação dos ganhos e perdas obtidos por cada agente. A principal limitação desta abordagem advém da dificuldade de especificação das funções associadas aos mecanismos de relaxação e de medição dos ganhos e perdas de cada agente.

A segunda abordagem adota uma negociação baseada na troca de argumentos entre os agentes [Laera et al. 2007; Doran et al. 2010]. Esta assenta numa estrutura de argumentação abstrata que captura (i) os argumentos existentes, (ii) as relações de ataque entre argumentos e (iii) o valor promovido por cada argumento. Normalmente, estes valores correspondem a tipos de argumentos. Neste caso particular, os valores possíveis advém da classificação dos algoritmos de mapeamento em cinco categorias distintas: Terminológico, Estrutural Interno, Estrutural Externo, Semântico e Extensional. Cada agente estabelece uma ordem de preferência sobre estes valores que é posteriormente explorada para avaliar o sucesso/insucesso dos ataques entre argumentos. Um ataque é bem sucedido quando o valor promovido pelo argumento que ataca é tão ou mais preferido que o valor promovido pelo argumento atacado. Assim, um agente altera a sua posição relativamente à inclusão ou exclusão de uma dada correspondência do alinhamento a acordar se o seu oponente apresentar argumentos que sob o seu ponto de vista são mais valiosos. Apesar da simplicidade e da eficácia desta abordagem, esta sofre de diversas limitações, nomeadamente:

- Os agentes usam um único e comum repositório de correspondências para obtenção de correspondências a partir das quais geram argumentos. Este facto é visto como uma limitação à autonomia dos agentes;

[4] Do inglês "preferred extension".

- Não está contemplada a possibilidade de cada agente poder usar internamente outros tipos de argumentos para além dos que estão prévia e comummente estabelecidos entre os agentes;
- Os agentes não têm a possibilidade de capturar possíveis dependências entre correspondências e consequentemente ter em consideração o efeito positivo ou negativo da aceitação/rejeição de uma correspondência noutra correspondência;
- Os agentes não podem expressar preferências sobre os argumentos através da combinação dos diversos tipos de argumentos. Por exemplo, um argumento do tipo terminológico é preferido relativamente a argumentos do tipo estrutural externo e estrutural interno, exceto se estes dois tipos de argumentos existem e estão de acordo entre si;
- Apenas o conceito de refutação[5] de argumentos é formalmente explorado na deteção de ataques entre argumentos;
- Devido ao processo de geração de argumentos adotado, formalmente um argumento apenas ataca e é atacado por um outro argumento que conclua a negação da sua conclusão. Consequentemente os ataques entre argumentos são sempre simétricos;

Dadas as limitações apontadas a esta abordagem, decidiu-se seguir uma linha de investigação diferente.

3. Processo de Negociação baseado em Argumentos

O processo de negociação baseado em argumentos proposto é suficientemente genérico para ser adotado em vários domínios de aplicação como, por exemplo, o mapeamento de ontologias, o comércio electrónico e sistemas de apoio à decisão.

Este processo segue uma abordagem iterativa e incremental de troca de argumentos e contra-argumentos semelhante à observada num cenário de argumentação entre humanos. Nesse sentido, o processo define e explora a noção de modelo de argumentação. Neste contexto, um modelo de argumentação é um artefacto que captura (parcial ou totalmente) a percepção e racionalidade que um agente tem relativamente ao processo de argumentação sobre um domínio específico (e.g. mapeamento de ontologias). Assim, assumindo que a negociação ocorre no âmbito de uma comunidade de agentes, esta é capaz de definir um modelo de argumentação público sobre o domínio de aplicação em causa que é partilhado, aceite e compreendido por todos os seus membros. Internamente, cada agente tem a possibilidade de evoluir o modelo de argumentação público de modo a que este reflita melhor as suas

[5] Do inglês "*rebuttal*".

necessidades e conhecimento. O resultado dessa evolução denomina-se modelo de argumentação privado. Com base nestes pressupostos e sob a perspectiva de um agente sistematizou-se e organizou-se as diversas fases/tarefas do processo de negociação, identificou-se os principais blocos de dados e o seu fluxo, bem como os diversos atores e o seu papel no processo de negociação.

As fases do processo de negociação proposto são:

- Configuração[6]: é nesta fase que os agentes estabelecem o contexto da negociação através da identificação dos agentes participantes na negociação, definição de parâmetros e restrições à negociação (e.g. tempo máximo de duração, qual o modelo de argumentação público a adotar). Contrariamente às outras fases, esta apenas ocorre uma vez;
- Aquisição de Dados: é nesta fase que cada agente identifica e seleciona as fontes de informação (e.g. outros agentes) que lhe providenciará a informação necessária à geração dos argumentos compreendidos no seu modelo de argumentação;
- Instanciação do Modelo de Argumentação: é nesta fase que o agente transforma a informação recolhida na fase anterior em argumentos;
- Avaliação dos Argumentos: é nesta fase que o agente avalia os argumentos existentes e as relações de ataque e suporte existentes entre eles. Posteriomente, seleciona os argumentos que constituem a sua extensão preferida;
- Tentativa de Acordo: é nesta fase que cada agente apresenta a sua proposta de acordo. Da análise das diversas propostas resulta: (i) um acordo candidato e (ii) a identificação dos conflitos existentes entre os vários participantes. Com base nisto os agentes decidem entre:
    - Continuar o processo de negociação e consequentemente prosseguirem para a fase de persuasão;
    - Terminar a negociação:
        - Com sucesso, se os agentes aceitam o acordo candidato. Neste caso, os agentes prosseguem para a fase de Aceitação de Acordo;
        - Sem sucesso, se os agentes não aceitam o acordo candidato e não pretendem continuar a negociar. Neste caso, a negociação termina sem um acordo;

[6] Do inglês "Setup".

- Persuasão: é nesta fase que relativamente aos conflitos identificados na fase anterior os agentes trocam argumentos entre si no sentido de persuadirem os seus oponentes a reverem/alterarem a sua posição;
- Refinamento do Modelo de Argumentação: esta fase tem por objectivo permitir aos agentes evoluírem o modelo de argumentação público que estão a usar de modo a que este reflita melhor a conceptualização da comunidade quanto aos argumentos trocados. Esta fase é opcional;
- Atualização da Base Conhecimento: nesta fase cada agente analisa, processa e possivelmente reclassifica os argumentos recebidos durante a fase de persuasão. Como resultado, existirão argumentos que serão adicionados à sua base de conhecimentos e outros que serão ignorados por já serem conhecidos. Os argumentos adicionados são tidos em consideração na próxima ronda de propostas. O processo de negociação prossegue novamente para a fase de Aquisição de Dados;
- Aceitação de Acordo: é nesta fase que os agentes transformam o acordo candidato num acordo definitivo. Esta fase é vista como iniciadora de um conjunto de transações que visam satisfazer os termos acordados.

4. TLAF: Estrutura de Argumentação em Três Camadas

As estruturas de argumentação abstratas não contemplam uma camada de modelação que satisfaça a noção de modelo de argumentação introduzida pelo processo de negociação proposto. Para suprir esta lacuna e com o intuito de reduzir o fosso existente entre as estruturas de argumentação abstratas e os sistemas de argumentação foi proposta uma nova estrutura de argumentação genérica, ou seja, independente do domínio de aplicação mas menos abstrata que as já existentes, denominada TLAF (Three-Layer Argumentation Framework). As novidades introduzidas por esta estrutura são:

- Adoção de uma estrutura de argumento geral e intuitiva em que um argumento é composto por três elementos distintos:
    - Um conjunto de premissas;
    - Uma conclusão; e
    - Um processo de inferência que partindo das premisas permite obter a conclusão especificada;

As premissas e as conclusões dos argumentos são capturadas sob a forma de declarações/afirmações[7]. Esta estrutura foi anteriormente proposta em [Walton 2006]).

- Contempla uma camada conceptual para capturar o conhecimento (estrutura e semântica) usado num determinado domínio de aplicação de forma que este possa ser partilhado e reutilizado. Permite a especificação formal e explícita de uma conceptualização de um domínio de argumentação, satisfazendo, assim, a noção de modelo de argumentação introduzida anteriormente. O conteúdo desta camada depende diretamente do domínio de aplicação a ser capturado e da percepção que alguém (e.g. um agente ou uma comunidade de agentes) tem sobre esse domínio;
- Explora a informação conceptual capturada e a estrutura dos argumentos adotada para automaticamente derivar as relações de suporte e de ataque existente entre argumentos.

É responsabilidade dos sistemas de argumentação que adotarem a TLAF definirem uma estrutura complementar adequada para as declarações e para os mecanismos de inferência usados no domínio de aplicação.

Apesar do processo de geração de argumentos ser totalmente dependente do domínio de aplicação, este beneficia do conhecimento capturado na camada conceptual para, por exemplo, restringir as conclusões e as premisas associadas a um argumento.

No que diz respeito à avaliação dos argumentos com vista à extração de uma extensão preferida, os sistemas que adotarem a TLAF poderão continuar a utilizar os métodos já existentes associados às estruturas de argumentação abstratas [Baroni and Giacomin 2009]. Contudo, os processos existentes [Cayrol and Lagasquie-Schiex 2005b; Amgoud et al. 2008; Karacapilidis and Papadias 2001; Verheij 2002] que exploram a bipolaridade advinda das relações de ataque e suporte entre argumentos não são capazes de lidar com relações cíclicas entre argumentos nem de tirar partido da informação capturada na camada conceptual da TLAF. Para superar estas limitações foi proposto um método de avaliação dos argumentos baseado em funções.

5. Negociação do Alinhamento

A abordagem seguida para a resolução de conflitos entre sistemas sobre o alinhamento a adoptar para as suas ontologias explora tanto o processo de negociação baseado em argumentos como a TLAF propostos anteriormente. Considerando que cada sistema/agente representa uma ontologia, assumiu-se que a negociação é bilateral, i.e. ocorre apenas entre dois agentes e que cada agente usa um modelo de argumentação privado que estende o modelo de argumentação público da comunidade de agentes.

[7] Do inglês “statement”.

A abordagem proposta pode, então, ser sumariamente descrita da seguinte forma:

- Na fase de Configuração cada agente informa o seu oponente da ontologia que está a usar e são pré-definidos alguns requisitos sobre o alinhamento a acordar;
- O resultado da fase de Aquisição de Dados é um conjunto de correspondências obtidas de vários algoritmos de mapeamento internos ou externos ao agente. É da responsabilidade do agente selecionar esses algoritmos;
- Na fase de Instanciação do Modelo de Argumentação é proposto um processo automático de geração de argumentos baseado num conjunto de parâmetros de entrada e em duas funções:
    - Função de interpretação de correspondências: permite transformar as correspondências obtidas na fase anterior em declarações (conclusões) e gerar por cada declaração um argumento concluindo essa mesma declaração;
    - Função de Condições: permite atribuir a cada argumento as declarações usadas como premissas e estabelecer conflitos (inconsistências) entre declarações;
- Na fase de Avaliação dos Argumentos adota-se o processo de avaliação de argumentos introduzido pela TLAF e analisa-se as dimensões a ter em consideração na especificação das funções de avaliação, nomeadamente a dimensão quantitativa, qualitativa e força dos argumentos. São ainda apresentados alguns princípios com vista à sua combinação;
- Na fase de Tentativa de Acordo, adota-se o princípio da unanimidade entre os agentes negociadores para a computação de um acordo candidato e identificação dos conflitos existentes. O processo de decisão sobre a aceitação de um acordo candidato pelos agentes não é especificado;
- As fases de Persuasão e de Atualização da Base Conhecimento encontram-se simplificadas visto que os agentes partilham o mesmo formalismo de representação de argumentos (TLAF).

As restantes fases (Refinamento do Modelo de Argumentação e Aceitação do Acordo) não são abordadas.

As contribuições apresentadas permitem suprir na totalidade as limitações anteriormente identificadas nas abordagens de negociação de mapeamento de ontologias baseada em argumentos.

6. Extensibilidade e Modularidade na TLAF

A abordagem descrita anteriormente exige que os agentes tenham capacidade para estender o modelo de argumentação público de forma a facilitar a exploração de conhecimento que não é compartilhado e consequentemente usufruir desse conhecimento. Como a TLAF carece de funcionalidades de modularidade e extensibilidade, propõe-se a inclusão dos construtores necessários a fornecer essas funcionalidades juntamente com a semântica respectiva. A estrutura de argumentação resultante dessa extensão denomina-se EAF (Extensible Argumentation Framewrok) e mantém todas as caracteristicas da TLAF.

Dado que a EAF é independente do domínio, a sua adoção no processo de negociação de mapeamento de ontologias implica alterações mínimas no processo de argumentação interno do agente e no processo de troca de argumentação. Estas alterações são descritas através de um exemplo demonstrativo de como as funcionalidades introduzidas podem ser exploradas, nomeadamente na fase de Persuasão.

7. Experiências

Com o intuito de avaliar a eficácia do processo de negociação de mapeamento de ontologias proposto adotou-se uma abordagem empírica. As experiências realizadas visam três objectivos fundamentais:

- Comparar os resultados do processo proposto com as abordagens baseadas em argumentos já existentes;
- Avaliar o efeito produzido pela capacidade dos agentes em capturarem dependências entre correspondências;
- Avaliar a relevância dos construtores introduzidos pela EAF no processo de negociação e mais concretamente nos resultados obtidos.

As experiências realizadas são analisadas atendendo a dois critérios distintos:

- A quantidade de conflitos resolvidos e a qualidade dessa resolução;
- A qualidade global do alinhamento acordado entre os agentes por comparação com os alinhamentos gerados pelos agentes antes do processo de negociação.

Em função dos resultados obtidos é possível concluir o seguinte:

- As abordagens baseadas em argumentos existentes na literatura podem ser fielmente mimetizadas no processo de argumentação proposto. Para tal, é apenas necessário restringir adequadamente o processo de geração de argumentos proposto e utilizar funções de avaliação de argumentos equivalentes;

- Alterando apenas as funções de avaliação dos argumentos, o processo proposto supera as abordagens existentes, tanto em quantidade e qualidade de conflitos resolvidos como na qualidade do alinhamento acordado;
- A capacidade para capturar dependências entre correspondências permite aumentar a qualidade dos conflitos resolvidos e do alinhamento acordado;
- A possibilidade de os agentes estenderem privadamente o modelo de argumentação público permite-lhes melhorar as suas capacidades de mapeamento e consequentemente reduzir a quantidade de conflitos a resolver pelo processo de negociação;
- Os construtores introduzidos pela EAF quando utilizados na extensão do modelo de argumentação público em combinação com a capacidade de reclassificação de argumentos potencia a capacidade de persuasão dos agentes e conduz a um aumento tanto da qualidade do alinhamento acordado como dos conflitos resolvidos.

8. Resultados Atingidos

Apesar da ênfase inicial do trabalho de investigação descrito ser a resolução através de argumentação de conflitos resultantes de diferentes perspetivas sobre o processo de mapeamento de ontologias, à medida que o trabalho de investigação progrediu as contribuições resultantes tornaram-se mais genéricas e, consequentemente, aplicáveis a outros domínios de aplicação onde a resolução de conflitos também é necessária. Assim, a ênfase inicial pode ser vista como um cenário de aplicação onde as contribuições genéricas foram aplicadas para superar as limitações identificadas na literatura, demonstrando assim a sua pertinência e aplicabilidade.

Consequentemente, conclusões sobre a aplicabilidade e validade das contribuições propostas devem ter em consideração a dicotomia e simbiose entre as contribuições genéricas e especificas de um domínio de aplicação. Sumariamente as contribuições apresentadas são:

- Contribuições genéricas:
    - Processo iterativo e incremental de negociação baseado em argumentos. Este processo introduz a noção de modelo de argumentação e permite aos agentes usarem argumentação para raciocínio teórico e prático, tanto internamente como externamente;
    - Estrutura de argumentação genérica (TLAF/EAF) que captura e satisfaz a noção de modelo de argumentação introduzida pelo processo de negociação

proposto. Contempla funcionalidades para capturar uma conceptualização de um domínio de aplicação de forma explícita, formal, extensível e partilhada. A TLAF/EAF reduze o fosso existente entre as estruturas de argumentação abstratas e os sistemas de argumentação;

- Contribuições específicas do domínio de mapeamento de ontologias:
    - Identificação de várias limitações nas abordagens existentes de negociação baseadas em argumentos;
    - Proposta de um novo processo de negociação também baseado em argumentos que suprime a maioria das limitações identificadas nas abordagens existentes. Este processo explora as duas contribuições genéricas anteriormente mencionadas. Para isso, particularidades do domínio foram tomadas em consideração;
    - Proposta de um processo automático de geração de argumentos baseado em correspondências fornecidas por algoritmos/agentes de mapeamento de ontologias;
    - A capacidade intrínseca da abordagem proposta para acomodar mais ou menos tipos de argumentos, consoante os requisitos e conhecimentos dos agentes;

Embora conclusões formais não possam ser retiradas, a comparação das limitações do estado-da-arte com as contribuições propostas bem como as experiências efetuadas fornecem evidências sobre a validade da tese.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF DEFINITIONS

LIST OF EXAMPLES

LIST OF ABBREVIATIONS

AF	Argumentation Framework
AIF	Argument Interchange Format
AM	Argumentation Model
ANP	Argument-based Negotiation Process
ANP-OM	Argument-based Negotiation Approach for Ontology Matching
API	Application Programming Interface
B2B	Business-to-Business
B2C	Business-to-Consumer
BAF	Bipolar Argumentation Framework
BDI	Belief–Desire–Intention
cf.	Latin expression: *confer*. It means: “compare”.
CMS	CROSI (Ontology) Mapping System
CMYK	Cyan Magenta Yellow black
COMA	a system for flexible COmbination of schema Matching Approaches
DL	Description Logics
EAF	Extensible Argumentation Framework
E-Business	Electronic Business
E-Commerce	Electronic Commerce
EDOAL	Expressive and Declarative Ontology Alignment Language
e.g.	Latin expression: *exempli gratia*. It means: “for example”.
et al.	Latin expression: *et alii*. It means: “and others” or “and co-workers”.
FDO	Flexible approach for Determining agents' Orientation on ontology mappings
FIPA	Foundation for Intelligent Physical Agents
FIPA-ACL	Foundation for Intelligent Physical Agents-Agent Communication Language
F-logic	Frame logic

FOAM	Framework for Ontology Alignment and Mapping
GECAD	Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão
GMO	Graph Matching for Ontologies
GOALS	GECAD Ontology Alignment System
i.e.	Latin expression: *id est*. It means: "that is" or "in other words".
iif	if and only if
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
MAFRA	MApping FRAmework
MAS	Multi-Agent System
MbA	Meaning-based Argumentation
NOM	Naive Ontology Mapping
OAEI	Ontology Alignment Evaluation Initiative
OLA	OWL Lite Alignment
OMR	Ontology Matching Repository
OWA	Ordered Weighted Average
OWL	Ontology Web Language
P2P	Peer-to-Peer
QOM	Quick Ontology Mapping
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RGB	Red Green Blue
SMOA	String Metric for Ontology Alignment
SQL	Structured Query Language
SWRL	Semantic Web Rule Language
TLAF	Three-Layer Argumentation Framework
URI	Uniform Resource Identifier
VAF	Value-based Argumentation Framework
V-Doc	Virtual Documents for Ontology Matching
vs.	Latin expression: *versus*. It means: "against".
XML	eXtensible Markup Language

FIRST PART

Chapter 1

INTRODUCTION

The aims of this chapter are four-fold:

- First, to describe the technological and socio-organizational context of the work described in this document;
- Second, to present the goal and thesis statement driving the research efforts;
- Third, to outline the research contributions;
- Fourth, to outline the content of the document.

1.1 Context

The globalization phenomenon affects many dimensions of our daily lives, namely economy, policies and social behaviors. While new and more complex technological challenges arise with the globalization, at the same time it profits from the technological evolution to foster its own development and vice-versa. One of the most important technological challenges occurs at the information and communication systems level.

Few years ago, organizations constituted large repositories of data poorly structured and in disparate forms of representation [Fensel et al. 2002]. The difficulty to access the intended information increases proportionally to the volume of data and its representation forms. The excess of supplied information and the difficulty using that information challenged the technology to transform information into knowledge [Hey 2004; Ackoff 1989; Bellinger, Castro, and Mills 2004], which is fostered by the combination of even more and disparate repositories [Cody et al. 2002; Mack, Ravin, and Byrd 2001; Marwick 2001].

According to [Silva 1998], systems are recommended to combine flexibility and adaptability with agility, information with knowledge, autonomy with cooperation, and reaction with partnership. Knowledge-based interoperability assumes therefore a central role in the evolution of the systems as it promotes internal agility and supports faster, better and cheaper implementation of partnerships [Neches et al. 1991]. Accordingly, information-based organizations should convert themselves into knowledge-based organizations evolving their information models to encompass semantics together with autonomous and cooperative mechanisms permitting to achieve semantic interoperability.

In that sense, the ability to (internally and externally) share the semantics and knowledge must be embedded into the organization's core principles and supporting systems [Snowden 1999; Studer, Benjamins, and Fensel 1998; Fensel et al. 2002; Staab et al. 2002]. Semantics related with data and information is more and more (publically) available by the means of ontologies, which can be seen as "an explicit, formal specification of a shared conceptualization" [Studer, Benjamins, and Fensel 1998]. Ontologies are made publicly accessible and sharable, allowing information communities to characterize their data/information according to the ontologies that best fit the intended semantics of the document content. Further, ontologies allow (semantic) reasoning about the facts and its structure, giving rise to new information.

Yet, while ontologies facilitate (semantic) reasoning, different agents make use of different ontologies and possibly different semantics. In order to support such ontology mediation [Euzenat, Scharffe, and Zimmermann 2007] (ontology-based information exchange), it is necessary to establish equivalence relations between the ontologies (or between parts) of the ontologies. The process of establishing such correspondences is commonly referred to as Ontology Matching [Euzenat and Shvaiko 2007]. The result of the ontology matching process (referred to as an alignment) is further exploited to enable knowledge-based interoperability between the aligned systems, thus facilitating and improving the match between the request and the available information. This process is recognized as a key technology for many applications/systems, namely:

- Information Integration. This is probably the most traditional and oldest scenario on which the need to reconcile models is required. As a general definition, information integration is the process whereby data is stored in multiple heterogeneous information sources (e.g. databases) and further provided to consumers through a uniform global information source (adapted from [Sheth and Larson 1990; Beneventano et al. 2001;

Halevy 2001]). Therefore, information integration gathers several distinct problems such as schema integration (also referred to as data integration) [Batini, Lenzerini, and Navathe 1986; Sheth and Larson 1990; Parent and Spaccapietra 1998], database integration [Chawathe et al. 1994; Wache et al. 2001; Draper, Halevy, and Weld 2001; Halevy et al. 2005], data warehousing [Bernstein and Rahm 2000] and catalogue integration [Agrawal and Srikant 2001; Ichise, Takeda, and Honiden 2003; Giunchiglia, Shvaiko, and Yatskevich 2005]. Despite the intrinsic characteristics of each one of these individual problems leading to a different integration process, they share the existence of multiple and heterogeneous information sources (local ontologies) that need reconciliation.

- Agent-based Systems. By definition and according to [Silva and Ramos 1999] “an agent is considered an entity capable of interacting with others and its environment, sensing and changing it, and according to its own and acquired knowledge, not only react to contextual stimulus but also build and execute action plans to reach its goals”. Therefore, the information and the knowledge are typically distributed among the agents but because agents are autonomous, heterogeneous and rational entities, different conceptualizations and representations of information and knowledge exist. Unless agents share the same content ontology, the agents are not able to understand one another, thus requiring matching their ontologies in order to translate their messages.
- Semantic Web. As the Web grows it becomes increasingly difficult to be used by humans. To ease that process, Berners-Lee and colleagues [Berners-Lee and Fischetti 2000; Berners-Lee, Hendler, and Lassila 2001] suggested that the Web must shift to a computer-aided processing paradigm where machines are able to semantically comprehend the documents’ content and process their information in an automatic way with a minimal (or no) human participation. For that, web information should be annotated with machine-processable descriptions of the information to enable software entities to mediate between humans and the information sources [Fensel 2001]. Ontologies were also suggested to model, represent and convey the machine-processable description of the information between information communities, namely through ontologies that best fit the intended semantics of the document content. In that sense, heterogeneity is seen as a feature whose consequences must be overcome by the dissemination of both ontologies and reconciliation relations.
- Peer-to-Peer Systems (P2P). These systems become popular by adopting a file (e.g. music, videos, photos) sharing paradigm on which the file content is described by a simple schema composed by a set of attributes (e.g. author, title, year). Yet, those

systems assume that parties (the peers) share the same schema and have equivalent functional capabilities in providing each other with data and services [Zaihrayeu 2006]. Totally autonomous peers may use different vocabulary and conceptualizations (ontologies) to represent their data. This is the case of semantic P2P systems that in pursuing the improvement of the search accuracy adopt complex specifications for their content (e.g. database schemas [Bernstein et al. 2002] and formal ontologies [Rousset et al. 2006]) than classical P2P systems. As peers adopt their own ontologies, queries and answers are made/described according to the peers' internal ontology. Thus, a meaningful information exchange between semantic peers requires the peers' ability to match their ontologies at run-time. Once an alignment is obtained, it is exploited by a mediator to translate the original query (or answer) to a new query (or answer) according to the target peer ontology.

- Business over Internet. Under the heading of business over Internet several terms (or expressions) are gathered, namely B2B, B2C, E-Commerce and E-Business [Fensel 2001]. Ultimately, these terms embody the paradigm of electronic business interchanges through the Internet in an automatically emergent fashion [Silva 2004]. A typical implementation of B2B and B2C relies on the adoption of agent-based systems, where each agent acts and behaves in representation of an individual business partner [Lomuscio, Wooldridge, and Jennings 2003]. Despite some proposals suggesting the inclusion of third party agents, this is seen as a facilitator mechanism to accomplish the business process, responsible for the mediation between the parties in order to achieve high accuracy and trustable agreement between the parties [Silva, Viamonte, and Maio 2009; Viamonte, Silva, and Maio 2011].

The ontology matching process suffers from several difficulties, namely due to the inherent subjectivity of data/information which is not completely solved by the ontologies. In fact, it is consensually recognized the natural incapacity to represent univocally the semantics underlying an information system [Ouksel and Ahmed 1999]. These difficulties increase when the ontology matching process is not carried out by a single intervenient, but instead by multiple participants that must agree about the correspondences. In that case, divergences between the participants arise, motivating additional difficulties in establishing consensual correspondences.

Ontology matching negotiation arises as a promising approach, in which the parties are willing to achieve an agreement without losing their decision autonomy, their reasoning capabilities and their private perspectives upon the alignment. There are two types of ontology matching negotiation approaches: relaxation-based approach [Silva, Maio, and Rocha 2005; Maio et al. 2006] and argument-based approach [Laera et al. 2007; Doran et al. 2010].

1.2 Thesis Statement

This thesis advocates the needs and benefits of adopting an explicit, formal and extensible specification of a shared argumentation model between argument-based negotiating agents in order to resolve conflicts and achieve better agreements, and in particular in the scope of ontology matching negotiation.

1.3 Research Methodology

The design-science paradigm [von Alan et al. 2006] was adopted as research methodology. Design research "addresses important unsolved problems in a unique or innovative way or solved problems in more effective or efficient ways" [von Alan et al. 2006] through an iterative and incremental process comprehending two distinct and complementary phases:

- The construction/build phase, whose output is a set of design artifacts, such as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices) and instantiations (prototype system);
- The evaluation phase, which provides essential feedback to the construction phase as to the quality of the design artifacts.

Further, this paradigm suggests seven research guidelines to assist both researchers and readers (e.g. reviewers, editors). These guidelines were followed as described next:

- Design as an artifact. The research efforts resulted in several artifacts. First, the Argument-based Negotiation Process (ANP) is proposed (cf. Chapter 4), which can be seen as a method of negotiation between agents using argumentation. Second, the TLAF argumentation framework (model) was proposed (cf. Chapter 5). Both the ANP and TLAF were then instantiated in the ontology matching domain. Last, and in order to overcome the limitations not addressed by TLAF, the EAF argumentation framework was proposed and instantiated in the ontology matching domain (cf. Chapter 7);
- The relevance of the problem is emphasized by identifying the state-of-the-art limitations (Chapter 3) and by the enumeration of several applications/systems that can profit from ontology matching improvements;
- Design Evaluation. To demonstrate the utility, quality and effectiveness of the produced artifacts, controlled experiments were performed. The obtained results were evaluated qualitatively through standard analytical metrics and by comparison with the alternative designs found on the literature (cf. Chapter 8);

- Research Contributions. Based on novelty, generality and significance of the designed artifacts, a clear and systematized identification of the contributions is presented in the next sub-section and further extended in Chapter 9;
- Research Rigor. The proposed artifacts are formally described in Chapters 4 to 7. The evaluation process makes use of the community's evaluation datasets and compares with the state-of-the-art results;
- Design is seen as a process searching for effective artifacts which requires knowledge of both the application domain and the solution domain. In that respect, the required background knowledge of the application domain (ontology matching) and the solution domain (argumentation and agents) to understand the proposed artifacts are concisely presented in Chapter 2. Further, the state-of-the-art in ontology matching negotiation are thoroughly analyzed in Chapter 3;
- Communication of Research. As a proof of the pertinence and validity of the contributions, most of them were presented and published in international conferences and workshops and, therefore, verified by the respective research community.

1.4 Research Contributions

As a result of the developed research, several scientific contributions were made, better distinguished between abstract (or generic) contributions, and application specific contributions.

1.4.1 Generic-Purpose Contributions

The generic contributions go beyond the ontology matching negotiation process, such that they are applicable to a diverse range of domains such as e-commerce, legal reasoning, decision making. The two main contributions of this kind are:

- An iterative and incremental negotiation process through argumentation (ANP) that can be adopted in Multi-Agents Systems (MAS). On this process argumentation is foreseen as an adequate modeling formalism to reduce the gap between models governing the internal and external behavior of agents. This process introduces and relies on the core notion of argumentation models and suggests the adoption of specific terminology used on the Belief-Desire-Intention (BDI) agents' architecture. This process was first published in [Maio, Silva, and Cardoso 2011a].
- An argumentation framework (TLAF/EAF) that captures and fulfills the core notion of argumentation models (introduced in the proposed argument-based negotiation process). This framework introduced two main novelties: (i) a conceptualization layer

that captures the semantics of the argumentation data employed in a specific context and (ii) defines a novel conceptual relation between argument-schemes called "argument affectations". These two novelties together with the general and intuitive argument structure, the modularity and extensibility features, simplify and reduce the burden on developing argumentation systems. This argumentation framework was first published in [Maio and Silva 2011b].

1.4.2 Ontology Matching Purpose Contributions

The application-specific contributions are focused on applying the generic contributions to the ontology matching negotiation process. The four main contributions of this kind are:

- The identification of several limitations on state-of-the-art approaches of ontology matching negotiation.
- A novel argument-based approach for the ontology matching negotiation problem that is able to overcome most of the identified limitations on the state-of-the-art approaches. This approach follows the iterative and incremental argument-based negotiation process and adopts the generic argumentation framework previously mentioned. For that, particularities of the ontology matching domain are considered.
- The argument instantiation process, which exploits correspondences provided by third party algorithms/agents to generate arguments with minimal user effort. It relies on an interpretation function that provides each agent with the ability to privately interpret correspondences and matching algorithms. Yet, knowing that with rare exceptions matching algorithms do not provide justifications for the proposed correspondences, a very simple yet effective and configurable approach based on a condition function is adopted to identify the premises (i.e. justifications) of arguments and existing conflicts.
- The intrinsic ability of the proposed approach to accommodate more or less types of arguments depending on the agents' matching knowledge and requirements. Therefore, as along as the agents matching knowledge evolves over time, the proposed approach can easily follow such evolution.

These contributions were published in [Maio, Silva, and Cardoso 2011b; Maio and Silva 2010].

1.5 Document Structure

This thesis is composed of ten chapters, one annex and the bibliography, grouped into four parts:

1. The first part includes Chapter 1 through Chapter 3:

   - Chapter 1 describes the context of research, its first motivations, the thesis statement and the contributions of the thesis. It also presents several motivation scenarios in which the use of ontology matching technology is necessary, advisable and/or adopted.

   - Chapter 2 introduces the reader to the background knowledge required in further chapters. The background knowledge refers to three distinct research areas: (i) agent-based systems, (ii) ontology matching and (iii) argumentation.

   - Chapter 3 presents the relevant literature addressing the ontology matching negotiation problem and identifies several limitations that will be overcome as the thesis contributions are described.

2. The second part is concerned with the description of the research work carried out in the scope of this thesis, and includes Chapter 4 through Chapter 8:

   - Chapter 4 describes and proposes a generic argument-based negotiation process that can be adopted in many domains, including the ontology matching domain.

   - Chapter 5 proposes a novel argumentation framework that reduces an existing gap between abstract argumentation frameworks and argumentation systems.

   - Chapter 6 describes the envisioned ontology matching negotiation process by the adoption of the introduced argument-based negotiation process and argumentation framework. It also shows that several state-of-the-art limitations are overcome.

   - Chapter 7 augments the previous generic argumentation framework with new generic features and further describes how those features can be exploited to enhance the envisioned ontology matching negotiation process.

   - Chapter 8 reports on the experiments performed to evaluate and compare the proposed ontology matching negotiation process with the state-of-the-art approaches.

3. The third part concludes the research description:

   - Chapter 9 reviews the performed research and achieved results.

   - Chapter 10 describes the ongoing research efforts and the foreseen future research.

4. The fourth part includes:

   - Annex 1 complements the description of the experiments by providing the details concerning the carried out argument instantiation process.

   - Bibliography presents the literary references used and referred to in this thesis.

Chapter 2

BACKGROUND KNOWLEDGE

The aims of this chapter are three-fold:

- First, to provide a general introduction to agent-based systems and their design principles in relation to argument-based systems;
- Second, to present an overview of the ontology matching research field and its main concepts used within this document;
- Third, to present some relevant literature on argumentation and argumentation systems which are required to understand the major contributions of the thesis.

2.1 Agent-based Systems

In Artificial Intelligence, the notion of agent refers to an entity (e.g. a piece of software) that observes and acts upon an environment on behalf of people and organizations with the purpose of achieve their goals [Wooldridge and Jennings 1995; Wooldridge 2000]. Thus, an agent is an entity that has goals, some (though typically incomplete) knowledge of its circumstances and the capability of acting in such a way as to seek to alter those circumstances in order to achieve its goals. An agent also has the ability to perceive consequences of its actions and to correct them if they move away from realizing current goals. Additionally, autonomous agents have the capability of modifying their goals and the capability of interacting with other agents. A multi-agent system (MAS) consists of a number of agents cooperating, coordinating and negotiating with each other in the same way people and organizations do in their everyday lives, such that some of these agents might be heterogeneous among themselves.

A common agent architecture is based on the BDI model [Bratman 1987; Bratman, Israel, and Pollack 1988; Paglieri and Castelfranchi 2005; Wooldridge 2009]. Within this model, an agent has a set of beliefs that are constantly being updated by sensory input from its environment and a set of desires (wants) that are then evaluated (by desirability and achievability) to form intentions. The agent's goals are represented by its intentions, which are stable over time and are not easily given up. Hence, a crucial problem of BDI agents concerns what the agent's beliefs should be and how those beliefs are used (i) to form new intentions, or (ii) to redraw/revise current intentions. On this matter, contributions of the argumentation research field may be exploited internally by BDI agents since argumentation can be used either for reasoning about what to believe (i.e. theoretical reasoning) and/or for deciding what to do (i.e. practical reasoning). Despite existing differences between both, from a standpoint of first-personal reflection, a set of considerations for and against a particular conclusion are drawn on both [Moran 2001].

Regarding the agents' interactions, they are pursued through the so-called agents' dialogues. According to [Walton and Krabbe 1995], agents' dialogues are classified based on its primary purpose as follows. Information-Seeking Dialogues are those where one participant seeks the answer to some question from another participant. In Inquiry Dialogues, participants collaborate to answer some question whose answer none of the participants knows. In Negotiation Dialogues participants bargain over some resource (e.g. goods or a service) trying to achieve an acceptable agreement to all participants. In Deliberation Dialogues, participants collaborate to decide which course of action must be taken by a mutually acceptable agreement. On both (Negotiation and Deliberation Dialogues), participants may have only partial conflicting information and/or preferences. In Persuasion Dialogues, one participant tries to persuade another participant to accept something (e.g. a proposition, a fact) that is not currently endorsed. In Eristic Dialogues, participant quarrel verbally as a substitute for physical fighting. Despite a clear distinction between each dialogue type, most of the agents' dialogue occurrences involve mixtures of these dialogue types. Yet, according to [Walton and Krabbe 1995], instances of individual dialogue types contained entirely within other dialogue types are said to be embedded. Considering these dialogue types, it is easily perceived that agents may apply argumentation externally in at least three dialogue types: negotiation, deliberation and persuasion. Furthermore, scenarios where agent interactions adopt argumentation as the grounds for the agent internal reasoning allows a reduction in the gap between models governing the internal and external agent behavior [Falappa, Kern-Isberner, and Simari 2009].

2.2 Ontology Matching

Different organizations have different interests and habits, use different tools and have different knowledge. Frequently this has different levels of detail. Due to these differences, organizations adopt different ontologies. Consequently, when those organizations need to interoperate with each other, they need to overcome the heterogeneity problem raised by the adoption of different ontologies. In that sense, and according to [Euzenat and Shvaiko 2007], Ontology Matching is perceived as the appropriate approach to address such heterogeneity in order to enable interoperability between such organizations.

2.2.1 The Matching Process

Informally, Ontology Matching is seen as the process of discovering (semi-) automatically the correspondences between semantically related ontological entities of the ontologies adopted by the organizations wishing to interoperate, such that the output is a set of correspondences called an alignment. Typically, such alignment is further used/exploited as input information for a variety of tasks, such as data transformation [Sheth and Larson 1990; Bernstein and Rahm 2000], information integration [Halevy et al. 2005; Wache et al. 2001; Draper, Halevy, and Weld 2001], information visualization [Gilson et al. 2008], inter-agent communication systems [Eijk et al. 2001; Wiesman, Roos, and Vogt 2001; Bailin and Truszkowski 2003], peer-to-peer information sharing [Zaihrayeu 2006], query answering [Lopez, Motta, and Uren 2006; Mena et al. 1996] or for navigation on the semantic web [Sabou, Lopez, and Motta 2006].

Formally, as stated in [Euzenat and Shvaiko 2007], the matching process can be represented as a function f which, from a pair of ontologies to match O_1 and O_2, a set of parameters p, a set of oracles and resources res and an input alignment A, it returns an alignment A' between the matched ontologies.

$$A' = f(O_1, O_2, p, res, A)$$

Ontologies O_1 and O_2 are often denominated as source and target ontologies respectively. When the matching process relates more than a pair of ontologies, it is referred to as holistic matching [Bellahsene, Bonifati, and Rahm 2011]. Therefore, the matching process can be redefined to take as input a set of ontologies $\{O_1, \dots, O_n\}$ with $n \geq 2$ such that:

$$A' = f(O_1, \dots, O_n, p, res, A)$$

In the scope of this thesis, by default the matching process is considered to occur only between a pair of ontologies.

An alignment is a set of correspondences between entities belonging to different ontologies which are expressed according to:

- Two entity languages Q_{L_1} and Q_{L_2} associated with the ontologies languages L_1 and L_2 of matching ontologies (respectively) defining the matchable entities (e.g. classes, object properties, data properties, individuals);
- A set of relations R that is used to express the relation holding between the entities (e.g. equivalence, subsumption, disjoint);
- A confidence structure φ that is used to assign a degree of confidence on a correspondence. It has a greatest element $\top$ and a smallest element $\bot$. The most common structure is the real numbers in the interval $[0, 1]$, where 0 represents the lowest confidence and 1 represents the highest confidence.

Given that, a correspondence (or a match) is formally defined as a 5-tuple (id, e, e', r, n) where:

- id is a unique identifier of the given correspondence;
- e is an entity of O_1 expressed in a ontology language Q_{L_1};
- e' is an entity of O_2 expressed in a ontology language Q_{L_2};
- r is the relation holding between e and e' such that $r \in R$;
- n is the degree of confidence in the relation holding between e and e' such that $n \in \varphi$.

This correspondence structure is able to express either simple or complex correspondences. Simple correspondences establish a binary relation (e.g. $=, \geq, \leq$) between a single entity of ontology O_1 with a single entity of ontology O_2.

Example 2.1 – Expressing a simple correspondence

The simple correspondence $(c_1, O_1{:}Address, O_2{:}Location, =, 1.0)$ express the semantic equivalence (i.e. $=$) between the entity $Address$ of ontology O_1 and the entity $Location$ of ontology O_2 with a degree of confidence of 1.0.

Complex correspondences are thought as the ones establishing non-binary relations such as operators (e.g. concatenation, split, arithmetic operations) and logical connectors for the purpose in hand. In the simplest expression, e and e' are sets of ontological entities.

Example 2.2 – Expressing a complex correspondence

For the purpose of data translation, the complex correspondence $(c_2, \{O_1{:}Address.street, O_1{:}Address.zipcode\}, O_2{:}Location.description,$

$Concatenation, 0.85)$ expresses that the instance values of entity $description$ associated with entity $Location$ (both in ontology O_2) should be the result of the concatenation of the instance values of entities $street$ and $zipcode$ associated with entity $Address$ of ontology O_1. The degree of confidence on this correspondence is 0.85.

More details on how to express complex correspondences based on the entity languages associated with the matching ontologies are provided in [Maedche et al. 2002; Euzenat 2008; Euzenat, Scharffe, and Zimmerman 2007; Scharffe 2011].

The alignment resulting from a matching process is characterized based on two distinct properties: (i) the alignment level and (ii) the alignment cardinality.

The level of an alignment is used to characterize the type of correspondences (grounded on its content) the alignment has or supports. In [Euzenat 2004] three alignment levels were identified:

- Level 0: resembles the most basic correspondences where a single entity of source ontology is related to a single entity of target ontology. The ontology entities are the ones provided by the ontology language (L). Consequently, correspondences are independent of the entity language (Q_L). It is required that each entity is discrete and identifiable by a path or URI;
- Level 1: this is a refinement of Level 0 allowing related entities of correspondences to be a set. Therefore, it stills independent of the entity language. Essentially, it fulfills the gap between Level 0 and Level 2;
- Level 2: resembles the most complex correspondences. It consists of a set of expressions of a particular entity language (Q_L) with variables. In this case, correspondences are typically directional and correspond to a clause ($\forall \overline{x_f}(f \Longrightarrow \exists \overline{x_g})$ where variables of the left-hand side are universally quantified over the whole formula and the variables of the right-hand side (the ones not occurring in the left-hand side) are existential quantified.

Example 2.3 – Correspondences expressed through an entity language

A logic-based language such as the SWRL [Horrocks et al. 2004]) can be used to express such complex correspondences (e.g. $grandParent(x,y) \Rightarrow parent(x,z) \wedge parent(z,y)$). In the database world, these kinds of correspondences can be expressed in SQL (e.g. $description = street + ' ' + zipcode$).

The cardinality of an alignment considers the two possible orientations of the alignment (from O_1 to O_2 and from O_2 to O_1) and it is based on the notion of total alignment and injective alignment:

- A total alignment means that all the entities of the source ontology must be successfully matched to the target ontology.
- An injective alignment means that all entities of the target ontology are part of at most one correspondence.

The orientation characteristics of the alignment depend on the ontology entity languages adopted for the alignment and the kind of relations that can be established between the ontological entities. For example, an alignment that only considers the equivalence relation between ontological entities is said to be bijective if it is total from both O_1 and O_2 and it is injective from one of them (from O_1 to O_2 or vice-versa). However, when relations other than equivalence are used, the injectivity does not guarantee the reversibility of the alignment used as a transformation.

Traditionally, in database schema modeling, the cardinality of a relation is defined in terms of one-to-one (1:1), one-to-many (1-n), many-to-one (m:1) and many-to-many (m:n). Yet, in order to match the notion of injective and total alignment, [Euzenat 2003] proposed another (but similar) nomenclature:

- 1 denotes a total and injective alignment;
- ? denotes an injective alignment;
- + denotes a total alignment;
- * for an alignment that is not neither total nor injective.

Therefore, the possible alignment cardinalities are: ?:?, ?:1, 1:?, 1:1, ?:+, +:?,1:+, +:1,+:+, ?:*, *:?, 1:*, *:1, +:*, *:+, *:*.

The possible values of these two properties (level and cardinality of the alignment) can be used as input parameters (or constraints) to the matching process.

2.2.2 Forms of Heterogeneity

Diverse forms of heterogeneity exist and must be carefully taken into consideration during the matching process. Below, the four most common types of heterogeneity are described.

The Syntactic heterogeneity occurs when two ontologies are expressed in two distinct computer formats (or ontology languages). This kind of mismatch usually occurs when the matching ontologies are modeled using different knowledge representation formalisms (e.g. OWL vs.

F-logic or XML schema vs. Relational Database Schema). However, it also occurs when matching a physical model (e.g. a directory hierarchy) with a conceptual model. Generally, this type of heterogeneity is addressed at a theoretical level by establishing equivalences between the constructs of the computer formats involved. Such equivalences are used to translate one ontology from one format to another by preserving the original meaning or, at least, minimizing the meaning loss [Euzenat and Stuckenschmidt 2003].

The Terminological heterogeneity occurs when two ontologies use different names to refer to the same entities. Usually, this is caused by the adoption of (i) different natural languages (e.g. English vs. Portuguese), (ii) synonyms within the same natural language (e.g. automobile and car) or (iii) different technical sublanguages. Thus, this kind of heterogeneity is usually tackled by exploiting natural languages resources such as thesaurus, dictionaries and lexical databases (e.g. WordNet [Miller et al. 1990; Fellbaum 1998]).

The Conceptual heterogeneity occurs when the same domain of interest is modeled in two distinct ways. According to [Klein 2001; Visser et al. 1997], it might happen for two reasons: (i) an explicitation mismatch and (ii) a conceptualization mismatch. The former relies on the way concepts are expressed. Generally, this kind of mismatch happens when objects are described by means of totally different primitive objects.

> **Example 2.4 – Heterogeneity introduced by an explicitation mismatch**
>
> A print industry organization probably adopts the colors cyan, magenta, yellow and black as the primitive objects to define any other color, i.e. they describe colors based on the CMYK color system. On the other hand, a scientist may use as primitive objects the colors red, green and blue, i.e. the RGB color system. Thus, the same color (e.g. gray) may be described in two distinct manners.

The conceptualization mismatch relies on different conceptualizations of the same concepts as the use of different (and sometimes, equivalent) axioms to describe the same concept. According to [Benerecetti, Bouquet, and Ghidini 2001], three main reasons lead to this kind of mismatch:

- Difference in coverage: both ontologies describes the same domain of interest with the same level of detail and from a unique perspective but with an overlapping common part (e.g. one ontology describes all countries of Europe and the other one describes all countries of Europe and Asia);
- Difference in granularity: both ontologies describes the same domain of interest from a unique perspective but with different levels of details (e.g. one ontology only describes cities but the other describes the cities and existing buildings in those cities);

- Difference in perspective: both ontologies describe the same domain of interest with the same level of detail but from a difference perspective (e.g. one ontology describes countries based on a political location and the other describes cities based on a geological location).

The Pragmatic heterogeneity as identified in [Bouquet et al. 2004] occurs when two ontological entities having the same semantic interpretation [Goguen 1999; Hogger 1990] are interpreted differently by humans due to the context of use. Therefore, and according to [Euzenat and Shvaiko 2007], this kind of heterogeneity is the most difficult to be detected by a computer and even more difficult to solve.

Several kinds of heterogeneity may occur simultaneously in ontologies. However, the ontology matching process is usually concerned with the terminological and conceptual types of heterogeneity only. The syntactic heterogeneity is usually addressed by the so called "translators" or "wrappers" [Wiederhold 1992], and pragmatic heterogeneity is starting to be addressed in scope of context-aware systems [Baldauf, Dustdar, and Rosenberg 2007].

2.2.3 Matching Techniques and Dimensions

Matching algorithms (or simply matchers) can be classified based on many independent dimensions. From the definition of the matching process previously introduced, the algorithms could be classified according to four relevant dimensions.

2.2.3.1 The Input Dimension

The input dimension concerns the input model and data on which matching algorithms relies on. Thus, algorithms can be classified based on the computer format (i.e. data model) in which the ontologies are expressed. In that sense, algorithms may support the relational, object-oriented and entity-relationship models (e.g. Artemis [Castano, Antonellis, and Vimercati 2000]), XML and relational models (e.g. Cupid [Madhavan, Bernstein, and Rahm 2001]) or RDF and OWL models (e.g. NOM [Ehrig and Sure 2004], QOM [Ehrig and Sure 2004], FOAM [Ehrig and Sure 2005], FALCON-AO [Jian et al. 2005], OLA [Euzenat and Valtchev 2004], oMap [Straccia and Troncy 2005]).

2.2.3.2 The Ontological Dimension

The ontological dimension related to the kind of ontological data exploited. Some algorithms exploit the schema-information (e.g. Cupid [Madhavan, Bernstein, and Rahm 2001], COMA [Do and Rahm 2002]), while others algorithms exploit the instance data (e.g. GLUE [Doan et al. 2004]) or both (e.g. QOM [Ehrig and Sure 2004]). Nevertheless, in general algorithms do not exploit all of the available information in the data model. Considering only algorithms exploiting schema information, some may focus on the labels assigned to the entities (i.e. syntactic algorithms) while others algorithms may focus on the relations that entities have with the others entities (i.e. external structural algorithms) or even in the types of their attributes (i.e. internal structural algorithms).

2.2.3.3 The Output Dimension

With regards to the output dimension, algorithms can be classified based on the result they produce. Considering that the result produced by algorithms is always an alignment, the resulting classification is obviously based on (i) the alignment cardinality, (ii) the kind of relations that are provided (i.e. the alignment level), (iii) the fact that the proposed alignment has to be final or not, (iv) the ability to grade the correspondences or just indicate if a relation holds or not (0 or 1), (v) the type of grade provided (similarity vs. dissimilarity) and its meaning.

2.2.3.4 The Process Dimension

The process dimension is concerned with the internal characteristics of the matching algorithms. First, as proposed in [Rahm and Bernstein 2001], algorithms are categorized into basic (or elementary) algorithms and hybrid algorithms. A basic algorithm generates correspondences based on a single matching criterion (e.g. considering a specific ontological dimension) while a hybrid algorithm generates correspondences combining the results of multiple basic algorithms, i.e. it exploits multiple matching criterions. In [Shvaiko and Euzenat 2005], the matching criterions are classified from two distinct perspectives.

The first perspective concerns the granularity and the way algorithms interpret the input. In terms of granularity, algorithms are classified as (i) Element-level, which are those that compute correspondences by analyzing each entity individually, ignoring the existing relationships with other entities and (ii) Structure-level, which are those that compute correspondences by analyzing how entities appear together in a structure, through existing relationships between entities. With respect to the way algorithms interpret the input data, they are classified as:

- Syntactic, which are those that interpret the input regarding its sole structure through some clearly defined method;
- External, which are those that interpret the input in the light of some external resources of a domain or of common knowledge;
- Semantic, which are those that interpret the input using some formal semantics (e.g. semantic theoretical models). In this case, the outputs are also justified based on the adopted formal semantics.

The second perspective is based on the type of data used as input. At a first level, it is distinguished by algorithms working on:

- Terminological data (i.e. strings). Terminological matchers can be classified further either as string-based (those that consider strings as sequences of characters) or as linguistic (those that consider strings as terms of natural language);
- Structure (structural). The structural matchers can be classified either as internal (those that consider the internal structure such as attributes and the data types) or as relational (or external, when considering the relations an entity has with the other entities);
- Models (or semantics). These matchers require a semantic interpretation of the ontologies;
- Extensional (data instances). These matchers exploit the current population of the ontologies.

Basic algorithms can be multiple classified as graphically depicted in Figure 2.1 (extracted/adapted from [Euzenat and Shvaiko 2007]), where the first layer represents the first perspective (Granularity/Input Interpretation), the second layer represents the basic algorithms and the third layer represents the second perspective (kind of input).

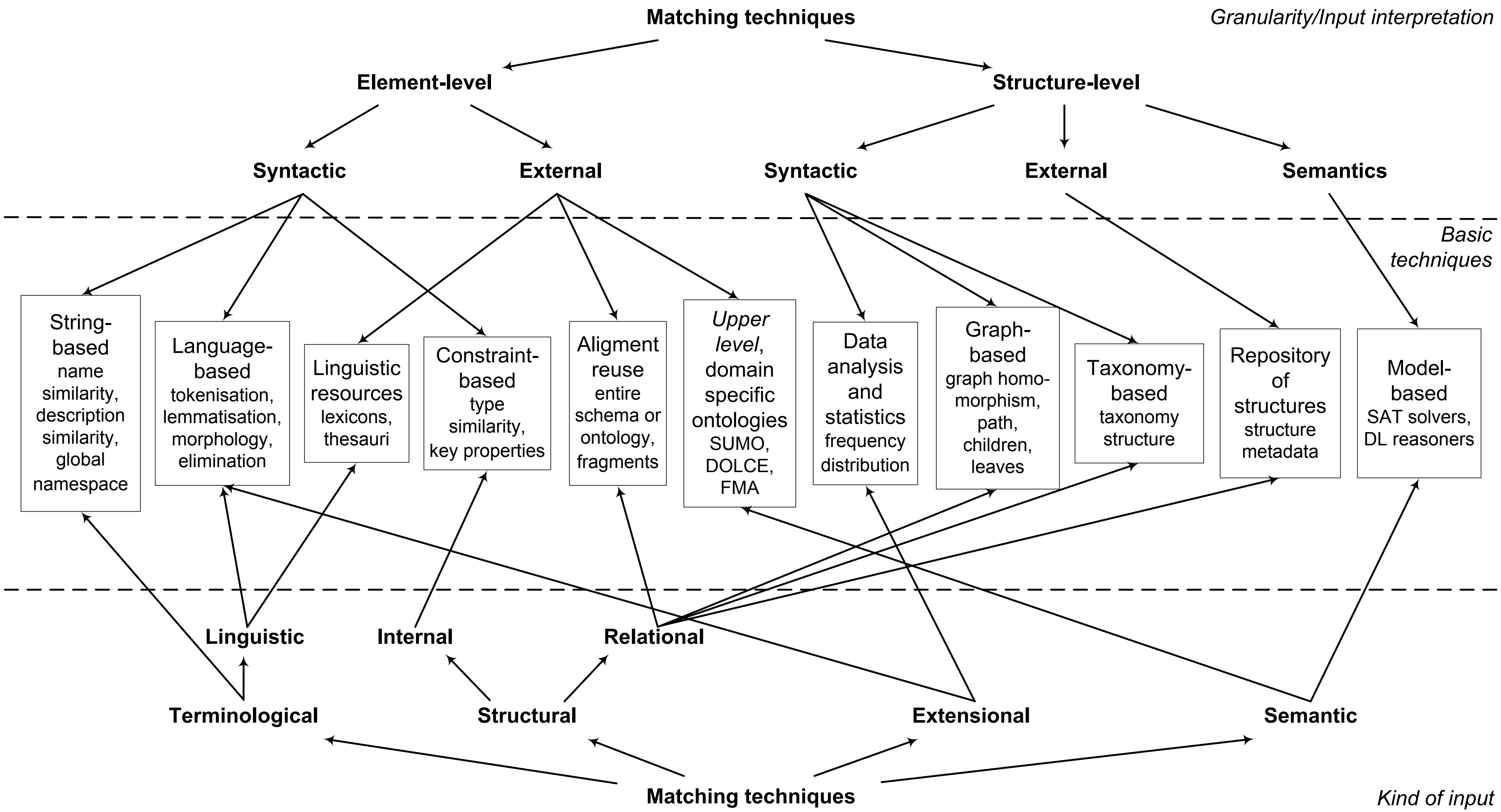

Figure 2.1 – Multi-classification of basic matching algorithms based on two distinct perspectives

2.2.4 Matching Strategies

State-of-the-art ontology matching systems [Euzenat and Shvaiko 2007; OAEI 2011] apply at least two of basic matching techniques yielding different and complementary competencies, to achieve better results. For that, from an architectural perspective, systems follow two distinct approaches: (i) a sequential approach and (ii) a parallel approach.

Respecting the former, the system includes as many algorithms as needed such that each algorithm computes a set of correspondences which is used to seed the next algorithm and so on. Respecting the latter approach, each included algorithm individually computes one set of correspondences, whose results are aggregated through a function (e.g. min, max, linear average, weighted average, OWA [Ji, Haase, and Qi 2008]) into one single alignment. Yet, systems may follow a hybrid approach where both approaches (sequential and parallel) co-exist.

Regardless of the adopted architecture, the result of matching is a large set of correspondences. The satisfactory set of correspondences that will be part of the resulting alignment remain to be extracted. This is the role of specialized extraction methods, which act on sets of correspondences already generated.

Threshold-based methods are seen as the simplest approach. A threshold method allows the retainment of the correspondences that have the higher similarity values on the relation holding between ontological entities. Several kinds of thresholds (e.g. Hard Threshold, Delta Threshold) have been identified in literature [Ehrig and Sure 2004; Do and Rahm 2002; Melnik, Rahm, and Bernstein 2003] and are summarized in [Euzenat and Shvaiko 2007]. Despite the simplicity of the methods, finding out the right threshold value(s) is still an open and big issue.

More complex methods can be used to extract an alignment. For example, strengthening and weakening functions (e.g. sigmoid and trigonometric functions) proposed by [Ehrig and Sure 2004], aim to explicitly create two zones: the positive and negative, i.e. to keep/discard correspondences. Another example is applicable to the alignment scenarios where the cardinality of the expected alignment is known *a priori*. The cardinality is used to choose an alignment extraction method performing either (i) local optimizations (e.g. the algorithm proposed in [Gale and Shapley 1962] to compute stable marriage problems) or (ii) global optimizations (e.g. the Hungarian method proposed in [Munkres 1957]).

Despite the matching systems allowing some parameterization, such as (i) threshold values, (ii) matcher weights (for a specific aggregation function) or even (iii) choosing the list of matching algorithms to participate in the alignment; the fact is that they are restrictive with respect to the internal process of the system. That is, one cannot set up the architecture of the system and the corresponding workflow. Furthermore, since algorithms should not be chosen only with respect to the given data but also adapted to the problem to be solved, the selection of the most suitable algorithm/system is still an open issue [Euzenat and Shvaiko 2007], but already addressed in [Maio and Silva 2009b; Ngo et al. 2011; Saruladha, Aghila, and Sathiya 2011]

2.2.5 Evaluation Measures

An important aspect of ontology matching is assessing the quality of resulting alignment. For that purpose, measures can be classified as (i) compliance measures and (ii) formal or logic-based measures.

Compliance measures are those that compare system outputted alignment with a reference alignment (or gold standard) which should be the complete set of all correct correspondences. The most commonly used are Precision and Recall (originating from information retrieval), or their harmonic mean, referred to as F-Measure. Precision corresponds to the ratio of correctly found correspondences over the total number of found correspondences while Recall corresponds to the ratio of correctly found correspondences over the total number of expected correspondences. Yet, in order to improve these measures, a Relaxed Precision and Relaxed Recall have been proposed [Ehrig and Euzenat 2005]. These are based on the idea that a proposed correspondence not existing in the reference alignment might be similar to an existing one. Instead of considering it incorrect, one can measure the correction effort to transform such correspondence into a correct one.

Semantic Precision and Semantic Recall are formal measures based on the comparison of deductive closure of both alignments (i.e. proposed and reference alignment) instead of a syntactic comparison [Euzenat 2007].

A major drawback to all these (i.e. compliance and formal) measures (except incoherence-based ones) is that they are grounded in the existence of one reference alignment which might not be available in real-world scenarios. A set of logic-measures based on the incoherence of correspondences has been proposed in [Meilicke and Stuckenschmidt 2008].

Besides these measures, there are other measures concerned with resource consumption (e.g. speed, memory, scalability), referred to as performance measures, that can be used to compare systems instead of the resulting alignments.

2.2.6 Summary

The previous survey aimed to show the huge diversity of ontology matching algorithms and emphasized the difficulties in combining them into accurate matching systems. While the diversity and combination possibilities allow the problem to be addressed from different perspectives, this leads to different proposals of correspondences and eventually to conflicts between parts involved in establishing the alignment.

2.3 Argumentation

There is an abundance of relevant literature in argumentation and argumentation systems. This section focuses on four main facets of argumentation: (i) the applied reasoning, (ii) the argument modeling formalisms, (iii) the arguments acceptability and (iv) the arguments interchange format. Each one of these facets is the subject of the following sub-sections.

2.3.1 Reasoning in Argumentation

Until the 1960s, the general consensus was that a good reasoning had to be deductively valid [Pollock 1987]. Since then, several important pieces of work [Pollock 1967; Pollock 1971; Rescher 1977; Toulmin 1958; Pollock 1987] began questioning the idea, by which recognizing other kinds of reasoning not deductively valid, but which clearly confer justification about their conclusions. Unlike deductive reasoning, conclusions of non-deductive reasoning (e.g. perception, induction, probabilistic reasoning, temporal projection) are not guaranteed to be true, since there is the possibility that new information leads to changes in previous conclusions, thus referred to as defeasible reasoning. In non-deductive reasoning, it is acceptable to derive conclusions that can be "defeated" by considerations that make it unreasonable to maintain previously derived conclusions. Using defeasible reasoning, a conclusion might be retracted by the addition of information without retracting any of the premises from which the previous conclusion was inferred. In contrast, the deductive reasoning is not "defeasible" because the deductive conclusions cannot be rationally denied without denying one or more of the premises.

The adoption of defeasible reasoning by the agents promotes the exchange of arguments between them, which potentially leads to persuasion, i.e. to derive new conclusions by the effect of the received arguments.

2.3.2 Argument Modeling Formalisms

The abstract argumentation frameworks AF [Dung 1995], BAF [Cayrol and Lagasquie-Schiex 2005a] and VAF [Bench-Capon 2003] are suitable formalisms to represent many different situations without being committed to any domain of application. Thus, the main concepts of these argumentation frameworks are described as follows.

The Argumentation Framework (AF) as proposed by Dung [Dung 1995] has two core entities: (i) *Argument*, and (ii) R_{att}, a binary relation between arguments. The R_{att} relation is known as the attack relation. An AF can be defined as a tuple $AF = (A, R_{att})$ where A is a set of arguments and R_{att} is a relation on A such that $R_{att} \subseteq A \times A$. For any two arguments, say a_1 and a_2, such that $a_1, a_2 \in A$, one says that a_1 attacks a_2 iif $(a_1, a_2) \in R_{att}$.

An AF instance may be represented by a directed graph whose nodes are arguments and edges represent the attack relation (example depicted in Figure 2.2).

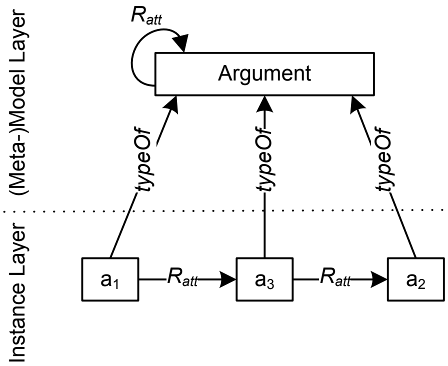

Figure 2.2 – Graph representation of an instantiation of AF

Yet, in Dung's work the attacks always succeed (i.e. an attack defeats the attacked arguments). While it is reasonable that attacks always succeed when dealing with deductive arguments, in domains where arguments lack this coercive force, arguments provide reasons which may be more or less persuasive and their persuasiveness may vary according to their audience. Accordingly, it is necessary to distinguish between attacks and successful attacks (i.e. defeats) prescribing different strengths to arguments on the basis of the values they promote and/or their motivation, in order to accommodate the different interests and preferences of an audience.

With that purpose, the Value-based Argumentation Framework (VAF) [Bench-Capon 2003] extended the AF [Dung 1995] with (i) the concept of *Value* and (ii) the function *promotes* relating an *Argument* with a single *Value*. Therefore, a VAF can be defined as 4-uple $VAF = (A, R_{att}, V, promotes)$ where A and R_{att} means the same as in the AF, a non-empty set of values V and the function $promotes: A \rightarrow V$ to map elements from A to elements of V. Consequently, an *audience* for a VAF instance corresponds to a binary preference relation $P \subseteq V \times V$ which is transitive, irreflexive and asymmetric. A pair $(v_1, v_2) \in P$ means that value v_1 is preferred to v_2 in the audience P. An attack between two arguments (i.e. $(a_1, a_2) \in R_{att}$) where a_1 promotes a value v_1 and a_2 promotes a value v_2 succeeds (i.e. a_1 defeats a_2) iif the adopted audience prefers v_1 to v_2, otherwise the attack fails. Figure 2.3 depicts an example of a VAF.

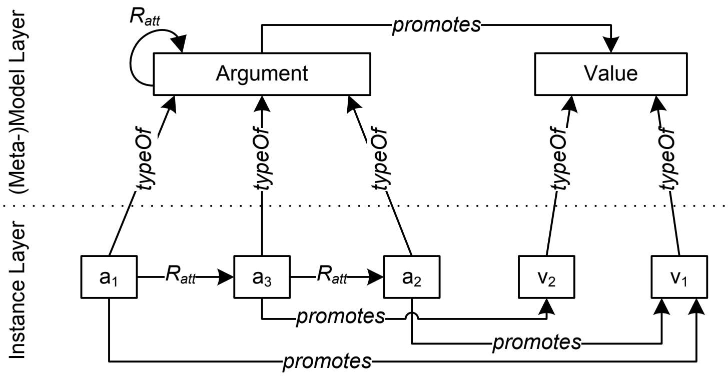

Figure 2.3 – Graph representation of an instantiation of VAF

The AF and the VAF assume that an argument a_1 supports an argument a_2 if a_1 attacks and therefore defeats an argument a_3 that attacks argument a_2. Thus, these frameworks only explicitly represent the negative interaction (i.e. attack), while the positive interaction (i.e. defense/support) of an argument a_1 to another argument a_2 is implicitly represented by the attack of a_1 to a_3. Since support and attack are related notions, these modeling approaches adopt a parsimonious strategy, which is neither a complete nor a correct modeling of argumentation [Cayrol and Lagasquie-Schiex 2005b].

Conversely, the Bipolar Argumentation Framework (BAF) [Cayrol and Lagasquie-Schiex 2005a] assumes the attack relation is independent of the support relation and both have a diametrically opposed nature and represent repellent forces. As a result, BAF extended the AF with the support relation (R_{sup}) in order to be explicitly represented. Thus, a BAF can be defined as a 3-uple $BAF = (A, R_{att}, R_{sup})$ where A and R_{att} means the same as in the AF, and R_{sup} is a binary relation on A such that $R_{sup} \subseteq A \times A$. Given that, for any two arguments, say a_1 and a_2, such that $a_1, a_2 \in A$, one says that a_1 supports a_2 iif $(a_1, a_2) \in R_{sup}$. Figure 2.4 depicts an example of a BAF.

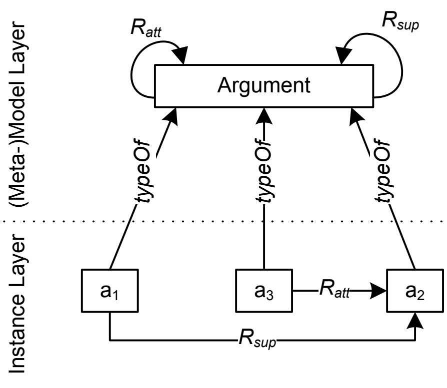

Figure 2.4 – Graph representation of an instantiation of BAF

For all of these frameworks, an argument is anything that may attack/support or be attacked/supported by another argument. The absence of an argument structure and semantics makes these frameworks suitable for the study of independent properties of any specific aspect that are relevant for any argumentation context that can be captured and formalized accordingly. On the other hand, this abstract nature represents an expressiveness limitation to the direct adoption of specific application contexts [Baroni and Giacomin 2009]. To overcome this limitation, argumentation-based systems usually adopt an abstract argumentation framework and extend it in order to get a less abstract formalism, dealing in particular with:

- The construction of arguments and their structure;
- The conditions under which argument-relations (i.e. attack and/or support) are established;
- The conditions under which an argument is defeated.

Nevertheless, these argumentation frameworks do not provide any constructs or features for facilitating and governing the framework extension process.

Recently, a general abstract framework for structured arguments has been presented by Prakken [Prakken 2010] where arguments are defined as inference trees formed by applying two kinds of inference rules: *strict* and *defeasible*. Based on that, arguments can be attacked by three syntactic categories of arguments:

- Undercutting arguments: are those that attack the inferential connection between reasons (or premises) and the conclusion. An argument a_1 is said to be an undercutting argument of a_2 when a_1 claims the negation of a reason used by a_2 to achieve its conclusion;
- Rebutting arguments: are those that are seen as reason for the opposite conclusion. An argument a_1 is said to be a rebutting argument of a_2 when a_1 claims c and a_2 claims its negation (i.e. $\neg c$). In this case, argument a_2 is also a rebutter of a_1;
- Undermining arguments: are those that attack the inference mechanism of an argument. An argument a_1 is said to be an undermining argument of a_2 when it claims the inference mechanism (e.g. a rule) used by a_2 is not valid or applicable in the context.

This framework integrates other relevant work on structured argumentation [Pollock 1987; Pollock 1994; Vreeswijk 1997; Caminada and Amgoud 2007; Verheij 2003], which can be seen as a special case of the general framework.

2.3.3 Arguments Acceptability

Given the nature of any argumentation process, one will have arguments for and against the issue(s) under discussion. Thus, it is clear that arguments cannot stand all together (i.e. some arguments hold while other arguments do not hold). In that sense, arguments need to be subject to an evaluation process to decide their status (e.g. accepted, rejected). An argument evaluation process typically comprises of two phases:

- A valuation of the arguments, which can be based only on the binary interactions between arguments (i.e. attacks and/or supports), or take into account the strength of each argument;
- A selection of some arguments based on argumentation semantics, where the valuation results are used to distinguish different levels/classes of arguments (e.g. the class of argument that are skeptical accepted and the class of argument that are credulous accepted).

In this context, argumentation semantics are a formal definition of a declarative or procedural method. In the literature, two kinds of approaches are used to define argumentation semantics:

- The labeling-based approaches specify how to assign to each argument a state label. The labels are taken from a predefined set of labels that represent the state alternatives. Two typical sets of predefined labels are $\{in, out\}$ and $\{in, out, undecided\}$ where the state in means the argument is accepted, the state out means the argument is rejected and the state $undecided$ means that argument is neither accepted nor rejected;
- The extension-based approaches postulate how to derive a set of extensions from an argumentation framework. An extension is a sub-set of the arguments that satisfy a given criteria. For example, an extension might be a set of arguments that are collectively acceptable or conflict-free. Yet, extension-based argumentation semantics are equivalently expressed by a labeling-based approach adopting at least two labels such that one label (typically in) is set to all arguments belonging to the extension membership and the other label (typically out) is set to the remaining arguments.

Next is presented the most relevant semantics for the AF [Dung 1995]. An extension-based approach is followed because this approach prevails in the literature.

Considering that $AF = (A, R_{att})$, one says that an argument y is attacked by a set of arguments S such that $S \subseteq A$ if S contains at least one argument attacking y. Grounded on that, the following notions are defined:

- An argument $a \in A$ is *acceptable* with respect to a set of arguments S, i.e. $acceptable(a, S)$, iif $\forall x: x \in A \wedge (x, a) \in R_{att} \rightarrow \exists y: y \in S \wedge (y, x) \in R_{att}$;
- A set of arguments S if *conflict-free* iif $\nexists x, y: x, y \in S \wedge (x, y) \in R_{att}$;
- A *conflict-free* set of arguments S is *admissible* iif $\forall x: x \in S \rightarrow acceptable(x, S)$;
- A set of arguments S is a *preferred extension* iif it is maximal (with respect to set inclusion) *admissible* set of A.

A preferred extension represents a consistent position within an argumentation framework, which is defensible against all attacks and cannot be further extended without introducing a conflict. Yet, for the same argumentation framework multiple preferred extensions can exist. Usually, this occurs when for a given argument it is necessary to make use of an additional criteria (not specified by the extension-based approach) to decide if the argument belongs to the set or not. Given that, one considers that (i) an argument is *sceptical admissible* if it belongs to any preferred extension and (ii) an argument is *credulous admissible* if it belongs to at least one preferred extension, but does not belong to all.

It is worth noticing that these notions (acceptable, admissible, conflict-free and preferred extension) are redefined accordingly for the VAF (cf. [Bench-Capon 2003] for details) and for BAF (cf. [Cayrol and Lagasquie-Schiex 2005a] for details). For a complete description on argumentation semantics applied to abstract argumentation frameworks refer to [Baroni and Giacomin 2009].

Most of the argumentation systems (e.g. the Prakken version of ASPIC [Prakken 2010]) use the abstract level provided by the adopted abstract argumentation framework as an abstraction of the overall system to make logical inferences. That is, systems start with a knowledge base, which is used to instantiate the adopted argumentation framework and then apply a given abstract argumentation semantics to select the conclusions of the associated sets of arguments. However, as studied in [Prakken 2010] and [Caminada and Amgoud 2007], in light of the arguments' content only, it is still possible that sets of arguments selected by an abstract argumentation criterion yield to inconsistent conclusions.

2.3.4 Arguments Interchange Format

One of the barriers faced by argumentation systems is the lack of a shared, agreed notation for interchange arguments. To overcome such barriers, several mark-up languages have been proposed in the context of tools concerned with (i) the arguments construction and visualization (cf. [Kirschner, Shum, and Carr 2003]) and (ii) the analyses and study of human argument (cf.

[Buckingham et al. 2007; Macagno et al. 2006; Buckingham, Motta, and Domingue 2000]). However, these attempts share two major limitations:

- Each particular language is mainly designed for the purpose of the specific tool rather than for facilitating the interoperability of arguments among a variety of tools;
- They are mainly focused on enabling users to structure arguments through diagrammatic linkage of natural language sentences.

To overcome these limitations, the Argument Interchange Format (AIF) [Chesñevar et al. 2006] has been proposed with the aim of enabling true interoperability of arguments and argument structures. In that sense, AIF is seen as an argument description language that can be extended beyond a particular argumentation theory and schemes. According to [Rahwan and Reed 2009], AIF as it stands represents a consensus 'abstract model' established by researchers across fields of argumentation, artificial intelligence and multi-agent systems. This abstract model is commonly implemented in ontology representation languages such as RDF(S) [Guha and McBride 2004] and OWL [Dean 2004].

2.3.5 Summary

The above survey aimed to introduce the core concepts underlying the argumentation research field and to show how those concepts are exploited by argumentation systems. Yet, it has highlighted the gap between the abstract argumentation frameworks and the argumentation systems.

Chapter 3

State-of-the-Art on Ontology Matching Negotiation

This chapter surveys the literature related to the ontology matching negotiation problem. There are two main approaches to ontology matching negotiation:

- Relaxation-based negotiation, in which each agent's exigencies about correspondences are privately relaxed;
- Argument-based negation, in which the agents exchange arguments to foster the change on the opponent's position upon the correspondences.

The following sections address these approaches, each one concluding with the systematization of their current limitations.

3.1 Relaxation-based Ontology Matching Negotiation

The ontology matching negotiation problem was firstly addressed in literature by [Silva, Maio, and Rocha 2005; Maio et al. 2006].

Organizations wishing to interoperate are represented by artificial agents that act on their behalf during the negotiation process. Thus, the negotiation occurs between two agents such that one of the agents represents the organization using the source ontology (O_1) and the other agent represents the organization using the target ontology (O_2). Yet, it is assumed that both agents are honest and co-operative with each other and each one is capable of devising an alignment between source and target ontologies. Additionally, the agents are free to request third party entities/agents to carry out their private tasks.

The object of the negotiation is the content of the alignment to be established between the source and target ontologies. The agents negotiate about the inclusion or exclusion of each correspondence suggested by one of them. The value that each agent associates with a correspondence is highly subjective and it can depend on several factors such as:

- All the ontology matching reasons previously introduced (e.g. terminological, structural, semantic);
- Relations between other correspondences (e.g. some correspondences may imply or depend on others correspondences in a valid alignment);
- Pertinence (importance) of the correspondence with respect to the interoperability.

3.1.1 Alignment Generation Process

In order to devise an alignment, each agent is free to follow the most appropriate approach, including the selection of the matching algorithms and their combination (cf. section 2.2.4).

As a generalization and simplification of the alignment generation process, it is assumed that for each kind of relation that a matching algorithm is able to provide (e.g. equivalence, subsumption, concatenation, split, currency converter), it has a distinct utility function (u). Thus, based on a set of parameters ($p_1, p_2, \dots, p_n$) each utility function is responsible for assigning the degree of confidence (n) of the correspondences establishing the relation (r). Furthermore, adopting a threshold-based selection approach (cf. section 2.2.4) a threshold value (t_r) constrains the correspondences to consider (see Table 3.1).

Table 3.1 – Example of information used in the alignment generation process

Matcher	Relation	Utility Function (u)	t_r
M_1	Equivalence	$u_e(p_1, p_2)$	0.90
M_1	Subsumption	$u_s(p_1, p_2, p_3)$	0,75
M_2	Concatenation	$u_c(p_1, p_2, p_3, p_4, p_5)$	0.60
M_3	Split	$u_s(p_1, p_2, p_4, p_5)$	0,65

3.1.2 Relaxation and Convergence Mechanisms

In order to provide relaxation capabilities to the agents, two novelties were proposed.

First, to extend the single rejection threshold approach to a four-level threshold approach (i.e. three new threshold values):

- Mandatory threshold (t_m), which determines the utility function value above which it is fundamental that the correspondence is accepted by the opponent agent;
- Proposition threshold (t_p), which determines the utility function value above which correspondences are proposed for acceptation to the opponent agent;
- Negotiable threshold (t_n), which determines the utility function value above which correspondences are considered as negotiable (i.e. correspondences are not proposed but in case the opponent agent proposes them, they might be accepted).

These threshold values must satisfy the condition: $0 \leq t_r \leq t_n \leq t_p \leq t_m \leq 1$. Therefore, five categories of correspondences are defined according to confidence value and the identified threshold values (see Figure 3.1):

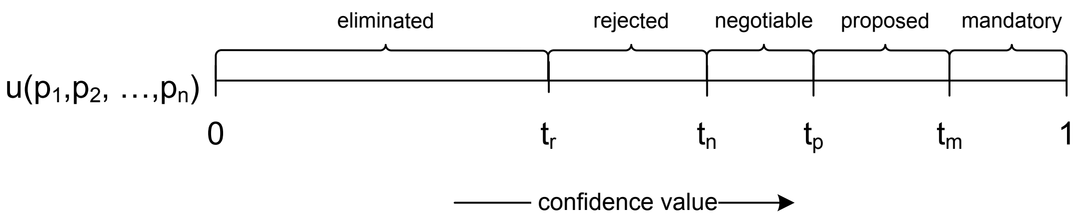

Figure 3.1 – The multi-threshold approach of the relaxation mechanism

- Eliminated correspondences (C^e set) are those whose confidence value (i.e. n_c) is less than the rejection threshold ($n_c < t_r$). These correspondences are automatically and definitely discarded by the alignment process;
- Rejected (or non-negotiable) correspondences (C^r set) are those whose degree of confidence is equal or greater than the rejection threshold and less than the negotiable threshold ($t_r \leq n_c < t_n$). These correspondences are also automatically discarded by the alignment process. However, contrary to the eliminated correspondences they may be presented and exploited during an interaction process with the user. Thus, unless the user changes explicitly its category they are not negotiable;
- Negotiable correspondences (C^n set) are those whose degree of confidence is equal or greater than the negotiable threshold and less than the proposition threshold ($t_n \leq n_c < t_p$). It means that the agent confidence on these correspondences is not enough to propose them to the opponent agent, but it is sufficiently to consider the possibility of revising (and relaxing) its confidence on those correspondences;

- Proposed correspondences (C^p set) are those whose degree of confidence is equal or greater than the proposition threshold and less than the mandatory threshold ($t_p \leq n_c < t_m$). It means that the agent is confident enough upon the correspondences so that it proposes them to the opponent agent;
- Mandatory correspondences (C^m set) are those whose degree of confidence is equal or greater than the mandatory threshold ($t_m \leq n_c$). It means that the agent is so confident of the pertinence and correctness of these correspondences, that these correspondences cannot be rejected by the opponent agent.

The second novelty concerns the ability of an agent to revise its perception about correspondences. For that, it was proposed the adoption of a meta-utility function (U) over the utility function (u). The purpose of the meta-utility function is to compute and provide an updated confidence value (n_c^u) that allows the re-categorization of correspondences from one category (e.g. negotiable) to another category (e.g. proposed or mandatory). Thus, it is responsible for (i) the identification of the parameters variation possibilities, (ii) the priorities over parameters variation and (iii) the conditions under which the variation may take place. Exploiting these elements, it might be necessary to iterate across different variations in order to find one variation that achieves the intended re-categorization. But, it might be the case that none of the possible iterations achieve the intended re-categorization. On the other hand, whenever a correspondence is re-categorized the agent makes an effort that must be measured. That effort is named convergence effort (e_c) and varies according to the priorities conditions and the parameter values of the meta-utility function. Therefore, the meta-utility function becomes also responsible for the evaluation of the convergence effort.

3.1.3 The Negotiation Process

The negotiation process exploits the relaxation and convergence mechanisms introduced above. It is composed by (i) a set of rules governing the outcome of the agents exchanged messages and (ii) five iterative phases as depicted in Figure 3.2.

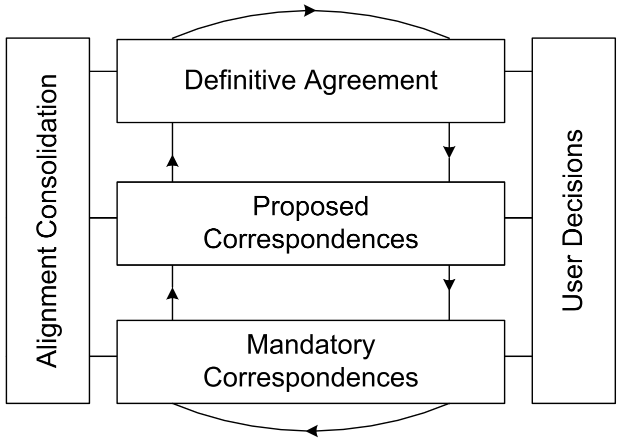

Figure 3.2 – Five phases of the Relaxation-based Negotiation Process

The set of rules governing the outcome of the agents exchanged messages is summarized in Table 3.2, where:

- "Failed" means that the correspondence is rejected and the overall negotiation fails;
- "Rejected" means that the correspondence is definitely rejected, but the negotiation process continues;
- "User" means that the decision about an agreement about the correspondence is forwarded to the user (if possible) when the negotiation between the agents finishes;
- "Accepted" means the correspondence is definitely accepted by both agents;
- "Negotiation" means that the correspondence is negotiated. In case of success the correspondence is conditionally accepted. If the correspondence is rejected by one of the agents, the correspondence is conditionally rejected.

Table 3.2 – Negotiation according to the category of correspondences suggested by two agents

Agent 1 \ Agent 2	Mandatory	Proposed	Negotiable	Non-Negotiable
Mandatory	accepted		accepted/failed	failed
Proposed			negotiation	rejected
Negotiable	accepted/failed	negotiation	user	n/a
Non-Negotiable	failed	rejected	n/a	n/a

The negotiation process starts in the Mandatory Correspondences Processing phase. Here each agent informs (and proposes) the opponent agent with the mandatory correspondences ($c \in C^m$) and, therefore, those must be immediately accepted. The set of accepted correspondences is denoted by C^a. At the end of this phase, the content of C^a is the union of the set of mandatory correspondences of both agents. Otherwise, the negotiation fails and invalidates any further negotiation. In case of success, the negotiation proceeds to the Proposed Correspondences Processing phase. In this phase, each agent proposes the correspondences categorized as proposed ($\forall c: c \in C^p$). As a result, for each proposed correspondence three situations may occur:

- The opponent agent also proposes the correspondence, in which case the correspondence is added to the accepted correspondences set ($C^a \leftarrow C^a \cup \{c\}$);
- The opponent agent categorized the correspondence as rejected, in which case the correspondence is rejected and no further discussion occurs about it;
- The opponent agent categorized the correspondence as negotiable ($c \in C^n$). In face of that, the opponent agent may iterate across its meta-utility function in order to re-categorize the correspondence as proposed. If the agent is not able to do such re-categorization then the correspondence is rejected. Otherwise, the correspondence is considered as tentatively accepted ($C^t \leftarrow C^t \cup \{c\}$).

In the Definitive Agreement phase, the tentatively accepted correspondences are subject to a definitive decision in order to ensure that the attempted agreement ($C^a \cup C^t$) is advantageous for both agents. Thus, this phase consists of deciding if the attempted agreement is globally advantageous (i.e. at the alignment granularity) and not only locally advantageous (i.e. at the correspondences granularity). This problem arises due to the convergence efforts made during the re-categorization made by one of the agents in the Proposed Correspondences Processing phase. Hence, for every re-categorized correspondence (c) that is included in the attempted agreement, a convergence effort has been evaluated by the meta-utility (e_c). The convergence efforts are considered undesirable to the agent and treated as a loss. Instead, the agreement upon the such correspondences provides some profit for the agent when it is re-categorized (p_c). In that sense, the balance between profits and losses is a function such that:

$$balance = \sum p_c - \sum e_c : c \in C^t$$

Depending on the balance value of each agent, the agents decide to agree upon the attempted agreement such that it becomes definitive or to propose a revision of the attempted agreement. At the end of this phase, the negotiation process ends successfully iif a definitive agreement was achieved. Otherwise, the negotiation process proceeds to the User Decisions phase or it ends without success.

The Alignment Consolidation phase and the User Decisions phase are orthogonal to the other three phases, which means that their inputs can occur at any stage of the negotiation process. The former phase is responsible for the correction and improvement of the alignment based on the existence of possible dependencies between correspondences considering the ultimate goal of the alignment (e.g. data transformation). The latter phase is optional and occurs in scenarios that do not require a fully automatic ontology matching negotiation. It consists of providing the user with the ability to drive the negotiation process by giving hints and taking decisions that influence the inputs and outputs of the other phases (e.g. to re-classify any existing correspondence).

3.1.4 Existing Limitations

From a theoretical point-of-view, the relaxation-based negotiation proposed in [Silva, Maio, and Rocha 2005; Maio et al. 2006] is relatively simple and easy to understand since it is based on the agents' ability to categorize and re-categorize correspondences and measure the profit and/or loss caused by the inclusion/exclusion of correspondences in the final agreement. However, from a practical perspective, this approach suffers from several issues, namely concerned with the specification of the required functions.

Contrary to the specification of the utility functions and its parameters which is straightforward based on the internal analysis of each matching algorithm, the specification of the meta-utility function is a very complex task requiring an enormous effort in terms of parameter identification, configuration and customization. During experiments [Gabriel et al. 2008] generic functions were adopted that increase the initial confidence value of correspondences based on a set of percentages (e.g. in the first iteration the confidence value is increased 5%, on the second 10% and so on). As a consequence of this difficulty, it is also very hard to measure properly the convergence effort spent in the re-categorization of a correspondence. During the experiments there were adopted linear and non-linear functions relating both the initial and the updated confidence value of the correspondence (e.g. $n_c^u - n_c$).

Similar issues and difficulties occur for the evaluation of the profit associated with tentatively accepted correspondences.

3.2 Argument-based Ontology Matching Negotiation

Regarding the argument-based approaches applied to the ontology matching domain it is convenient to distinguish between two kinds of work.

First, the work described in [Trojahn et al. 2008] and [Isaac et al. 2008] for ontology matching composition through argumentation (see section 2.2.4). This work is related to the setup phase of the matching process where different matching algorithms are selected and combined into a larger and more complex matcher. On both the VAF (cf. section 2.3.2) was adopted and extended, namely the notion of audience. In [Trojahn et al. 2008] the notion of audience was extended to include the concepts of certainty and uncertainty. In [Isaac et al. 2008] the notion of audience is extended to include the concept of strength of argument. This strength is directly given by matching algorithms which are themselves the audience values. Despite this work's proposal of argumentation for ontology matching, this thesis focuses on achieving an agreed alignment by resolving the matching conflicts between agents (i.e. applying argumentation for external practical reasoning), while in the described works the argumentation is used for internal theoretical reasoning to achieve a private ontology alignment.

Second, the Meaning-based Argumentation (MbA) [Laera et al. 2007], further improved into the flexible approach for determining agents' orientation on ontology mappings (FDO) [Doran et al. 2010] addressing the resolution of conflicts about correspondences between two agents wishing to interoperate. This work is described and analyzed next.

3.2.1 Meaning-based Argumentation

The Meaning-based Argumentation (MbA) [Laera et al. 2007] adopts the VAF (cf. section 2.3.2) as the underlying argumentation framework for ontology matching negotiation. The agents express their matching preferences based on the following classification of the matching algorithms:

- Terminological (T): are those that compare the names, labels and comments related to the ontologies' entities;
- Internal Structural (IS): are those that exploit the internal characteristics of entities such as the cardinality, transitivity, symmetry, domain and range of the properties;
- External Structural (ES), are those that exploit the (external) relations that an entity has with the other entities of the ontology such as super-entity, sub-entity or sibling;
- Semantic (S): are those that utilize theoretical models to determine whether there is a correspondence or not between two entities.
- Extensional (E): are those that compare the set of instances of entities being evaluated.

Each of these five categories is represented by an element on the set of values V of a VAF instance ($V = \{T, IS, ES, S, E\}$), such that each agent expresses its preferences (P) establishing a pre-order of V elements.

In MbA an argument $a \in A$ is a triple $a = (J, c, pos)$ where c is a correspondence, J is the grounds justifying the prima facie[8] belief that the correspondence does or does not hold, and pos is one of $\{+, -\}$ depending on whether the argument is for (i.e. c holds) or against c (i.e. c does not hold).

[8] *Prima facie* is a Latin expression whose literal meaning is "at first appearance" or "at first sight". In fields such as law and philosophy the expression is used to denote an evidence (or a proof) that something (e.g. a fact or a proposition) holds. However, such evidence might be rebutted further.

An agent generates arguments from a set of correspondences C based on (i) its preferences P, (ii) a function $T: C \rightarrow V | T(c) = v \in V$ and (iii) a threshold value such that $tr \in [0, 1]$. An agent sets the position pos of an argument $a \in A$ about a correspondence $c = (e, e', n, r)$ such that $c \in C$ as follows:

$$pos = \begin{cases} +, & if\ n \geq tr \wedge \max(P) = T(c) \\ -, & otherwise \end{cases}$$

In [Doran et al. 2010] the authors evolved the single threshold approach to a multi-threshold approach stated by a function $\phi: V \rightarrow [0, 1]$ such that $tr = \phi(T(c))$. As a result, an argument is in favor ($pos = +$) of correspondence c iif $n \geq \phi(T(c))$. As a consequence, the argument generating process becomes independent of the agents' preferences since the argument position is no more determined on the maximal preference ($max(P)$). The FDO will be considered from now on instead of MbA unless stated otherwise.

An argument $a_1 \in A$ attacks another argument $a_2 \in A$ ($(a_1, a_2) \in R_{att}$) iif a_1 and a_2 are arguments about the same correspondence c, but with contradictory pos.

Example 3.1 – The attack between arguments in the MbA/FDO approach

Consider three arguments such that: $a_1 = (J_1, c, +)$, $a_2 = (J_2, c, -)$ promote the terminological value while $a_3 = (J_3, c, +)$ promotes the external structure value. Thus, arguments a_1 and a_3 would attack and be attacked by argument a_2 as depicted in Figure 3.3.

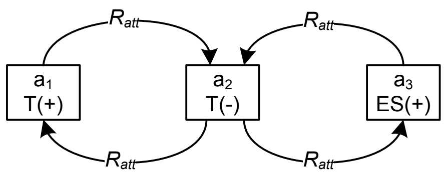

Figure 3.3 – The attack relationships between arguments in the MbA/FDO approach

The agents are thus able to generate and exchange arguments about correspondences and according to their preferences decide which arguments belong to their preferred extension. A correspondence is mutually accepted by a set of agents (typically 2 agents) if all agents have in their preferred extension an argument whose position is in favor of that correspondence. Because with the same preferences and the same set of arguments an agent can have several preferred extensions, a correspondence is said to be (i) skeptically agreed if it is accepted by all preferred extensions of all agents and is said to be (ii) credulously agreed if it is accepted in some preferred extensions by every agents.

Example 3.2 – Correspondences skeptically and credulous agreed

Consider the three arguments introduced in Example 3.1 and their relationships. An agent Ag_1 preferring the structural value (ES) to the terminological value (T) ($P_{Ag_1} = \{ES, T\}$) would have a single preferred extension such that $prefext_1^{Ag_1} = \{a_1, a_3\}$. Thus, it would skeptically agree on the correspondence c proposed by both arguments. On the other hand, an agent Ag_2 preferring the terminological value to the external structural value ($P_{Ag_2} = \{T, ES\}$) would have two possible preferred extensions such that $prefext_1^{Ag_2} = \{a_1, a_3\}$ and $prefext_2^{Ag_2} = \{a_2\}$. Thus, it would be only able to credulous agree on the correspondence c. Agent Ag_2 need to adopt/choose $prefext_1^{Ag_2}$ instead of $prefext_2^{Ag_2}$, otherwise agents would not agree about the correspondence in dispute.

3.2.2 Existing Limitations

Despite these argument-based approaches being simple and effective for adoption by agents, they suffer from several limitations:

1. Collection and Classification of Matchers and Correspondences. The agents generate arguments from correspondences provided by a single and common ontology matching repository (OMR), which stores the correspondences provided by several matching algorithms. Based on this, the argument generation process has two alternatives:
   - The OMR is responsible for classifying the matching algorithms and therefore the correspondences. This means that all the agents inherit the same perception of the correspondences, except the agent's preferences over the classification (P) and thresholds (ϕ). This alternative significantly constrains the agents' autonomy to interpret and exploit the correspondences. As a consequence, the argumentation outcome basically corresponds to the intersection of the alignments proposed by the agents;
   - The agents are capable of classifying the matching algorithms and therefore the correspondences stored in the OMR as terminological, structural, semantic and extensional only. If this is the case:
     - There is no justification for the use of a single OMR, which is a requirement in both MbA and FDO;

- It is necessary, at least, to perceive the impact (if any exists) in the argumentation process of multiple agents classifying differently the same pair matcher/correspondence. However, in both works nothing is said on that.

2. Private arguments. The MbA/FDO approach is focused in supporting argument exchange between agents. For that, MbA/FDO proposed the definition of a common list of argument types. I.e. it does not support the use of different (private) arguments by an agent, thus not supporting internal argument-based (i.e. theoretical or practical) reasoning.

3. Dependency between correspondences under negotiation. In the MbA/FDO approach, agents cannot capture and therefore do not take into consideration the (positive or negative) effect of accepting or rejecting a correspondence on the acceptability of other correspondences under discussion. However, in ontology matching (as in other domains), the acceptance/rejection of certain correspondences under negotiation depends on the acceptance/rejection of other correspondences under negotiation. Argumentation-based negotiation lacking this feature is incomplete.

4. Preferences. The agents are only able to express preferences about correspondences regarding a few general types of ontology mismatch. Furthermore, agents are not able to express preferences based on the combination of those types. It would be relevant to describe the following preference: a terminological argument is preferred to an external structural argument or to an internal structural argument, except if both internal and external arguments are present. This kind of preference cannot be expressed. This is in fact a limitation of VAF and not of the MbA/FDO *per se*.

5. Rebuttal, Undercut and Undermining arguments. The MbA/FDO attack relationships between arguments exploit the concept of rebuttal arguments only (cf. section 2.3.2). As a consequence, MbA/FDO does not allow agents to argue with each other about the premises of the arguments, but only to attack arguments.

Example 3.3 – The need to use undercutting argument in ontology matching negotiation

Imagine the case where an agent (Ag_1) has the intention to accept a correspondence based on the premise that the related entities are terminologically similar, but another agent (Ag_2) has the intention to reject the same correspondence based on the premise that the related entities do not have a similar terminology describing them. Considering that, in the MbA/FDO approach these two opposite positions are undefeatable because the agents are not able to argue about the reasons leading them to contradictory position. However, to persuade the opponent agent to change its position such a feature is required. In this case, for instance, agent Ag_1 could present an argument stating that the labels of the related entities are synonyms as a support for its position while agent Ag_2 could present another argument stating that the comments of the related entities are syntactically different. Further, agents could continue arguing for and against these arguments and so on.

Similar reasoning might be presented for undermining arguments.

However, considering that undercut and undermining arguments are valid and useful in many argumentation scenarios, restraining argumentation to rebuttal arguments severely constrains the agents' argumentation capabilities.

6. Symmetric attacks. As a consequence of the adopted argument instantiation process, formally an argument a only attacks and is attacked by the assertion of its negation $\neg a$. Yet, it was briefly and informally suggested that non-symmetric attacks might exist grounded on:

   - The notion of support between arguments, which is not captured by the original VAF neither by the MbA instantiation of VAF;
   - The formal semantics of the underlying ontology language (OWL), but its application in the scope of argument instantiation (and particularly defining the attack and support relationships) is not explicitly defined and ambiguously used in the MbA's illustrative example (cf. [Laera et al. 2007]);

   Consequently, the conditions under which an argument supports another through a support relation to a third argument are informal, ill-specified and ambiguous;

7. Multiple preferred extensions. The MbA/FDO does not provide any technique/approach to choose a preferred extension from a set of preferred extensions.

Considering the previous limitations and the adopted argumentation framework (VAF), it is impossible (at least very difficult)[9] to evolve these approaches to overcome the identified limitations. As a consequence, instead of improving the MbA/FDO, the adopted approach follows a different line of research.

[9] Except limitation 7.

SECOND PART

Chapter 4

THE ARGUMENT-BASED NEGOTIATION PROCESS

Despite the focus on the ontology matching negotiation problem, the negotiation process described in this chapter is general enough to be adopted in other application domains. As such the contributions presented in this chapter represent a general argument-based negotiation process first published in [Maio, Silva, and Cardoso 2011a].

Observations show that negotiation through argumentation between humans follow an iterative and incremental process where arguments and counter-arguments are successively presented, enabling humans to identify the existing conflicts and further present more arguments and counter-arguments to (tentatively) resolve such conflicts.

Concerning the arguments formulation, humans usually exploit a huge diversity of information sources which may provide information that is more or less reliable, (in)complete, (in)coherent, (in)consistent and so on. Thus, each human usually selects and exploits information provided by the sources that are considered more reliable and trustable for the problem in hands.

Concerning the arguments understanding and reasoning, each human has a unique (i.e. its own) perception and rationality over the domain of the problem in hands. Therefore, arguments are seen, interpreted and evaluated in light of that individual perception. This fact enables humans to extract from the same set of arguments several distinct and contradictory conclusions.

Typically, the argumentation process ends either (i) when no more conflicts exist or (ii) when no more arguments are presented by any of the participants. In the former case, the argumentation always ends successfully since no conflicts exist anymore. In the latter case, the argumentation may end successfully or unsuccessfully depending on the degree of importance that each one gives to the remaining conflicts when compared to the agreement in hand. Thus, if the parties agree that the agreement in hand is better than no agreement at all then the argumentation ends successfully, otherwise it ends unsuccessfully.

At least but not less important, with respect to human beings' natural ability to evolve their knowledge and perception of the world and particularly about the domain under which they are arguing. A classical situation occurs when a human faces an argument put forward by another human and (s)he does not know its meaning or how that argument relates and affects the others known arguments. In such cases, that human may require a conceptual description of that kind of argument in order to figure out the missing knowledge and therefore acquire it. The resulting knowledge acquisition contributes to the evolution of its perception of the domain under discussion.

4.1 The Argument-based Negotiation Process

This section describes the Argument-based Negotiation Process (ANP).

Within the ANP the negotiation entities (e.g. persons, organizations) are represented by agents. As with any other negotiation process, the proposed argument-based negotiation process happens, at least, between two agents. Furthermore, it is assumed that the negotiation occurs in the scope of a given community of agents. When joining a community, the agent is (implicitly or explicitly) accepting a set of rules by which all agents interactions are governed. One of the main rules is related to the key notion/concept of argumentation model, which in turn substantially constrains the characteristics of the argumentation process.

> **Definition 4.1 – Argumentation Model**
> An argumentation model (AM) is an artifact that captures (partially or totally) the perception and rationality that an agent has about a specific domain (e.g. ontology matching) regarding the argumentation process.

According to Definition 4.1, the argumentation model might conceptually define the vocabulary used to form arguments, the arguments' structure and even the way arguments affect (i.e. attack and support) each other. Hence, a model is a specification used for stating model commitments.

In practice, a model commitment is an agreement to use a vocabulary in a way that is consistent (but not necessarily complete) with respect to the theory specified by the model [Gruber 1993]. Agents commit to models which are designed so that the domain knowledge can be shared among these agents.

The community of agents on which the negotiation process occurs is responsible for defining a public argumentation model.

> **Definition 4.2 – Public Argumentation Model**
> A public argumentation model is a shared argumentation model capturing the minimal common understanding about argumentation over the domain problem being addressed (e.g. ontology matching) of a community of agents.

All agents of that community are able to understand the defined public argumentation model and reason on it. Further, each agent must be able to extend the public argumentation model so it better fits its own needs and knowledge. As a result, the agents freely specify their private argumentation model.

> **Definition 4.3 – Private Argumentation Model**
> A private argumentation model is an argumentation model capturing the understanding about argumentation over the domain problem being addressed (e.g. ontology matching) of a single agent.

While a public argumentation model represents a shared knowledge/perception between agents, a private argumentation model represents the individual perception/knowledge that an agent has.

Because the agents adopt their own private argumentation model, each agent has the responsibility for searching, identifying and selecting the sources of information that can provide the most relevant and significant information needed to instantiate its private model. After the private model instantiation each agent has a set of arguments that need to be evaluated in order to extract the agent consistent position, i.e. a preferred extension. A preferred extension includes two kinds of argument: intentional arguments and non-intentional arguments. The former ones define the intentions of the agent with respect to the agreement, while the latter ones represent the set of reasons supporting the intentions. Therefore, by exchanging the intentional arguments of their preferred extensions, agents are able to identify the existing conflicts. Such conflicts are further addressed by exchanging the non-intentional arguments.

Considering these premises, a general argument-based negotiation process was devised. Figure 4.1 depicts the phases of each agent's negotiation process, the flow of data and the interactions with other agents.

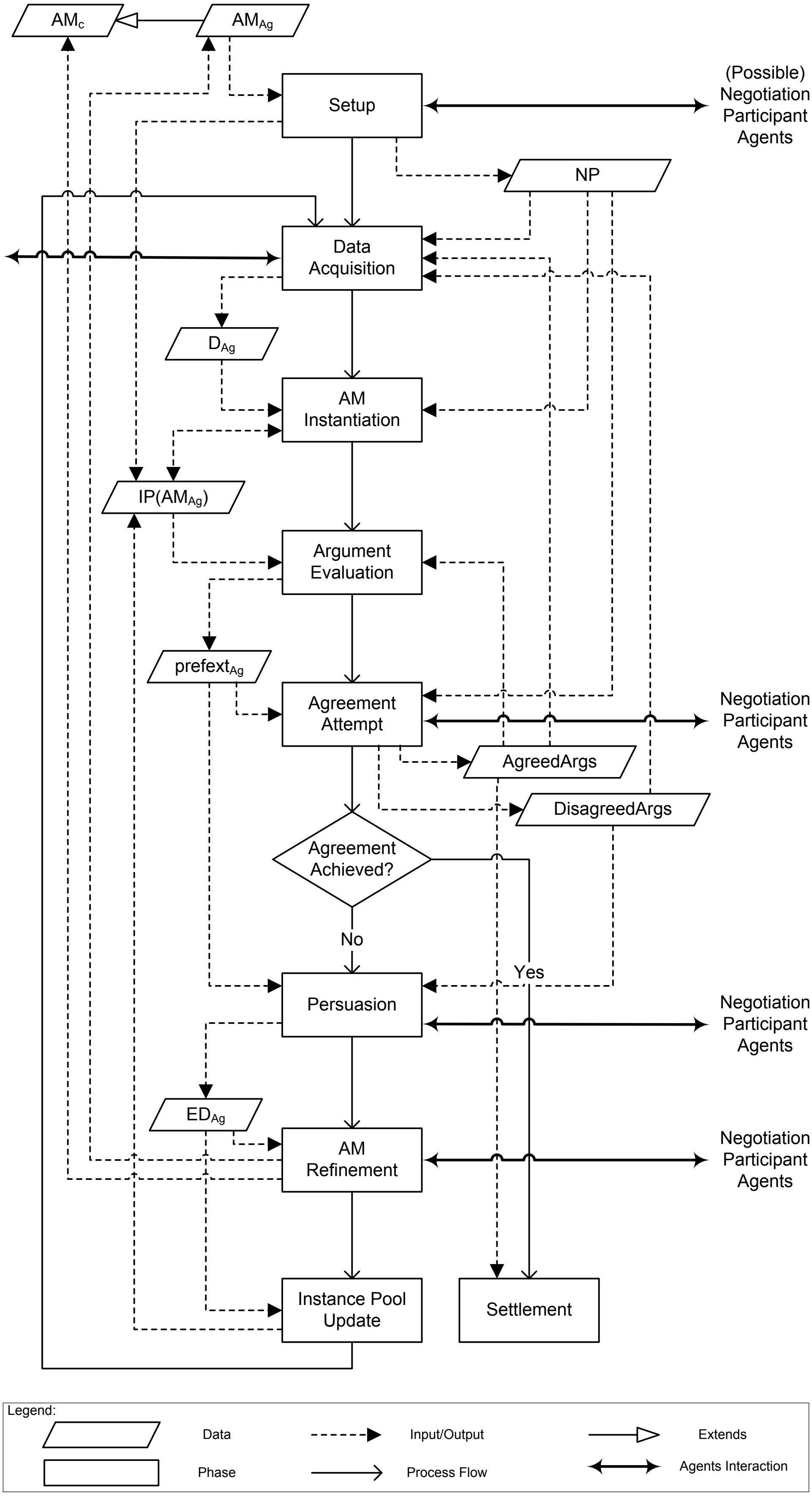

Figure 4.1 – The Argument-based Negotiation Process

The following sub-sections describe individually each phase of the negotiation process, recurring to the terminology and symbols used in the previous figure (e.g. AM_C).

4.1.1 Setup

The Setup phase defines the context of the negotiation. At the end of this phase all participating agents know and agree with this context. For that, the participating agents will engage in a set of interactions aiming for:

- The identification of the (possible) negotiation participants;
- The identification of the community's minimal common understanding, i.e. the public argumentation model AM_C between all participants;
- The definition of the required negotiation parameters/constraints such as deadline for achieving an agreement;
- The specification of the argument exchange method used by each agent;
- The specification of the negotiation method to compute a possible agreement between participants (e.g. by consensus between all participants or by the majority of participants' opinions);
- The establishment of special rights for some of the participants;
- The sharing of the data/information that is required by the agents in order to participate in the negotiation (e.g. the ontology used by each agent).

These interactions will result in the definition of a set of constraints called the negotiation parameters (NP).

Complementary to the negotiation parameters, each participant creates an instance-pool of its own argumentation model ($IP(AM_{Ag})$) that will capture the argumentation data.

In contrast to the other phases, this phase occurs only once.

4.1.2 Data Acquisition

During the Data Acquisition phase the agent collects the data/information that constitutes the grounds to generate the arguments (called D_{Ag}). For that, the agents may interact with other agents not directly participating in the negotiation process. It might be the case of specialized agents on the subject under discussion (e.g. agents specialized in ontology matching). The tentative agreements generated in the upcoming phases may be used as input information to the data-collecting mechanisms too.

For example, in the context of the ontology matching negotiation, the specialized agents may request correspondences as input seeds from third-party ontology matchers or retrieve correspondences from ontology matching repositories.

4.1.3 Argumentation Model Instantiation

The goal of the AM Instantiation phase is the instantiation of the agent's instance-pool of the argumentation model ($IP(AM_{Ag})$) based on the collected data (D_{Ag}). For that, the agent makes use of one or more data transformation processes over the collected data, generating a set of arguments structured according to the adopted argumentation model.

In the context of the ontology matching negotiation, the set of generated arguments correspond to (i) a set of correspondences and (ii) a set of reasons for and against those correspondences.

4.1.4 Argument Evaluation

In the Argument Evaluation phase, each agent extracts a preferred extension, i.e. a consistent position within $IP(AM_{Ag})$ which is defensible against any attack and cannot be further extended without introducing a conflict. According to the agent's $IP(AM_{Ag})$ one or more possible *preferred extensions* may be extracted.

If the argument evaluation process extracted more than one *preferred extension* then it is necessary to select one. The selection criterion has a special relevance during the negotiation process because it directly defines the agent's intentions and the reasons behind those intentions. Given that, instead of a simple criterion, a more elaborate selection criterion may be taken into consideration. For example, instead of the "selection of the preferred extension that is maximal with respect to set inclusion", one may consider "the preferred extension that minimizes the changes in respect to the previous one.

This phase occurs iteratively depending on the acquisition of new data/information and especially on the exchange of arguments between the agents during the persuasion phase. Because any change made to $IP(AM_{Ag})$ suggests that the agent's consistent position may change, a re-evaluation of the preferred extension is necessary.

4.1.5 Agreement Attempt

In the Agreement Attempt phase each participant makes an agreement proposal to the other agent(s) (called the candidate agreement). If accepted it will be settled by all participants.

This phase consists of two steps. In the first step, each agent makes its agreement proposal by exchanging the intentional arguments of its preferred extension only (called the intentional preferred extension). As a result of all proposals, two sets of arguments are derived and shared by all agents:

- The set of arguments agreed/proposed by all agents ($AgreedArgs$) which represents a candidate agreement;
- The set of arguments which at least one agent disagrees ($DisagreedArgs$). For a negotiation between n agents where $iprefext_{Ag_i}$ is the intentional preferred extension of agent i, these sets can be computed differently depending on the agents and according to the setup phase. One of the simplest agreement evaluation forms is based on their intersection (for n agents):

$$AgreedArgs = \bigcap_{i=1}^{n} iprefext_{Ag_i}$$

$$DisagreedArgs = \left(\bigcup_{i=1}^{n} iprefext_{Ag_i}\right) - AgreedArgs$$

In the second step, each participant evaluates its level of satisfaction with the current candidate agreement. For that, the agent considers the defined negotiation parameters/constraints (NP) and the content of the $AgreedArgs$ and $DisagreedArgs$ sets. According to the level of satisfaction, the participants must decide whether to:

- Continue the negotiation, and therefore proceed to the Persuasion phase, or
- Conclude the negotiation, which is either:
    - Successful if all agents accept the candidate agreement ($AgreedArgs$). In this case the process proceeds to the Settlement phase, or
    - Unsuccessful if the candidate agreement is not accepted by all agents and they do not continue the negotiation. The negotiation ends without an agreement.

4.1.6 Persuasion

In the previous phase a set of conflicts/disagreements have been identified (in the form of intentional arguments) that were not accepted by at least one participant ($DisagreedArgs$). In this phase each agent tries to persuade the others to accept its intentions. For that, each agent exchange arguments supporting its preferred extension and arguments attacking the other agents' preferred extension(s).

Each agent first selects from its preferred extension a (sub-) set of arguments supporting or attacking the arguments existing in $DisagreedArgs$. The selected arguments will be exchanged with the opponent agents to persuade them. There are two forms of exchanging the arguments:

1. The arguments are exchanged according to the AM_c and not according to AM_{Ag}, so the other agents can understand them. Thus, arguments represented according to the agent's argumentation model AM_{Ag} that cannot be expressed in terms of AM_c are not exchanged;
2. The arguments are exchanged according to the AM_{Ag} along with the parts of AM_{Ag} that allow the other agent to transform the arguments to AM_c. This arguments exchange method requires that agents have the ability to teach and to learn from other agents such that agents may evolve over time their perception/knowledge.

Independently of the exchanged method (decided in the Setup phase), at the end of this phase each agent has collected a new set of arguments presented by the other negotiating agents (ED_{Ag}). These arguments will be exploited in the Instance-Pool Update phase.

4.1.7 Argumentation Model Refinement

This phase concerns the refinement of the community's argumentation model (AM_c) according to the exchanged arguments and the agents' argumentation models (AM_{Ag}).

While it is not the aim of this description to present an evolution process of the argumentation model, nor the agents' reasoning process leading to such evolution, it is important to emphasize the need to evolve (over time) the community's argumentation model according to the agents' needs.

Due to the envisaged difficulty of the related tasks, this phase is seen as optional and, therefore may be skipped.

4.1.8 Instance Pool Update

In this phase, the agent analyzes, processes and possibly reclassifies the ED_{Ag} arguments in light of its AM_{Ag}. The ED_{Ag} arguments that are understood (in the light of AM_{Ag} or AM_C) and do not exist in $IP(AM_{Ag})$ are added while duplicated arguments are discarded. The added arguments are taken into consideration by the agent in the next round of proposals. The negotiation process proceeds to the Data Acquisition phase.

4.1.9 Settlement

The goal of the settlement phase is to transform the candidate agreement into a definitive agreement according to the settlement parameters of NP. In that respect, this phase is seen as an initiator of a set of transactions that occur after the agreed terms are known in order to fulfill the terms.

The set of transactions varies according to the domain of application and the negotiation object (e.g. goods/service or an ontology alignment) as well as the participating agent. For example, in the ontology matching negotiation scenario it may imply informing a given Message Translation Service to apply the agreed alignment when translating messages exchanged between the agents that participated in the negotiation. On the other hand, in an e-commerce scenario, fulfilling an agreement for selling physical goods may imply forwarding the agreement to the logistic and financial services.

4.2 Summary

The described negotiation process is a conceptual description of the argument-based negotiation occurring between parts. In that sense, the description does not specify/defend/provide any implementation details, such as:

- The agents' communication language to adopt;
- The exchanging messages, their structure and protocol;
- The algorithms to be adopted in each task/phase;
- The data sources to be exploited (including ontology matchers);
- The argumentation framework.

While all these dimensions are important, the abstraction proposed by the ANP allows:

- Identifying the core notion of argumentation model and its influence on the other dimensions;
- Defining nine distinct and iterative task-oriented phases.

As a consequence, the process is sufficiently generic to be adopted in a wide range of domains.

Considering the argument modeling formalisms described in section 2.3.2 and its expressiveness limitations concerning the direct adoption by argumentation-based systems, it becomes obvious that the notion of argumentation model introduced during the described argument-based negotiation process is not completely fulfilled, namely the incapability to explicitly and formally specify the agents' (shared) conceptualization of the arguments.

Thus, the remaining chapters focus on:

- Introducing/defining a new argumentation framework addressing these requirements;
- Showing how this new framework is exploited by the agents in ANP, and particularly in the context of ontology matching negotiation.

Chapter 5

THREE-LAYER ARGUMENTATION FRAMEWORK

The model layer of the abstract argumentation frameworks (cf. section 2.3.2) only comprehends a very restrictive vocabulary which is clearly insufficient to capture the knowledge of an agent with respect to a given domain. In light of such limitations and lack of an effective model layer satisfying the introduced notion of argumentation model, a new generic (but less abstract) argumentation framework was proposed in [Maio and Silva 2011b]: the Three-Layer Argumentation Framework (TLAF).

Its main purpose is to reduce the observed gaps between abstract argumentation frameworks and argumentation-systems. The proposed framework novelties are threefold:

- It adopts a general and intuitive argument structure;
- It includes a conceptual layer for the specification of the semantics of argumentation data applied in a specific domain of application (e.g. ontology matching negotiation, e-commerce) which meets the notion of argumentation model;
- It exploits the conceptual information and the defined argument structure to automatically derive the attack and support relationships between arguments.

Yet, because the information represented according to TLAF is easily transformed to BAF [Cayrol and Lagasquie-Schiex 2005a], the argument-based system can profit from the inherent suitability of abstract argumentation frameworks on the study of independent properties, namely in the scope of arguments' acceptability. Despite this, a novel method is proposed that takes advantage of the TLAF conceptual layer to evaluate the acceptability of arguments.

An informal overview of TLAF is provided next, further complemented with a formal definition.

5.1 Informal Overview

Unlike the abstract argumentation frameworks which comprehend only two layers, i.e. the (meta-) model layer and the instance layer, TLAF features three layers as depicted in Figure 5.1[10].

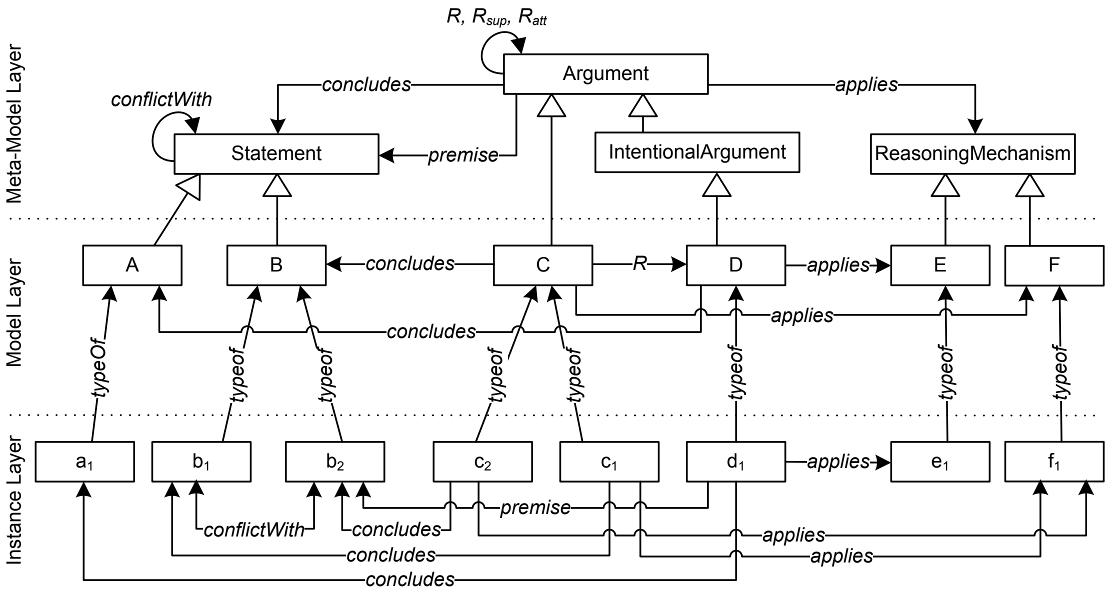

Figure 5.1 – The three TLAF modeling layers as captured by the respective OWL ontology

While the TLAF Meta-Model Layer and the TLAF Instance Layer roughly correspond to the (meta-) model layer and the instance layer of AF, BAF and VAF, the TLAF Model Layer does not have any correspondence in the surveyed abstract argumentation frameworks.

The TLAF Model Layer captures the notion of argumentation model, i.e. an explicit and formal specification of a shared and/or shareable conceptualization of arguments. Its content directly depends on:

- The domain of application to be captured;

[10] The line ending with a hollow triangle means specialization/generalization.

- The perception one (e.g. a community of agents or an individual agent) has about that domain.

Therefore, it provides vocabulary to define (i) argument-instances as instances of (ii) argument-types defined at the Model Layer. Similarly, it provides vocabulary to define (i) the relations between types of arguments, and (ii) the relationships between instances of arguments.

In TLAF, the meta-model layer defines an argument as being made of three parts:

- A set of premise-statements (or grounds),
- A conclusion-statement (or claim), and
- An inference from premises to the conclusion, enabled by a reasoning mechanism.

This argument structure is very intuitive and corresponds to the minimal definition presented by Walton in [Walton 2006]. For that, the meta-model layer defines the notion of Argument, Statement and Reasoning Mechanism, and a set of relations between these concepts.

Furthermore, following the notion of the BDI model [Bratman 1987; Wooldridge 2009], the concept of *Intentional Argument* was adopted, corresponding to the type of arguments whose content (the claim) corresponds to an intention. Domain data and its meaning are captured by the notion of *Statement*. This mandatorily includes the domain intentions, but also the desires and beliefs. The distinction between arguments and statements allows different interpretations of the same domain data (*Statement*), allowing its use in different context. The notion of *Reasoning Mechanism* captures the rules, methods, or processes applied by arguments.

At the model layer, an argument-type (or argument scheme) is characterized by the statement-type it concludes, the applied class of reasoning mechanism (e.g. Deductive, Inductive, Heuristic) and the set of affectation relations (R) it has. The R relation is a conceptual abstraction of the attack (R_{att}) and support (R_{sup}) relationships. The purpose of R is to define at the conceptual level that argument-instances of an argument-type may affect (either positively or negatively) instances of another argument-type. For example, according to the model layer of Figure 5.1, $(C, D) \in R$ means that the instances of argument-type C may attack or may support instances of argument-type D depending on the argument-instances' content. On the other hand, if $(X, Y) \notin R$ it means that instances of argument-type X cannot (in any circumstance) attack/support instances of argument-type Y. This explicit definition is not possible in any of the surveyed abstract argumentation frameworks, nor in MbA/FDO.

At the instance layer, an argument-instance applies a concrete reasoning mechanism to *conclude* a conclusion-statement-instance from a set of premise-statement-instances. All instances existing in the instance layer must have a type in the model layer and meet the type's modeling.

The relation *conflictWith* is established between two statement-instances only. A statement-instance b_1 is said to be in conflict with another statement-instance b_2 when b_1 states something that implies or suggests that b_2 is not true or does not hold. The *conflictWith* relation is asymmetric. Consequently, if a symmetric *conflictWith* is to be defined, two relationships must be established.

Example 5.1 – Symmetric *conflictWith* relationship between statements

In Figure 5.1 two *conflictWith* relationships between statement-instances are defined: one between b_1 and b_2 and another between b_2 and b_1 too. A possible scenario for this is, for instance, b_1 represents the statement "Peter is an expert on PCs." and b_2 represents the statement "Peter is not an expert on PCs".

5.2 Formal Definition

TLAF [Maio and Silva 2011b] is formally described as follows.

Definition 5.1 – TLAF

A TLAF structure is a singleton $TLAF = (E)$, where E is the set of entities pertaining to TLAF.

A TLAF represents a self-contained unit of structured information. Elements in a TLAF are called argumentation entities.

Definition 5.2 – TLAF Model Layer

A model layer associated with a TLAF is a 6-tuple $ML(TLAF) = (A, IA, S, M, R, \sigma)$ where:

- $A \subseteq E$ is a set of argument-types (or schemes);
- $IA \subseteq A$ is the sub-set of argument-types whose instances claim corresponds to an intention [Bratman 1987; Wooldridge 2009];
- $S \subseteq E$ is the a set of statement-types;
- $M \subseteq E$ is the set of reasoning mechanisms;
- $R \subseteq A \times A$ establishes a reflexive relation between two argument-types called arguments' affectation;
- σ is a function that assigns to every argument-type (i) the concluded statement-type and (ii) the reasoning mechanism applied, such as $\sigma: A \to S \times M$ where:
    - Function $concl: A \to S$;

- Function $reason: A \rightarrow M$.

Each TLAF has a model layer associated with it. The information captured within the model layer plays an important role by conducting and governing the instantiation process of the framework by an application, namely concerning the construction and the semantics of instances and the existing relationships between them. In that sense, the model layer is applied in the validation of the TLAF Instance Layer. Yet, and contrary to arguments, TLAF does not specify any structure for statements or reasoning mechanisms. The responsibility to specify such entities is left to the application level.

Within a TLAF model, the argument-types do not define the statement-types used as premises. Instead, these are derived from the R relation established between arguments. Thus, an argument-instance of type Y can only have as premises statements of type S iif S is concluded by an argument-type X and X affects Y ($(X, Y) \in R$). Considering the model layer of Figure 5.1, instances of argument-type D can only have as premises statements of type B because D is affected by argument-type C only.

Definition 5.3 – TLAF Instance Layer

An instance layer associated with a TLAF is a 6-tuple

$$IP(TLAF) = (I, instA, instS, instM, \Sigma, sconflict)$$

where:

- $I \subseteq E$, is a set of instances;
- Function $instA: A \rightarrow 2^I$ relates an argument-type with a set of instances. Consequently, the set of all argument instances AI is defined as follows:

$$AI = \bigcup_{\forall x: x \in A} instA(x)$$

  Furthermore, the inverse function is defined as $instA^{-}: AI \rightarrow A$;
- Function $instS: S \rightarrow 2^I$ relates a statement-type with a set of instances. Consequently, the set of all statement instances SI is defined as follows:

$$SI = \bigcup_{\forall x: x \in S} instS(x)$$

  The inverse function is defined as $instS^{-}: SI \rightarrow S$;
- Function $instM: M \rightarrow 2^I$ relates a reasoning mechanism with a set of instances. Consequently, the set of all reasoning mechanism instances MI is defined as follows:

$$MI = \bigcup_{\forall x: x \in M} instM(x)$$

The inverse function is defined as $instM^{-}: MI \rightarrow M$;

- Function $\Sigma: AI \rightarrow SI \times MI \times 2^{SI}$, defines for every argument-instance (i) the statement-instance concluded, (ii) the reasoning mechanism instance used to infer the conclusion and (iii) the set of statement-instances used as premises, where:
    - Function $iconcl: AI \rightarrow SI$, defines the statement-instance that plays the role of conclusion of an argument-instance. Indeed, an argument-instance has only one statement-instance as conclusion while a statement-instance is concluded by at least one argument-instance;
    - Function $ireason: AI \rightarrow MI$, defines the reasoning mechanism instance that is used by an argument-instance;
    - Function $ipremise: AI \rightarrow 2^{SI}$, defines the statement-instances used as premises on an argument-instance. Moreover, statement-instances used as premises are also concluded by other arguments;
- Function $sconflict: SI \rightarrow 2^{SI}$, defines the set of statement-instances that are in conflict with a given statement-instance. A statement-instance s_1 is said to be in conflict with another statement-instance s_2 when s_1 states something that implies or suggests that s_2 is not true or does not hold.

The instance layer corresponds to an instantiation of a given model layer for any particular argumentation process (i.e. interaction) between two or more agents. In that sense, this layer is concerned with the existing argument-instances (AI), statement-instances (SI), reasoning instances (MI) and their inter-relationships (Σ and $sconflict$).

Definition 5.4 – TLAF Interpretation

An interpretation of a TLAF is a structure $\mathfrak{I} = (\Delta^{\mathfrak{I}}, A^{\mathfrak{I}}, S^{\mathfrak{I}}, M^{\mathfrak{I}}, I^{\mathfrak{I}})$, where:

- $\Delta^{\mathfrak{I}}$ is the domain set;
- $A^{\mathfrak{I}}: A \rightarrow 2^{\Delta^{\mathfrak{I}}}$ is an argument interpretation function that maps each argument-type to a subset of the domain set;
- $S^{\mathfrak{I}}: S \rightarrow 2^{\Delta^{\mathfrak{I}}}$ is a statement interpretation function that maps each statement-type to a subset of the domain set;

- $M^{\mathfrak{I}}: M \rightarrow 2^{\Delta^{\mathfrak{I}}}$ is a reasoning mechanism interpretation function that maps each reasoning mechanism to a subset of the domain set;
- $I^{\mathfrak{I}}: I \rightarrow \Delta^{\mathfrak{I}}$ is an instance interpretation function that maps each instance to a single element in the domain set.

An interpretation is a model of TLAF if it satisfies the following properties:

- $\forall a, i: a \in A \wedge i \in instA(a) \Rightarrow I^{\mathfrak{I}}(i) \in A^{\mathfrak{I}}(a)$;
- $\forall s, i: s \in S \wedge i \in instS(s) \Rightarrow I^{\mathfrak{I}}(i) \in S^{\mathfrak{I}}(s)$;
- $\forall m, i: m \in M \wedge i \in instM(m) \Rightarrow I^{\mathfrak{I}}(i) \in M^{\mathfrak{I}}(m)$;
- $\forall a: a \in IA \Rightarrow A^{\mathfrak{I}}(a)$ are intentions;
- $\forall a, i: a \in A \wedge i \in instA(a) \Rightarrow I^{\mathfrak{I}}(iconcl(i)) \in S^{\mathfrak{I}}(concl(a)) \wedge$
  $I^{\mathfrak{I}}(ireason(i)) \in M^{\mathfrak{I}}(reason(a))$;
- $\forall a, i, p: a \in A \wedge i \in instA(a) \wedge p \in ipremise(i) \Rightarrow$
  $\exists x, y: I^{\mathfrak{I}}(y) \in A^{\mathfrak{I}}(x) \wedge p = iconcl(y) \wedge (x, a) \in R$;
- $\forall a, s: a \in A \wedge s \in S \Rightarrow A^{\mathfrak{I}}(a) \cap S^{\mathfrak{I}}(s) = \emptyset$;
- $\forall a, m: a \in A \wedge m \in M \Rightarrow A^{\mathfrak{I}}(a) \cap M^{\mathfrak{I}}(m) = \emptyset$;
- $\forall s, m: s \in S \wedge m \in M \Rightarrow S^{\mathfrak{I}}(s) \cap M^{\mathfrak{I}}(m) = \emptyset$.

Definition 5.5 – Argument Properties

An argument-type $a \in A$ and all its argument-instances ($\forall a_i: a_i \in AI \wedge a_i \in instA(a)$) are said to be:

- Intentional if $a \in IA$;
- Non-intentional if $a \notin IA$;
- Defeasible if $\exists x: x \in A \wedge x \neq a \wedge (x, a) \in R$;
- Indefeasible if $\forall x: x \in A \wedge x \neq a \wedge (x, a) \notin R$.

Arguments may be used for two purposes: (i) to represent and communicate intentions (i.e. intentional arguments) and (ii) to provide considerations (i.e. beliefs, desires) for and against those intentions (i.e. non-intentional arguments). Thus, an intentional argument may be affected by several non-intentional arguments. Additionally, to capture dependency between intentions, intentional arguments may be also affected (directly or indirectly) by other intentional arguments. A defeasible argument is affected by other (sub-) arguments (i.e. the ones concluding its premises or the ones undermining those premises) while an indefeasible argument can only be affected by its negation since it cannot have premises.

Example 5.2 – Intentional and non-intentional arguments

Regarding the ontology matching negotiation scenario, an intentional argument represents the intention to accept/reject a correspondence while a non-intentional argument provides a reason for or against a correspondence. Both (i.e. correspondences and reasons for/against correspondences) are captured in the instance layer by statement-instances.

5.3 Deriving Arguments Relationships

Despite the fact that the instantiation is a domain dependent process, it profits from the subjacent TLAF model, namely by exploiting the rules complementing the $iconcl$, $ipremise$ and $sconflict$, that have the ability to conduct and simplify the process.

According to the formal definitions previously introduced, the R_{att} and R_{sup} relationships between argument-instances pertaining to an $IP(TLAF)$ are not explicitly defined. Instead, these relationships are derived from two distinct kinds of information:

- Extensional information (existing at the instance layer):
    - The premises and conclusions of the argument-instances;
    - The conflicts between statement-instances, and;
- Conceptual information (existing at the model layer), namely the R relations defined between argument-types.

5.3.1 Deriving Support Relationships

In a TLAF instance-pool ($IP(TLAF)$) a support relationship between two argument-instances (say x and y) is automatically established (i.e. $(x, y) \in R_{sup}$) when the argument-type of x (say a) affects the argument-type of y (say b), i.e. $(a, b) \in R$, and either (i) the conclusion of x is a premise of y or (ii) both argument-instances have the same conclusion. The following rules (graphically depicted in Figure 5.2 and Figure 5.3) capture the conditions required to establish support relationships between argument-instances:

R1. $\forall a, b, x, y: a, b \in A \wedge (a, b) \in R \wedge x \in instA(a) \wedge y \in instA(b) \wedge x \neq y \wedge iconcl(x) \in ipremise(y) \Rightarrow (x, y) \in R_{sup}$ (Figure 5.2);

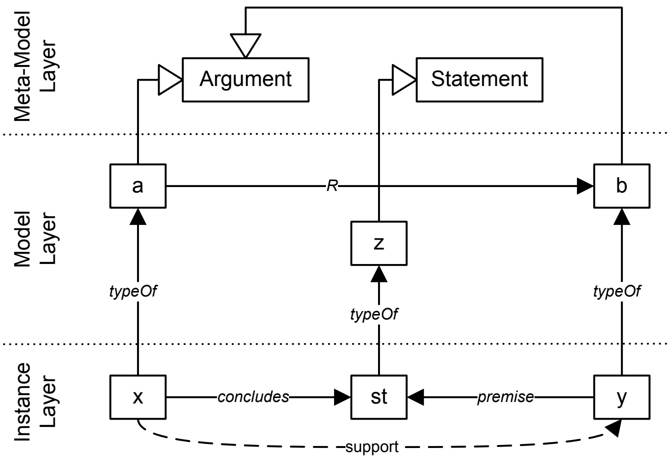

Figure 5.2 – Rule 1 to derive a support relationship between two argument-instances

R2. $\forall a, b, x, y: a, b \in A \wedge (a, b) \in R \wedge x \in instA(a) \wedge y \in instA(b) \wedge x \neq y \wedge$ $iconcl(x) = iconcl(y) \Rightarrow (x, y) \in R_{sup}$ (Figure 5.3).

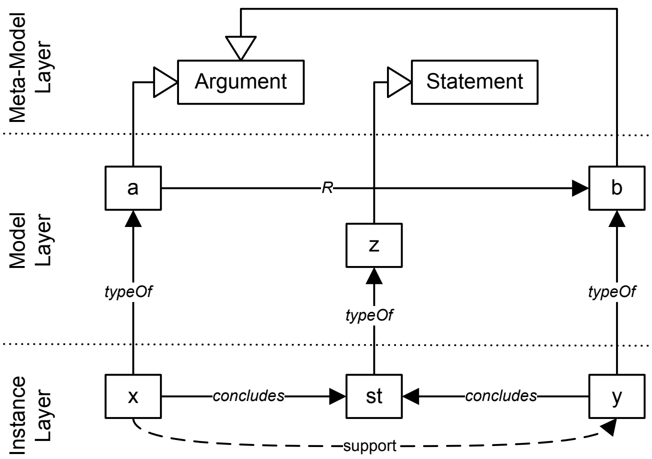

Figure 5.3 – Rule 2 to derive a support relationship between two argument-instances

Notice that two argument-instances might achieve the same conclusion starting from a different set of premises and/or reasoning mechanisms. In those circumstances, a support relation between argument-instances exists if there is a R relation between both (depicted in Figure 5.3). For a mutual support, two R relations are required: one from a to b (i.e. $(a, b) \in R$) and another one from b to a (i.e. $(b, a) \in R$).

5.3.2 Deriving Attack Relationships

In a TLAF instance-pool ($IP(TLAF)$) an attack relationship between two argument-instances (say x and y) is established (i.e. $(x, y) \in R_{att}$) when the argument-type of x (say a) affects the argument-type of y (say b), i.e. $(a, b) \in R$, and either (i) the conclusion of x is in conflict with any premise of y or (ii) the conclusion of x is in conflict with the conclusion of y. The following rules (graphically depicted in Figure 5.4 and Figure 5.5) capture the conditions required to establish attack relationships between argument-instances:

R3. $\forall a, b, x, y, s: a, b \in A \wedge (a, b) \in R \wedge x \in instA(a) \wedge y \in instA(b) \wedge x \neq y \wedge s \in ipremise(y) \wedge s \in sconflict(iconcl(x)) \Rightarrow (x, y) \in R_{att}$ (Figure 5.4);

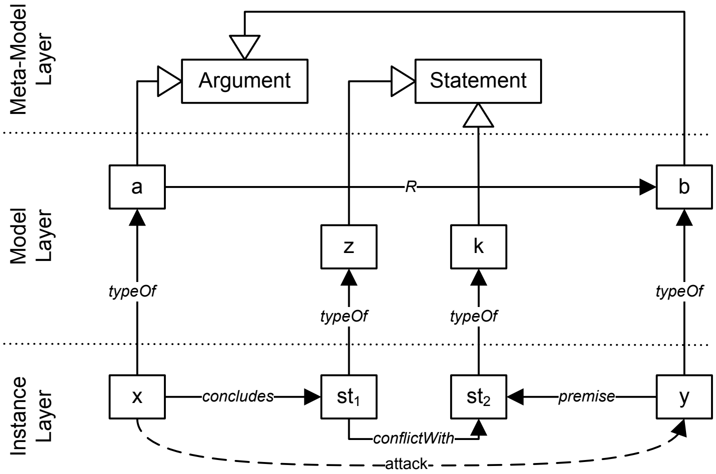

Figure 5.4 – Rule 1 to derive an attack relationship between two argument-instances

R4. $\forall a, b, x, y: a, b \in A \wedge (a, b) \in R \wedge x \in instA(a) \wedge y \in instA(b) \wedge x \neq y \wedge iconcl(y) \in sconflict\big(iconcl(x)\big) \Rightarrow (x, y) \in R_{att}$ (Figure 5.5).

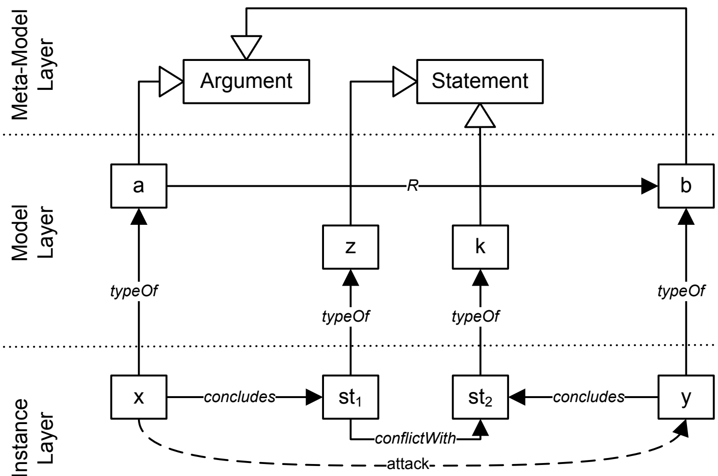

Figure 5.5 – Rule 2 to derive an attack relationship between two argument-instances

According to the rule/scenario depicted in Figure 5.5, one cannot say that argument y also attacks argument x because the conflict relation between statements is asymmetric. However, that would happen iif statement st_2 is also in conflict with statement st_1 ($st_1 \in sconflict(st_2)$) and a R relation between b and a ($(b, a) \in R$) exists too.

5.3.3 Exploiting Attack Relationships Rules

While the main purpose of the two last rules (R3 and R4) is to derive the attack relationships between argument-instances, their embedded knowledge might be used to constrain the scope of conflict relationships between statements, and therefore reduce the search/combination space of a process aiming to identify conflicts between statements.

A conflict relationship between two statement-instances (say st_1 and st_2) can be established only if their statement-types (say z and k respectively) satisfy at least one of the following conditions:

- There is an argument-type (say a) concluding z that affects any other argument-type (say b), i.e. $(a, b) \in R$, where statement-instances of type k can be used as premises of argument-instances of type b (from R3);
- There is an argument-type (say a) concluding z that affects any other argument-type (say b), i.e. $(a, b) \in R$, where k is concluded by b (from R4).

Both conditions can be verified using the information captured at the model layer only. On the other hand, if a conflict relationship is established between two statement-instances and none of these conditions apply then it has no impact on derived attack relationships between arguments.

5.4 Argument Evaluation

In any argumentation system, one of the most important processes is to determine the acceptability of argument-instances, i.e. to state which argument-instances hold (are undefeated) and which argument-instances do not hold (are defeated). Since this process usually consists of evaluating (the strength of) each argument-instance based on the relationships it has with other argument-instances (i.e. support and attack), from now on this process is referred to as the argument evaluation process.

TLAF does not impose any particular argument evaluation process, i.e. argumentation systems are free to apply the argument evaluation process that is thought to be the most suitable for the case in hand. In that sense, because a TLAF instance-pool can be easily represented in a more abstract formalism such as BAF[11] and as AF[12], argumentation systems may opt to apply abstract argumentation semantics such as the ones described in [Baroni and Giacomin 2009] to select the conclusions of the associated sets of argument-instances. Yet, because TLAF assumes that

[11] the set of argument-instances (AI) and the derived argument-instances relationships (R_{sup} and R_{att}) of the TLAF correspond to the BAF elements A, R_{sup} and R_{att} respectively.

[12] TLAF instance-pool is represented as an AF by first representing TLAF as a BAF and further representing the resulting BAF as an AF. The process to represent a BAF as AF is described in [Cayrol and Lagasquie-Schiex 2010].

bipolarity is important for the application domain, argumentation systems may also opt to apply an argument evaluation process that exploits the bipolarity such as the ones proposed in [Cayrol and Lagasquie-Schiex 2005b; Amgoud et al. 2008; Karacapilidis and Papadias 2001; Verheij 2002].

However, any of these processes are able to:

- Deal with the cyclic relationships that may exist between argument-instances;
- Take advantage of the TLAF Model Layer.

In order to exploit TLAF features and also to overcome these limitations a TLAF argument evaluation process was devised comprehending two complementary sub-processes:

- Argument Evaluation Process, in which the agent determines the strength of each argument-instance based on the strength value of the argument-instances supporting and attacking the argument-instance being evaluated;
- Preferred Extension Selection Process, in which a set of argument-instances are selected based on its strength constituting the preferred extension of the agent. This corresponds to the agent's proposal to the opponents.

These two sub-processes are described next.

5.4.1 Argument Evaluation Process

The proposed argument evaluation process grounds on the idea that different argument-types may demand different forms of evaluation (see Example 5.3).

> **Example 5.3 – Argument-types demanding different ways of evaluation**
>
> Evaluating type 1: An argument-type applying a deductive reasoning method may be evaluated by a function that returns a value stating that an argument-instance holds iif the argument-instance being evaluated is not attacked by any other argument-instance, otherwise the function returns a value stating that the argument-instance does not hold.
>
> Evaluating type 2: On the other hand, an argument-type applying a voting reasoning method may be evaluated by a function that considers the difference between the number of argument-instances attacking and supporting the argument-instance being evaluated to state if the argument-instance holds or not.

Accordingly, the argument evaluation process was defined as follows.

Definition 5.6 –TLAF' Argument Evaluation

An argument evaluation structure associated with a TLAF is a 6-tuple

$$Eval(TLAF) = (F, mapF, mapV, alg)$$

where:

- $F \subseteq E$, is a set of domain dependent argument evaluation functions, where each function $f \in F$ represents a distinct way to evaluate arguments;
- Function $mapF: A \to F$ assigns to each argument type the evaluation function that will be used to evaluate all its argument-instances. I.e. the strength of an argument $a \in AI$ is evaluated by a function $f \in F$, such that $f = mapF\big(instA^{-}(a)\big)$. To ensure coherence to the evaluation process all instances of the same type must be evaluated by the same function;
- $mapV$ is a matrix of the argument-instances' strength values, where each column represents an argument-instance and each line represents the strength of every argument-instance in a given iteration. Therefore, $mapV_i$ denotes the values of all argument-instances in the i^{th} iteration, $mapV_i^a$ denotes the strength of an argument-instance a in the i^{th} iteration. In particular, $mapV_0^a$ denotes the initial strength of the argument-instance a and $mapV^a$ denotes the strength of an argument a in the last iteration (i.e. row) of the matrix;
- alg is an algorithm that iterates over the argument evaluation functions (F) to (re)evaluate the argument-instances strength until a defined criteria is reached.

Distinct argument evaluation functions may exploit differently the relationships between argument-instances and the strength/information of those argument-instances. Despite those differences, it is necessary that the values returned by all functions follow a common semantic understood by alg. Thus, for the sake of simplicity, consider an argument evaluation function defined as follows.

Definition 5.7 – Argument Evaluation Function

A function $f \in F$ is defined as $f: (AI, mapV_i) \to \mathcal{V}$, where $\mathcal{V}$ is an ordered set $\{\mathcal{V}_{min}, \dots, \mathcal{V}_{\overline{m}}, \dots, \mathcal{V}_{max}\}$ with at least three possible values, such that:

- $\mathcal{V}_{min}$ represents the minimal strength value,
- $\mathcal{V}_{max}$ represents the maximal strength value, and
- $\mathcal{V}_{\overline{m}}$ represents a value whose distance to $\mathcal{V}_{min}$ and $\mathcal{V}_{max}$ is the same.

Example 5.4 – Values for a Gradual-based Argument Strength Semantics

For a gradual valuation of the arguments' strength, one can define $\mathcal{V} = [-1, 1]$ such that $\mathcal{V}_{min} = -1$, $\mathcal{V}_{max} = 1$ and $\mathcal{V}_{\overline{m}} = 0$.

The semantics of the argument strength given by a function is defined next.

Definition 5.8 – Argument Strength Semantics

The strength value of an argument $a \in AI$ in the iteration i evaluated by $f \in F$, such that $mapV_i^a = f(a, mapV_{i-1})$, has the following semantics:

- $mapV_i^a > \mathcal{V}_{\overline{m}}$, means that the argument a holds and therefore it is undefeated. In addition, if $mapV_i^a > mapV_i^b > \mathcal{V}_{\overline{m}}$ it means that the confidence on considering argument a undefeated is bigger than the confidence on considering argument b undefeated;
- $mapV_i^a < \mathcal{V}_{\overline{m}}$, means that the argument a does not holds and therefore it is defeated. In addition, if $\mathcal{V}_{\overline{m}} > mapV_i^a > mapV_i^b$ it means that the confidence on considering argument b defeated is bigger than the confidence on considering argument a defeated;
- $mapV_i^a = \mathcal{V}_{\overline{m}}$, means that the argument a has an undefined status, i.e. it might be considered either as defeated or as undefeated. This means that the positive force given by the supports relationships and the negative force given by the attacks relationships are equivalent.

5.4.2 Preferred Extension Selection

The result of the execution of the algorithm alg is therefore the $mapV$ matrix, populated with the arguments strength values evaluated by the evaluation functions given by $mapF$.

This matrix is further applied as input information in the preferred extension selection. To describe this selection process, first consider the following sets and functions:

- AI is the set of all argument-instances (as defined previously);
- $IAI \subseteq AI$ is the set of intentional argument-instances, i.e. $\forall a: a \in AI \wedge instA^{-}(a) \in IA \Rightarrow a \in IAI$;
- $Y \subseteq AI$ is the set of non-intentional argument-instances, i.e. $Y = AI - IAI$;
- T is an empty set of argument-instances. It will contain all argument-instances of the preferred extension whose type is an intentional argument;

- T' is another empty set of arguments-instances. It will contain all argument-instances of the preferred extension that endorses the intentional argument-instances of the preferred extension. The endorsement evaluation is described below based on the following functions;
- Function $sup: AI \rightarrow 2^Y$ identifies the set of argument-instances that directly support an argument-instance, excluding the intentional argument-instances, such that:

$$sup(a) = \{x : x \in Y \wedge (x,a) \in R_{sup} \wedge mapV^x > \mathcal{V}_{\overline{m}}\}$$

- Function $att: AI \rightarrow 2^Y$ identifies the set of argument-instances that directly attack an argument-instance excluding the intentional argument-instances, such that:

$$att(a) = \{x : x \in Y \wedge (x,a) \in R_{att} \wedge mapV^x > \mathcal{V}_{\overline{m}}\}$$

- Function $sup^*: AI \rightarrow 2^Y$ identifies the set of argument-instances that directly or indirectly support an argument-instance, such that:

$$sup^*(a) = sup(a) \cup \bigcup_{\forall x: x \in Y \wedge (x,a) \in R_{sup}} sup^*(x) \cup \bigcup_{\forall x: x \in Y \wedge (x,a) \in R_{att}} att^*(x)$$

- Function $att^*: AI \rightarrow 2^Y$ identifies the set of argument-instances that directly or indirectly attack an argument-instance, such that:

$$att^*(a) = att(a) \cup \bigcup_{\forall x: x \in Y \wedge (x,a) \in R_{att}} sup^*(x) \cup \bigcup_{\forall x: x \in Y \wedge (x,a) \in R_{sup}} att^*(x)$$

Given this, the selection process is as follows. For each argument-instance $a \in IAI$:

1. If the defeat status of a is undefeated then a is added to T and all argument-instances supporting it are added to T', i.e. if $mapV^a > \mathcal{V}_{\overline{m}} \Rightarrow T = T \cup \{a\} \wedge T' = T' \cup sup^*(a)$;
2. If the defeat status of a is defeated then a is not added to T but all argument-instances attacking it are added to T', i.e. if $mapV^a < \mathcal{V}_{\overline{m}} \Rightarrow T' = T' \cup att^*(a)$;
3. If the defeat status of a is undefined ($mapV^a = \mathcal{V}_{\overline{m}}$) it means that multiple preferred extensions exist, resulting in the execution of one of the above steps alternatively.

The preferred extension is obtained by the union of T and T' ($prefext = T \cup T'$), such that the set T corresponds to the intentional preferred extension ($iprefext$) while the set T' corresponds to the belief preferred extension ($bprefext$). Thus, a TLAF preferred extension is composed by the undefeated intentional arguments and all the non-intentional arguments that support (directly or indirectly) the undefeated intentional arguments. Again, notice that the undefined status of argument-instances gives rise to multiple preferred extensions.

Given a preferred extension ($prefext$), the intentions and beliefs of an agent correspond to the statement-instances concluded by the argument-instances of the preferred extension, such that:

- $intentions = \{iconcl(a): a \in iprefext\}$;
- $beliefs = \{iconcl(a): a \in bprefext\}$.

5.5 Related Work

The TLAF conceptual model layer and the consequent adoption of a structured argumentation are exploited to reduce the existing gap between the abstract argumentation frameworks and its adoption by applications, namely concerning the instantiation process.

Regarding the conceptual model only, the most similar work existing in literature is the Description Logic formalizations proposed in [Rahwan and Banihashemi 2008; Rahwan et al. 2011] for the Argument Interchange Format (AIF) [Chesñevar et al. 2006]. In common to the AIF-based work, TLAF has mainly two aspects:

- The adopted argument structure suggested by Walton [Walton 2006]; and
- The TLAF model layer representation in an OWL ontology [Maio and Silva 2011a].

Despite both works adopting the same argument structure they diverge on their purpose and consequently on the modeling approach:

- While the main purpose of the AIF-based work is to take advantage of the reasoning capabilities of OWL to automatically classify argument types (or argument schemes) and argument instances, the TLAF purpose is to promote the advantages respecting the adoption of the model layer as conceptual representation of the structure and semantics of arguments used and how they affect each other in the scope of a domain and/or community;
- The modeling approach taken by both works diverge on several issues too. TLAF explicitly distinguishes between argument-types and the reasoning mechanisms, while in the AIF-based work the reasoning mechanisms are implicit in the name of the argument-scheme;
- They also diverge in the way premises of argument-types are defined. In the AIF-based work each argument-type defines explicitly the set of statement-types it has as premises. On the contrary, in the TLAF the set of admissible statement-types that an argument-type has as premises is inferred through the R-relations established between argument-types. Therefore, TLAF adopts a more argument-oriented modeling;

- Similarly to the Carneades framework [Gordon, Prakken, and Walton 2007], in TLAF an argument has zero or more statements as premise. On the contrary, in AIF-based work an argument has at least one statement as premise;
- Another difference between AIF-based work and TLAF concerns poly-classification of argument instances. In the AIF-based work an argument-instance can be classified into several types (one or more) while in TLAF an argument-instance is single classified (the most specific/representative one of that instance). While the poly-classification of argument-instances is useful for several tasks (e.g. querying of arguments), it raises acceptability problems that are not completely understood yet, thus leaving the respective semantics to the application/agent [Rahwan and Banihashemi 2008].

In the general abstract framework for rule-based argumentation proposed by Prakken [Prakken 2010] arguments apply either a *strict* or a *defeasible* rule over a set of axioms (i.e. premises) to conclude another axiom, such that axioms are defined in a logical language. In TLAF, these two kinds of rules may correspond to two kinds of reasoning mechanisms and the concrete rules may correspond to instances of those reasoning mechanisms. The three types of attack relationship between argument-instances described in [Prakken 2010]: (i) rebutting, (ii) undercutting and (iii) undermining (cf. section 2.3.2) are captured by the TLAF rules to derive such relationship.

Prakken's work also describes arguments as trees of inference rules such that an argument contains other sub-arguments concluding intermediate conclusions and so on. TLAF captures such trees of arguments at the model layer (through the R-relation) and also at the instance layer such that the root of the trees are intentional arguments. Instead, the Carneades framework [Gordon, Prakken, and Walton 2007] captures arguments in graph-shaped structures, where it is assumed that argument graphs contain no cycles. Similarly, TLAF allows graph-shaped structures of arguments, but instead its argument graphs may contain cycles since no restriction exists at the model layer level.

5.6 Summary

Considering the need (i) to have an argumentation framework that comprehends and fulfills the notion of argumentation model and (ii) to reduce the gap between applications and abstract argumentation frameworks, the TLAF has been proposed and described in this chapter. At the end, TLAF is seen as a less abstract argumentation framework that can be adopted by argumentation-systems/applications with the following advantages:

- It comprehends a model layer whose purpose is to capture the knowledge (structure and semantics) applied in a specific domain of application in order to be shared and reused, i.e. it provides the constructs to develop explicit and formal specification of a conceptualization of a domain of argumentation;

- It constrains and conducts the modeling process of the argumentation specific scenario. Though for the same scenario very different modeling approaches are possible;
- Despite the fact that the argument-instances generation process is fully domain dependent, it profits from the established TLAF model in several aspects, such that:
    - It constrains the scope in which it is valuable to establish a conflict relationship between statements ($sconflict$) and, therefore, simplifies the automation of the process that discovers or instantiates the $sconflict$ relation, by reducing and driving the search/combination space between statements;
    - It constrains the type of conclusion and premises, and the reasoning mechanism associated with an argument-instance and, therefore, simplifies the automation of the process that establishes the premises and conclusion relationships of arguments with statements;
- States a clear and minimal set of rules defining the circumstances on which attack and support relationships are established between argument-instances. These relationships are then automatically established;
- Enables the use of more abstract formalism such as AF and BAF as an abstract level of the overall system to benefit from the AF and BAF capabilities already identified in the literature, namely those concerned with arguments acceptability and the study of independent properties;
- Ensures that the applications adopt an argument structure that (i) is widely consensual in the argumentation community and (ii) promotes the interoperability and compatibility between different agents/systems since argument-instances can be easily represented in a common format such as the AIF [Chesñevar et al. 2006].

Argumentation systems adopting TLAF are not committed to any particular argument evaluation process. Despite that, TLAF suggests that:

- Argument-instances are evaluated based on a set of functions related to its type;
- Through an algorithm that is able to deal with circularity and mutual dependency between argument-instances.

The next chapter describes how the ANP and TLAF can be used in the ontology matching negotiation problem.

Chapter 6

THE ARGUMENT-BASED ONTOLOGY MATCHING NEGOTIATION APPROACH

This chapter describes the application of the argument-based negotiation process and the TLAF, described in Chapter 4 and Chapter 5 respectively, regarding the ontology matching negotiation problem as follows:

- First, the overall approach is described (section 6.1);
- Second, a TLAF Model Layer is presented which has two purposes: (i) to facilitate the comparison with the state-of-the-art approaches and (ii) to provide a concrete example applied in the remainder of the document (section 6.2).
- Third, a detailed description of the proposed approach is provided for the arguments instantiation phase (section 6.3);
- Fourth, a detailed description of the proposed approach is provided for the argument evaluation phase (section 6.4).

Descriptions provided in this chapter were previously published in [Maio, Silva, and Cardoso 2011b; Maio and Silva 2010].

6.1 The Approach

In the context of ontology matching negotiation, it is assumed that the negotiation occurs between two distinct agents (say Ag_1 and Ag_2) using ontologies (O_1 and O_2 respectively) to represent their knowledge about an arbitrary domain.

The goal of the agents engaging in the ontology matching negotiation process is to agree about an alignment between those ontologies, satisfying and ensuring confidence for the business interaction process.

Figure 6.1 graphically depicts an overview of the proposed argument-based negotiation approach for the ontology matching domain where TLAF is employed as a specification formalism of the public and private argumentation models.

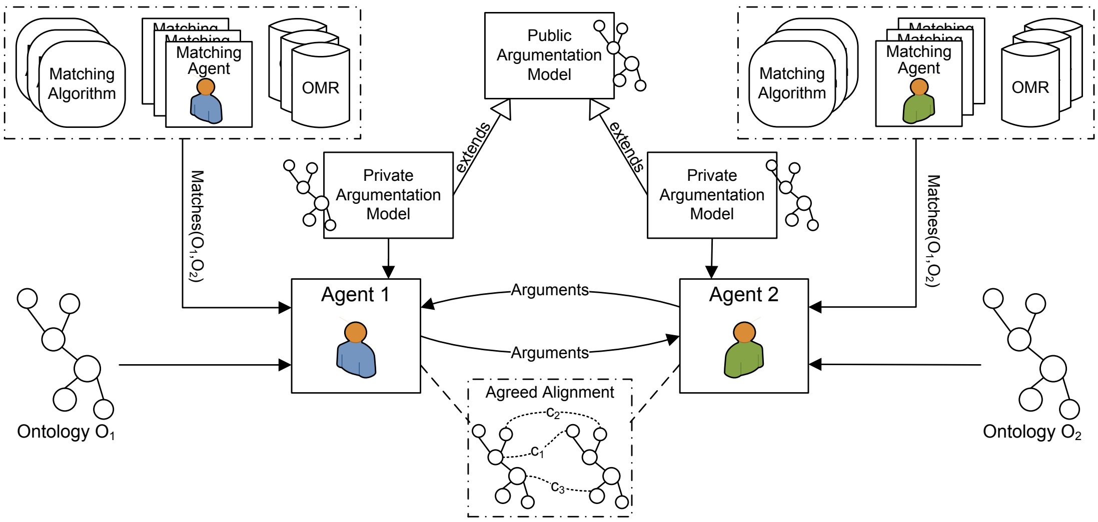

Figure 6.1 – Overview of the proposed argument-based ontology matching negotiation approach

The negotiation occurs in the scope of a given community of agents which is able to define a public argumentation model. This model captures the community minimal common understanding about argumentation over ontology matching that all agents of the community are able to understand. Yet, each agent is free to extend the public argumentation model to better fit its own needs and knowledge.

It is assumed that each agent follows, internally and externally, the nine phases of the argument-based negotiation process described in Chapter 4. The following sub-sections describe the proposed approach for the ontology matching negotiation domain.

6.1.1 Setup Phase

Bearing in mind the argument-based negotiation process, it is during the Setup Phase that:

- Each agent informs the opponent agent of the subject ontology (i.e. the agent's ontology to align);
- *A priori* alignment properties can be established between the agents (e.g. the alignment level and cardinality);
- The agents identify and accept the public argumentation model (AM_C) provided by the community as the minimal common understanding between them. As a consequence, the agent's private argumentation model is the same as AM_C or extends it ($AM_C \sqsubseteq AM_{Ag}$).
- Others possible negotiation parameters are defined between the agents.

6.1.2 Data Acquisition Phase

Since the agents adopt their own private argumentation model (AM_{Ag}), each agent has the responsibility to search, to identify and to select the sources that can provide the most relevant and significant information needed to instantiate its private model properly. Consequently, during the Data Acquisition Phase, it is foreseen that agents may collect information simultaneously from a set of ontology matching repositories (OMRs), a set of agents providing matching services or even through a set of internal matching algorithms, as depicted in Figure 6.1. The expected result of all these information sources is a set of correspondences between both ontologies. Yet, it is foreseen that the correspondences that are temporarily agreed (but not settled as definitive) in the upcoming phases may be used to feed the data-collecting mechanisms too. This is especially relevant for the agents wishing to apply matching algorithms (e.g. semantic algorithms) in which the receiving correspondences play the role of anchors or inductive facts.

6.1.3 Argumentation Model Instantiation Phase

The collected data (D_{Ag}) is further interpreted by agents (in light of its argumentation model) to generate argument-instances. A proposal of this process is presented in section 6.3.

6.1.4 Argumentation Evaluation Phase

The argument-instances generated are evaluated by the agent in order to extract a preferred extension. In this context, a preferred extension enables the agent (i) to figure out the desired alignment and (ii) to negotiate the desired alignment with the opponent agent. It defines the correspondences that an agent wants to include in the alignment (through the intentional arguments) and a set of reasons (through the non-intentional arguments) supporting those correspondences. A proposal of this process is presented in section 6.4.

6.1.5 Agreement Attempt Phase

In the Agreement Attempt Phase, agents exchange the intentional arguments of their preferred extensions to perceive their convergences (i.e. correspondences proposed/accepted by both agents) and their divergences (i.e. correspondences proposed/accepted by a single agent).

The set of correspondences proposed/accepted by both agents represents an alignment about which agents must opt between:

- Settle the alignment as definitive and, therefore, proceed to the Settlement Phase;
- Discard the alignment (definitively or temporarily) and, therefore, proceed to the Persuasion Phase.

6.1.6 Persuasion Phase

In order to persuade its opponent to accept or to give up the disagreed correspondences, each agent exchanges the set of non-intentional arguments existing on its preferred extension supporting its position, and therefore attacking the other agents' divergent positions.

6.1.7 Argumentation Model Refinement Phase

Grounded on the arguments exchanged during the Persuasion Phase, agents may evolve/refine their argumentation model in the Argumentation Model Refinement Phase. While such a task is optional, it requires the agents' ability to learn from agent interactions and from the other agent knowledge. This task is out of scope in this thesis.

6.1.8 Instance Pool Update Phase

In the Instance Pool Update Phase, each agent analyses the arguments received during the Persuasion Phase in a way that adds new arguments and/or updates existing arguments. As a result of this update, the previous preferred extension becomes invalid and is discarded. At this point, an iteration of the argumentation process is concluded.

The process proceeds to the Data Acquisition Phase where each agent may acquire new information based on the outputs of the previous iteration. Such information is taken into consideration within the others phases of the process.

The process has as many iterations as needed to reach an agreed alignment or, instead, until no more (new) arguments are generated by agents. Yet, it might be the case that a maximum number of iterations is previously defined in the Setup Phase. On the two latter cases, the negotiation may end without an agreement, and therefore unsuccessfully.

6.1.9 Settlement Phase

In the Settlement phase, the set of tasks to carry out are dependent on the business interaction process that had previously taken the agents to the ontology matching negotiation process. In that sense, each agent makes use of the agreed alignment to develop the business interaction process.

6.1.10 Summary

The nine phases of the argument-based negotiation process can be grouped in two distinct categories: (i) those requiring the execution of some agent's internal process (e.g. Argument Model Instantiation, Argument Evaluation) and (ii) those requiring interaction between agents. This latter category can be split in to two other sub-categories: (i) the one requiring interaction between the opponent negotiation agent(s) (e.g. Setup, Agreement Attempt, Persuasion) and (ii) those where the agent interacts with other agents not participating in the negotiation (e.g. Data Acquisition).

The phases requiring agent interaction are mostly concerned with:

- The adopted agent communication language (e.g. FIPA-ACL [FIPA00061 2002; FIPA00037 2002; Fensel and FIPA 2002], KQML [Finin, Labrou, and Mayfield 1997] or KIF [Genesereth and Fikes 1992]) and consequent protocol;
- The representation format of arguments (e.g. AIF [Chesñevar et al. 2006], TLAF [Maio and Silva 2011b]);
- The representation format of correspondences (e.g. Alignment API [David et al. 2011], EDOAL [Scharffe 2011] or SBO [Maedche et al. 2002]);

These are not addressed further in the document.

The remaining of this section is focused on the automation of the following phases:

- Argument Model Instantiation Phase (see section 6.3);
- Argumentation Evaluation Phase (see section 6.4).

Before that however, the next section introduces the TLAF Model that will support the walk-through example of the remaining sections.

6.2 A TLAF Model for Ontology Matching Negotiation

This section presents an argumentation model for the ontology matching negotiation domain based on the adoption of TLAF as the underlying argumentation framework. In order to facilitate the reader comprehension and comparison with the state-of-the-art approaches, the presented argumentation model is inspired by the MbA/FDO examples and is initially kept as simple as possible. However, to demonstrate how the limitations previously identified in literature are overcome, the presented argumentation model is further evolved.

The argumentation model (graphically and partially depicted in Figure 6.2) for the ontology matching negotiation domain considers that:

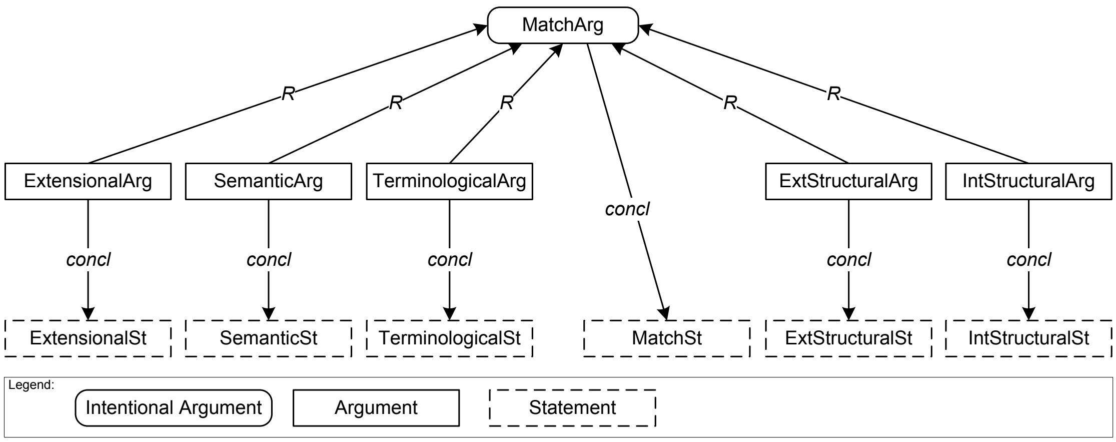

Figure 6.2 – Partial representation of a TLAF model for the ontology matching negotiation

- The intentional arguments (rounded rectangle) are those concluding that a given correspondence must (or must not) be included into the final alignment, i.e. into the final agreement;
- The classification of matching algorithms is that used by the MbA [Laera et al. 2007] approach: i.e. Terminological, External Structural, Internal Structural, Extensional, and Semantic (see section 3.2.1). A correspondence $c = (e, e', r, n)$ is interpreted by the agent as follows:
  - If the algorithm is classified as Terminological then the relation r holds between the set of lexical elements (e.g. labels, comments) associated with entities e and e';
  - If the algorithm is classified as Internal Structural then the relation r holds between the internal structure (e.g. the range of attributes) associated with entities e and e';

  - If the algorithm is classified as External Structural then the relation r holds between the external structure (e.g. the set of relations each entity has with other entities) associated with entities e and e';
  - If the algorithm is classified as Semantic then the relation r holds between the semantics associated with entities e and e' given by the ontology/models;
  - If the algorithm is classified as Extensional then the relation r holds between the known set of instances of entities e and e'.
- Each correspondence generated by an algorithm classified in one of those categories is seen as a sufficient reason in favor or not favor the intention of including the correspondence into the alignment.

For each of the previous interpretations a statement type (e.g. *TerminologicalSt*) and an argument type (e.g. *TerminologicalArg*) are defined, such that the argument type concludes the respective statement type. Similarly, to capture the intention of accepting/rejecting a correspondence it is defined the intentional argument *MatchArg* that concludes the statement *MatchSt*. This argument type is affected (i.e. attacked/supported) by all arguments derived from the interpretations (e.g. $(TerminologicalArg, MatchArg) \in R$).

Additionally, considering the types of matching algorithms two distinct reasoning mechanisms are defined: *Heuristic* and *Deductive* (this is not depicted in the Figure 6.2 for the sake of clarity). Except the *SemanticArg* argument that applies the *Deductive* reasoning mechanism, all the others arguments apply the *Heuristic* reasoning mechanism.

Thus, the agent's possible interpretations over the matching algorithms and their suggested correspondences represent the units of information that can be used to argue over the correspondences.

Unlike arguments, TLAF does not establish any structure for the statements and reasoning methods. These are the responsibility of the domain/application. In the context of the ontology matching negotiation domain, such structures have been defined as follows.

Definition 6.1 – Structure of a Statement-Instance

A statement-instance is a 3-tuple $s = (G, c, pos)$ where c is a correspondence, G is a univocal matcher identification and $pos \in \{+, -\}$ states the position of G about c, i.e. states if G is for $(+)$ or against $(-)$ c.

Definition 6.2 – Structure of a Reasoning Method Instance

An instance of a reasoning method is a tuple $rm = (\Gamma, desc)$ where Γ is a univocal identification of the algorithm used by the matcher and $desc$ is a textual description of Γ.

For the sake of simplicity and in order to be able to distinguish between different matchers using the same base algorithm Γ but with different configuration-parameters, G is the univocal identification of the algorithm Γ instance.

6.3 Argument Instantiation Process

In the argument instantiation phase an agent interprets, processes and transforms into argument-instances the data collected during the Data Collection phase. In the context of ontology matching, the collected data is about ontology correspondences captured according to the following structure.

Definition 6.3 – Structure of Collected Data

Being D_{Ag} the set of data/information collected by an agent, $d \in D_{Ag}$ is a pair (G, c) where c is a correspondence and G is the univocal identification of the matcher/agent from where c was collected.

An example of a set of data collected by an agent regarding the existence of an equivalence relationship between e_1 and e_1' is provided in Table 6.1.

Table 6.1 – An example of a set of data collected by an agent

Generated by Matcher	Correspondence	
	ID	Content
G_1	c_1	$(e_1, e_1', =, 0.95)$
G_2	c_2	$(e_1, e_1', =, 0.90)$
G_3	c_3	$(e_1, e_1', =, 0.50)$
G_4	c_4	$(e_1, e_1', =, 0.85)$
G_5	c_5	$(e_1, e_1', =, 1.00)$

Given the structure of the collected data, it is required that an agent has two distinct mechanisms:

- A mechanism that enables the agent to determine if a matcher is in favor or against a given correspondence (i.e. the matcher position), addressed in section 6.3.1;
- A mechanism that allows the agent to transform the correspondences into an argumentation data structure, in the light of its private argumentation model. This includes the following steps:
    - Interpretation of correspondences (section 6.3.2);

- Creation of argument-instances (section 6.3.3);
- Classification of the argument-instances (see section 6.3.4);
- Identification and definition of the argument-instances' premises (section 6.3.5);
- Identification and definition of conflicts between statement-instances (section 6.3.6).

6.3.1 Determining the Matcher's Position

An agent determines the position of a matcher about a correspondence based on the degree of confidence the matcher has on that correspondence (Definition 6.4).

Definition 6.4 – Matcher's Position

The position pos of a matcher G with respect to a correspondence $c = (e, e', r, n)$ is:

- In favor of c if its confidence value on c is equal or greater than a given threshold value ($n \geq tr_{+}$);
- Against c if its confidence value on c is less than another threshold value ($n < tr_{-}$);
- Neither in favor nor against c if $tr_{-} \leq n < tr_{+}$ and therefore c may be ignored.

When a matcher does not propose any correspondence about a given pair of ontology entities, an agent has two mutually exclusive alternatives:

- Consider that the matcher has no opinion about that and therefore is neither in favor nor against any correspondence relating those entities;
- Consider that the matcher is against any correspondence relating those entities.

Typically, instead of a single value of tr_{+} and tr_{-} common to all matchers, one value of tr_{+} and tr_{-} is settled on by the matcher. The appropriate threshold values can be set by the user or suggested by automatic mechanisms (e.g. [Maio et al. 2007]).

6.3.2 Correspondences Interpretation

In order to transform the correspondences into arguments, an interpretation function representing the knowledge that an agent has about the ontology matching domain is applied. In practice, this function maps the type of correspondence to the agent's private argumentation model (Definition 6.5).

Definition 6.5 – Interpretation Function

Let $AM_{Ag} = (A, IA, S, M, R, \sigma)$ be the private TLAF model of an agent. An interpretation function is defined as $\psi: G \times c \rightarrow S \times M \times pos$ where G is a univocal identification of the generator of correspondence c, and S and M are a statement type and a reasoning mechanism of AM_{Ag} respectively, and pos is the value resulting from the interpretation of the matcher's position.

Table 6.2 presents an example of the parameters to determine the matcher's position and the correspondence interpretation in respect to the TLAF model described in section 6.2.

Table 6.2 – An example of an interpretation function

Matcher	Correspondence Content			Statement Type	Reasoning Mechanism	tr_+	tr_-
	e	e'	r				
G_1	any			MatchSt	Heuristic	0.90	0.70
G_2	any			TerminologicalSt	Heuristic	0.85	0.80
G_3	any			ExtStructuralSt	Heuristic	0.95	0.75
G_4	any			IntStructuralSt	Heuristic	0.90	0.80
G_5	any			SemanticSt	Semantic	1.00	1.00

Notice that this interpretation function depends only on the matcher type (because correspondence content is "any"). However, the interpretation function may also depend on a categorization of the correspondences' content. For example, based on the type of the related entities (e.g. concept or property) or based on the established relation (e.g. equivalence, narrow).

It is worth noticing that:

- Because each agent has its own interpretation function, it is possible that two agents using the same matcher have distinct interpretations; and
- Despite an agent being able to understand all the arguments defined in the TLAF model, it might be able to generate only a sub-set of them.

Based on the two distinct mechanisms described above, a TLAF model for the ontology matching negotiation domain is instantiated following four distinct and complementary steps, which are described next.

6.3.3 Creating Argument-Instances

Through the interpretation function ψ, every collected datum ($d \in D$) gives rise to:

- A statement instance s of type S whose content is (G, c, pos) according to the defined matching statement structure;
- A reasoning mechanism instance m of type M representing the algorithm used by G;

- An argument instance a concluding s and applying m.

Algorithm 6.1 captures this process.

Algorithm 6.1. Generating TLAF instances.

Require: The TLAF model to instantiate, an interpretation function ψ and a set of collected data D

Ensure: a set of arguments, statements and reasoning methods

for all $d \in D$ **do**
    $(G, c) = d$
    $(S, M, pos) = \psi(G, c)$
    StatementInstance $s = createStatementInst(G, c, pos, S)$
    ReasoningInstance $m = createReasoningMethodInst(G, M)$
    ArgumentInstance $a = createArgumentInst(s, m)$
end for

Considering:

- The TLAF model presented in section 6.2;
- The set of data described in Table 6.1;
- The interpretation function described in Table 6.2,

the Algorithm 6.1 will populate a TLAF instance with the information depicted in Table 6.3, Table 6.4 and Table 6.5.

Table 6.3 – The resulting statement-instances after Algorithm 6.1

Statement-Instances		
ID	**Type**	**Content**
s_1	MatchSt	$(G_1, c_1, +)$
s_2	TerminologicalSt	$(G_2, c_2, +)$
s_3	ExtStructuralSt	$(G_3, c_3, -)$
s_5	SemanticSt	$(G_5, c_5, +)$

Table 6.4 – The resulting reasoning mechanisms instances after Algorithm 6.1

Reasoning Mechanisms		
ID	**Type**	**Content**
m_1	Heuristic	G_1
m_2	Heuristic	G_2
m_3	Heuristic	G_3
m_5	Deductive	G_5

Table 6.5 – The resulting argument-instances after Algorithm 6.1

Argument-Instances				
ID	**Type**	**Premises**	**Concludes**	**Applies**
a_1	Argument	$\emptyset$	s_1	m_1
a_2	Argument	$\emptyset$	s_2	m_2
a_3	Argument	$\emptyset$	s_3	m_3
a_5	Argument	$\emptyset$	s_5	m_5

Notice that, the correspondence c_4 did not give rise to any instance because its degree of confidence is between tr_- and tr_+ associated with the matcher G_4 ($0.80 \leq 0.85 < 0.90$). Therefore, this correspondence was ignored since one cannot determine the position of G_4 about c_4 (cf. Definition 6.4).

The current state of the arguments instantiation process is depicted in Figure 6.3.

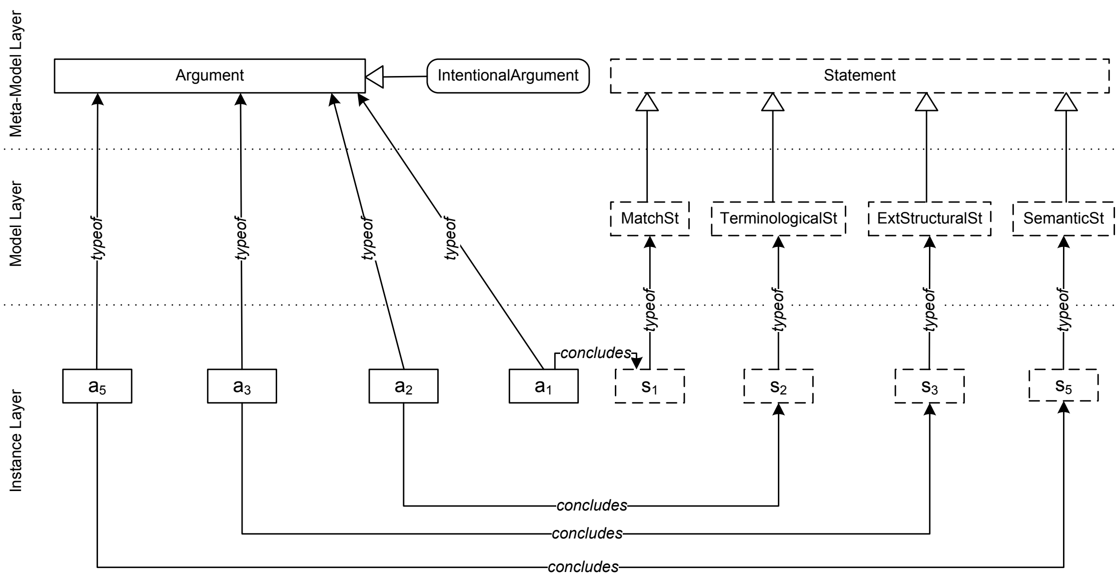

Figure 6.3 – The created instances and their relations after the creation instances step

6.3.4 Classifying Argument-Instances

As the reader may notice, this previous step did not associate an argument type to the argument-instances. This is a responsibility of this step.

For that, it is suggested to apply an automatic rule-based process, whose specification exploits the information captured in the TLAF model.

Because each argument type of the TLAF model is related to a unique pair formed by the type of the statement concluded and the applied reasoning mechanism, one can make use of a rule-based process whose rules capture the following sentence: set argument type at to any argument-instance whose conclusion-statement is of type st and the reasoning mechanism applied is of type mt. Each argument type in the TLAF model will have a corresponding rule. Notice that in simple cases these rules can be automatically generated considering only the content of the TLAF model.

The rule-based process for the TLAF model described in section 6.2 would have six distinct rules (one for each argument type). The rules for argument types *MatchArg* and *TerminologicalArg* are:

- $\forall a, s, m: a \in AI \wedge s = iconcl(a) \wedge s \in instS(MatchSt) \wedge m = ireason(a) \wedge m \in instM(Heuristic) \rightarrow a \in instA(MatchArg)$;
- $\forall a, s, m: a \in AI \wedge s = iconcl(a) \wedge s \in instS(TerminologicalSt) \wedge m = ireason(a) \wedge m \in instM(Heuristic) \rightarrow a \in instA(TerminologicalArg)$.

As a result, argument-instances described in Table 6.5 would be reclassified as depicted in Table 6.6.

Table 6.6 – The classified argument-instances

Argument-Instances				
ID	**Type**	**Premises**	**Concludes**	**Applies**
a_1	MatchArg	$\emptyset$	s_1	m_1
a_2	TerminologicalArg	$\emptyset$	s_2	m_2
a_3	ExtStructuralArg	$\emptyset$	s_3	m_3
a_5	SemanticArg	$\emptyset$	s_5	m_5

The classification process changes the state of the argument's instantiation process as depicted in Figure 6.4.

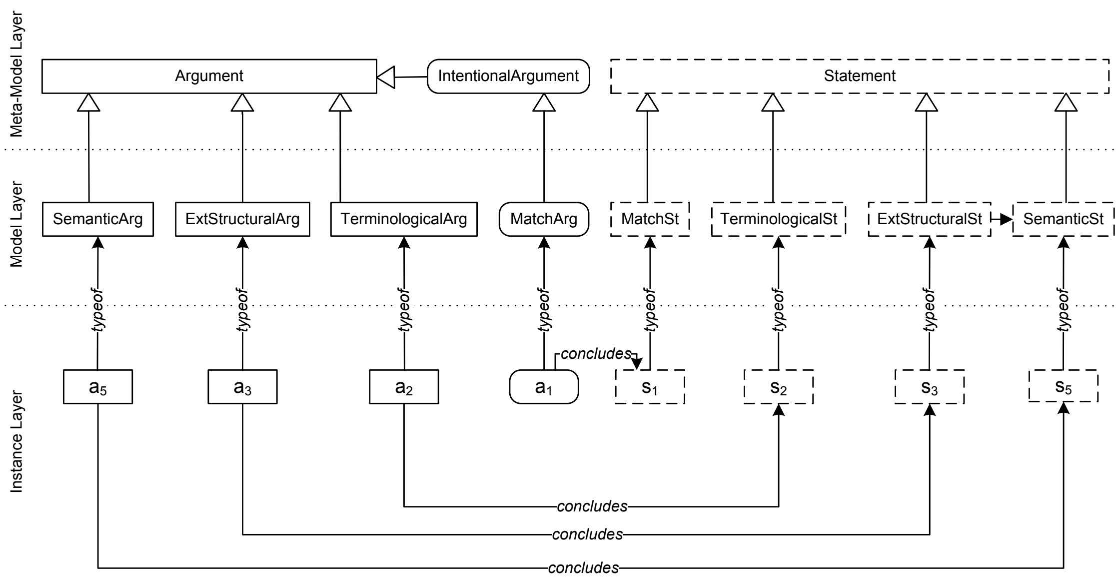

Figure 6.4 – The instantiation process after classifying the argument-instances

6.3.5 Identifying Premises of Argument-Instances

As the reader may have noticed, argument-instances were initially instantiated with no statement-instances as premises. Because only a few ontology matching algorithms are able to justify the suggested correspondences (e.g. [Shvaiko et al. 2005]), it is not possible to rely on the matchers' generated data. To overcome this limitation, the TLAF model will be exploited.

Premises of argument-instances are identified by exploring the R-relations between argument types defined in the TLAF model.

Definition 6.6 – Argument-Instances' Premises

A statement-instance s_a is a premise of an argument-instance b of type y and concluding a statement-instance s_b iif s_a is concluded by an argument a of type x which in turn is related with y through R (i.e. $(x,y) \in R$) and both statement-instances (s_a and s_b) have the same position ($pos_a = pos_b$) and satisfy a given condition (cf. bellow $condition(s_a, s_b)$).

Algorithm 6.2 captures this rule.

Algorithm 6.2. Setting premises of argument-instances.

Require: A TLAF model instantiated with argument and statement instances

Ensure: Establishes the premises of the argument-instances

for all $(x,y) \in R$ **do**
    for all $a \in instA(x)$ **do**
        $(G_a, c_a, pos_a) = s_a = iconcl(a)$
        for all $b \in instA(y)$ **do**
            $(G_b, c_b, pos_b) = s_b = iconcl(b)$
            if $(pos_a = pos_b)$ *and* $condition(s_a, s_b)$
                $ipremise(b) = ipremise(b) \cup \{s_a\}$ *//Adding* s_a *as premise of* b
            end if
        end for
    end for
end for

The aim of the $condition$ function is to check the existence of domain dependent relations between two statement-instances (s_a and s_b). This function is formally defined as follows.

Definition 6.7 – Condition Function

Let SI be the set of statement-instances of a TLAF instance. A condition function is defined as $condition: SI \times SI \rightarrow b \in \{true, false\}$.

Given the defined structure of ontology matching statements, the most evident condition to be checked is if both statement-instances are about the same correspondence (i.e. $c_a = c_b$). Thus, being $s_a = (G_a, (e_a, e'_a, r_a, n_a), +)$ and $s_b = (G_b, (e_b, e'_b, r_b, n_b), +)$, such a condition would be defined as:

$$cnd_1(s_a, s_b) = (e_a \equiv e_b \wedge e'_a \equiv e'_b \wedge r_a \equiv r_b)$$

However, it is envisaged that depending on the type of the input statement-instances, the condition function may check the existence of multiple relations. For an undefined number of conditions (say N) the condition function is defined as:

$$condition(s_a, s_b) = \begin{cases} true: & \bigvee_{i=1}^{N} cnd_i(s_a, s_b) \\ false, & otherwise \end{cases}$$

Considering the information depicted in Table 6.3 and Table 6.6 (depicted in Figure 6.4), only the argument-instance a_1 would be set with premises (see Table 6.7 and Figure 6.5). Thus, one can interpret that a_1 concludes s_1 (and therefore c_1) because (i) the related entities of c_1 (e_1 and e_1') are described by similar terminologies (defined in s_2) and (ii) both entities are considered semantically equal (defined in s_5).

Table 6.7 – The argument-instances premises after third step

Argument-Instances				
ID	**Type**	**Premises**	**Concludes**	**Applies**
a_1	MatchArg	$\{s_2, s_5\}$	s_1	m_1
a_2	TerminologicalArg	$\emptyset$	s_2	m_2
a_3	ExtStructuralArg	$\emptyset$	s_3	m_3
a_5	SemanticArg	$\emptyset$	s_5	m_5

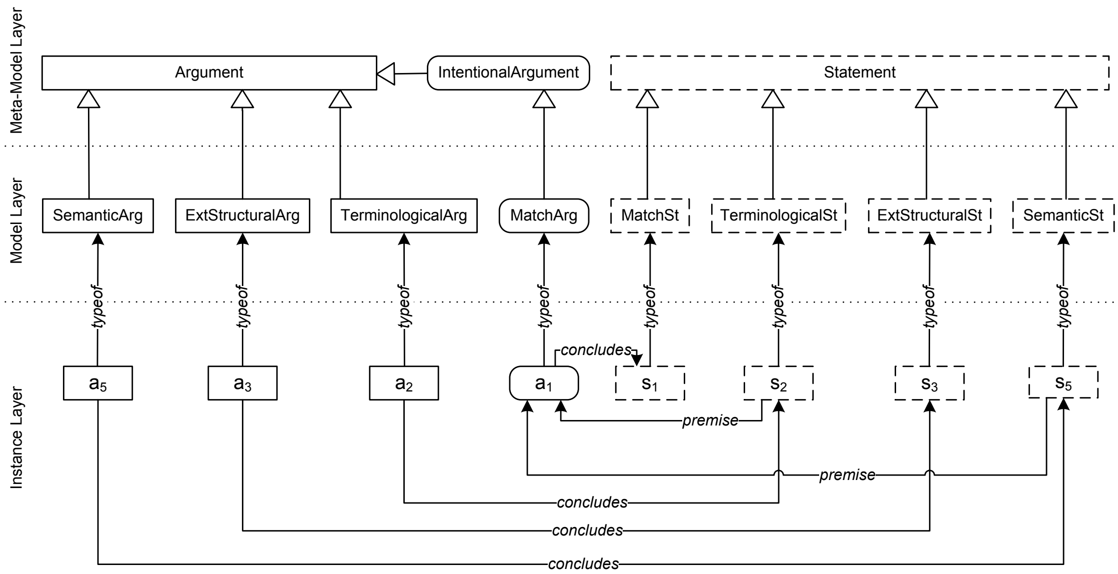

Figure 6.5 – The instantiation process after identifying the argument-instances' premises

Notice that the TLAF model depicted in Figure 6.2 states that only *MatchArg* instances can have premises since this is the unique argument type that is affected by others argument types.

Thus, in respect to the argument types that do not allow premises, agents are restricted to agree or disagree with their conclusions but they cannot argue upon those arguments. In other words, *MatchArg* instances are defeasible arguments while all other argument type instances are indefeasible arguments.

6.3.6 Identifying Statement' Conflicts

Finally, to automatically derive the support and attack relationships between argument-instances, it is necessary to establish the conflicts between statement-instances.

Definition 6.8 – Conflicts between statement-instances

A statement-instance s_a is in conflict with a statement-instance s_b iif s_a is concluded by an argument a of type x and s_b is concluded by an argument b of type y and x is related with y through R ($(x, y) \in R$) and both statement-instances satisfies a given condition ($condition(s_a, s_b)$) but their positions are contradictory ($pos_a \neq pos_b$).

Algorithm 6.3 applies this rule to all the statement-instances.

Algorithm 6.3. Setting conflicts between statement-instances.

Require: A TLAF model instantiated with argument and statement instances
Ensure: Establishes the set of conflicts between statement-instances

for all $(x, y) \in R$ **do**
    for all $a \in instA(x)$ **do**
        $(G_a, c_a, pos_a) = s_a = iconcl(a)$
        for all $b \in instA(y)$ **do**
            $(G_b, c_b, pos_b) = s_b = iconcl(b)$
            if $condition(s_a, s_b)$ *and* $(pos_a \neq pos_b)$
                $sconflict(s_a) = sconflict(s_a) \cup \{s_b\}$ $//s_a$ *is in conflict with* s_b
            end if
        end for
    end for
end for

Concerning this algorithm, it might be noticed two facts:

- The $condition$ function defined in the previous step is reutilized;
- The algorithm exploits the knowledge embedded in the rules to derive attack relationships (R3 and R4) between argument-instances with the purpose of reducing and driving the search/combination space between statements (as suggested in section 5.3.2).

Considering the running example, only one conflict between statement-instances exists (as depicted in Table 6.8 and Figure 6.6): s_3 is in conflict with s_1 because s_3 states that the external structure of entities related on s_1 (e_1 and e_1') is not similar, which conflicts with $e_1 = e_1'$ defended by s_1.

Table 6.8 – Existing conflicts between statement-instances after the fourth step

Statement-Instances			
ID	**Type**	**Content**	**sconflict**
s_1	MatchSt	$(G_1, c_1, +)$	
s_2	TerminologicalSt	$(G_2, c_2, +)$	
s_3	ExtStructuralSt	$(G_3, c_3, -)$	$\{s_1\}$
s_5	SemanticSt	$(G_5, c_5, +)$	

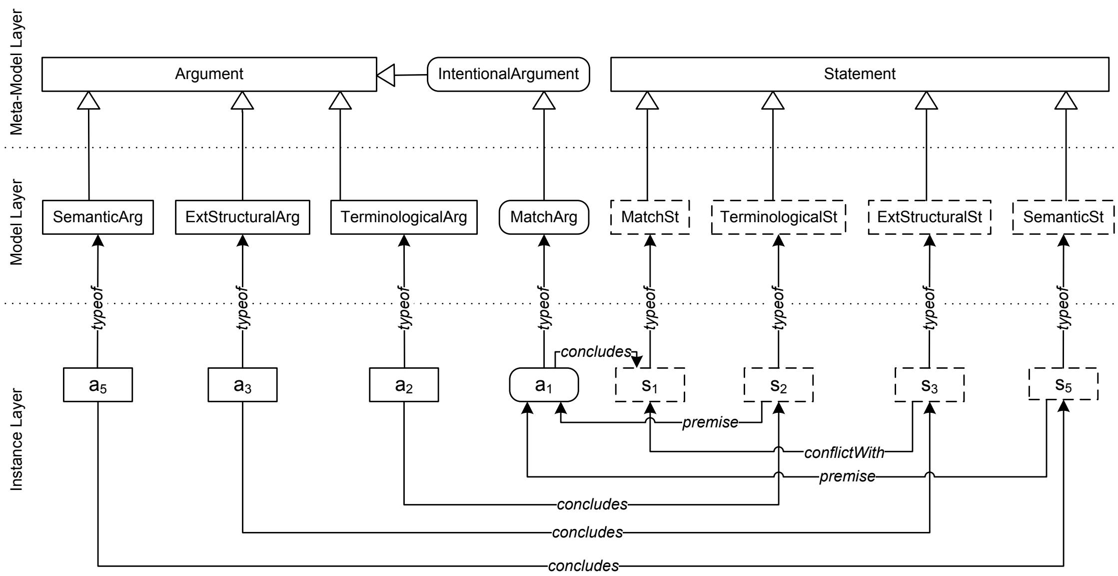

Figure 6.6 – The instantiation process after identifying the conflicts between statement-instances

On the other hand, one cannot say that s_2 is in conflict with s_3 or vice-versa because there is no relation between what is stated on each statement-instance. I.e. stating that two entities share a terminological relation is not seen as a reason to believe that the external structure of those entities also share the same relation and vice-versa. This is captured in the TLAF model by not specifying the R relation between *TerminologicalArg* and *ExtStructuralArg*. A similar reasoning can be done for statement-instance s_2 and s_5.

6.3.7 Deriving support and attacks relationships

Once the statement conflicts are defined, it is enough to apply the defined rules (described in sections 5.3.1 and 5.3.2) to automatically derive the support and attack relationships between argument-instances. The result is:

- Argument-instances a_2 and a_5 support argument-instance a_1, i.e. $\{(a_2, a_1), (a_5, a_1)\} \subseteq R_{sup}$;
- Argument-instance a_3 attacks argument-instance a_1, i.e. $\{(a_3, a_1)\} \subseteq R_{att}$.

The derived support and attack relationships are depicted in Figure 6.7.

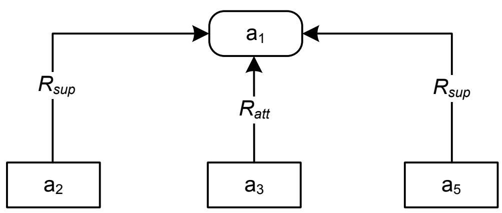

Figure 6.7 – Derived supports and attacks relationships

Once the support and attack relationships are derived, the argument instantiation phase is complete and therefore the $IP(AM_{Ag})$ is ready to be processed further in scope of the ANP.

6.4 Argument Evaluation Process

The process described in this section relies on and complements the arguments evaluation process described for the TLAF in section 5.4.

The set of evaluation functions of the TLAF argument evaluation process, described in section 5.4, (may) consider three dimensions:

- The strength values of (i) the argument-instance being evaluated and of (ii) its supporting and attacking argument-instances;
- The quantity of the direct support and direct attack of the argument-instance being evaluated;
- The quality of the direct support and direct attack of the argument-instance being evaluated.

Furthermore, it is quite obvious that these dimensions can be combined in several disparate ways. Thus, instead of providing concrete examples of evaluation functions, the aim of this section is to provide, in the context of the ontology matching negotiation scenario:

- An identification and a description of the information to be exploited by each of these dimensions;
- Generic rules to combine the information.

6.4.1 Strength Value Dimension

Since any evaluation function is responsible for evaluating the strength value of argument-instances, the strength value dimension is seen as an orthogonal dimension.

Example 6.1 – The strength value being used by a quantitative function

A very simple quantitative method is the subtraction of the linear average of the strength value of all argument-instances supporting the argument-instance being evaluated and the linear average of the strength value of all argument-instances attacking the argument-instance being evaluated.

Example 6.2 – The strength value being used by a qualitative dimension

A qualitative function may apply a similar method but instead of using the linear average it would use the weighted average such that the weight of each argument-instance attacking/supporting the argument-instance being evaluated reflects the quality of the attack/support.

The key to address the mutual dependency between the strength dimension and the other two dimensions relies on a method to provide an initial strength value to every argument-instances, such that the strength value is further used by the quantity and quality dimensions to generate a new strength value.

To provide the initial strength value of the argument-instances, it is proposed that the agent exploits the information from which the argument was generated. Hence, having in mind that the argument-instances are generated from:

- The correspondences ($c = (e, e', r, n)$) provided by matchers through the interpretation function (ψ);
- A pair of threshold values (tr_+ and tr_-) for each matcher to determine the matcher's position about the correspondence;
- The ordered set $\{\mathcal{V}_{min}, \dots, \mathcal{V}_{\overline{m}}, \dots, \mathcal{V}_{max}\}$ of values that an evaluation function returns,

the method to generate the initial strength value of an argument-instance grounds on the following principles/rules:

- If the matcher is in favor of the correspondence ($n \geq tr_+$) then:
    - The initial strength value of the resulting argument instance must be greater then $\mathcal{V}_{\overline{m}}$;
    - The higher the difference between the matcher confidence value on the correspondence (n) and the positive threshold value (tr_+) is, the higher the distance of the initial strength value to $\mathcal{V}_{\overline{m}}$ must be;
    - The initial strength value must be $\mathcal{V}_{max}$ if the confidence value on the correspondence is the maximum admissible value;
- If the matcher is against the correspondence ($n < tr_-$) then:
    - The initial strength value of the resulting argument-instance must be less then $\mathcal{V}_{\overline{m}}$;
    - The higher the difference between the negative threshold value (tr_-) and the matcher confidence value on the correspondence (n) is, the higher the distance of the initial strength value to $\mathcal{V}_{\overline{m}}$ must be;

- The initial strength value must be $\mathcal{V}_{min}$ if the confidence value on the correspondence is the minimum admissible value.

6.4.2 Quantitative Dimension

The quantitative dimension of an evaluation function is by definition domain-independent since it does not exploit the content (besides strength) of the argument-instances. Instead, it exploits the information concerning the relations between argument-instances only, namely the relations R_{sup} and R_{att} relationships to and from the argument-instance being evaluated. In that sense, it might exploit the quantity and the strength value of each argument-instance related to the one being evaluated. The Example 6.1 (see above) illustrates this approach. Conversely, Example 6.3 has a quantitative function that does not exploit the strength dimension.

Example 6.3 – A quantitative function ignoring the strength dimension

Consider a quantitative function that evaluates each support (or attack) relationship between arguments as a vote for (or against) the argument-instance being evaluated (respectively). In this sense, each support/attack relationship is valued as one, instead of its strength value. Thus, the resulting value would be the difference of votes for and against the argument-instance being evaluated.

6.4.3 Qualitative Dimension

The qualitative dimension of an evaluation function is domain-dependent since it relies on specific knowledge and preferences of the domain. In that sense, the type of the argument-instances related to the argument-instance being evaluated and their content (i.e. statements and reasoning mechanisms) is analyzed to determine the relevance/importance/influence that each relationship has. The way such relevance/importance/influence is expressed varies according to the kind of preference (see Example 6.4).

Example 6.4 – Evaluation functions exploiting domain preferences

Considering the TLAF model of section 6.2, the quality dimension of an evaluation function assigned to the instances of argument *MatchArg* might embody a preference by terminological argument (*TerminologicalArg*) over external structural arguments (*ExtStructuralArg*). Such preference may be:

- Expressed in an absolute way, i.e. the position of terminological arguments prevails over the external structural arguments or, alternatively,

- Expressed in a relative way, such that the position of terminological argument has a specific weight and the position of external structural argument has another (as suggested in Example 6.2).

Notice that, if the qualitative dimension is omitted in every evaluation function then the evaluation process becomes domain-independent.

6.4.4 Fixed-Point Computation Algorithm

This section specifies the algorithm required by the TLAF Evaluation process (cf. Definition 5.6) to iterate over the argument evaluation functions (F) in order to (re)evaluate the argument-instances strength.

Due to the cyclic, graph-shape structure established between the argument-instances through the R_{sup} and R_{att} the adoption of a fixed-point computation algorithm is proposed under the following assumptions:

- Every function $f \in F$ is defined as $f \rightarrow \mathcal{V}$ where $\mathcal{V} = [-1, 1]$ such that:
    - $\mathcal{V}_{min} = -1$;
    - $\mathcal{V}_{\overline{m}} = 0$; and
    - $\mathcal{V}_{max} = 1$;
- The initial strength of the all argument-instances ($mapV_0$) are already defined.

The proposed algorithm (Algorithm 6.4) computes iteratively over the arguments' evaluation functions until the maximum difference between the strength-value of every argument-instance in two consecutive iterations is less than a pre-defined constant ε. If the computation does not converge, it terminates after some maximal number of iterations (N). It is worth noticing that the convergence properties of the algorithm depend on the adopted evaluation functions (F).

Algorithm 6.4. Fixed-point algorithm.

Require: The TLAF instance-pool ($IP(TLAF)$) to evaluate, the maximal number of iterations (N) and a constant ε to determine if the computation convergence has been achieved.
Ensure: Every argument-instance has a new strength value computed.

```
Integer i = 0
do
    for all a ∈ AI do
        Function f = mapF(instA⁻(a))
        mapV^a_{i+1} = f(a, mapV_i)
    end for
    Double d = max(|mapV_{i+1} − mapV_i|)
    i = i + 1
until (i > N) ∨ (d < ε)
```

6.5 Summary

During this chapter, the generic Argument-based Negotiation Process and the Three-Layer Argumentation Framework, both previously introduced in Chapter 4 and Chapter 5 respectively, are exploited in the ontology matching negotiation context. The resulting approach is summarized as follows:

- In the Setup phase each agent informs the opponent agent of which ontology it is using and are defined some alignment requirements and/or properties;
- The result of the Data Acquisition phase is a set of correspondences collected from several sources freely selected by each agent;
- In the Argumentation Model Instantiation phase a new correspondences-to-arguments interpretation process was proposed and adopted. This process automatically generates arguments from the collected correspondences, based on a few parameters (thresholds) and two functions (ψ and $condition$);
- In the Argument evaluation phase, the quantitative, qualitative and strength dimensions of the arguments were analyzed and several rules were proposed for their combination. Furthermore, given the cyclic, graph-shaped structure of the argumentation instantiation, the adoption of a fixed-point computation algorithm was proposed;
- In the Agreement Attempt phase, the method to compute tentative agreements follows the unanimity principle between the negotiating agents. However, nothing was said regarding the method/strategy that each agent may adopt to decide either to accept the tentative agreement or reject it;
- The adaptation of the Persuasion phase to the ontology matching negotiation is simplified because the agents share the same argument representation formalism that is used to exchange the argument-instances.

The research and contributions presented so far overcomes most of the limitations identified in the state-of-the-art ontology matching argument-based negotiation approaches (cf. section 3.2.2). Some of the features introduced will be further demonstrated in the next chapter.

However, while the basis for allowing and supporting private argumentation models are already present in TLAF, this limitation will be objectively and extensively addressed in the next chapter. Furthermore, while the multiple preferred extensions issue was not directly addressed so far, this issue is indirectly addressed by the adoption of arbitrarily complex domain (dependent and independent) evaluation functions. The experiences presented in Chapter 8 will explicitly address this.

Chapter 7

EXTENSIBLE ARGUMENTATION FRAMEWORK

The ontology matching argument-based negotiation approach described in the previous chapter comprehends and requires the ability of the agents to extend the common argumentation model. Such ability allows an agent to exploit the knowledge that is not shared among the community members and, therefore, to profit from that knowledge.

Because the proposed argumentation framework (TLAF) lacks modularity and extensibility features it has been enhanced with modularity and extensibility constructs together with the respective semantics. These new features are described on section 7.1.

Because these new features are generic and domain independent, section 7.2 describes its application to the ontology matching negotiation context. This application demonstrates how the second and third limitations identified in the literature are overcome.

7.1 Formal Definition

Despite grounding on the Three-Layer Argumentation Framework (TLAF), the introduction of the modularity and extensibility features gives rise to a new argumentation modeling framework named Extensible Argumentation Framework (EAF). The EAF is formally described next.

Definition 7.1 – EAF

An EAF structure is a tuple $EAF = (E, INC)$, where:

- E is the set of entities of the EAF (similar to TLAF), and
- INC is the set of the included EAFs.

Accordingly, an EAF represents a self-contained unit of structured information that may be reused. Elements in an EAF are argumentation entities. An EAF may include a set of other EAF (represented through the set INC). The structure and semantics of EAF inclusion is formalized later in Definition 7.6.

Definition 7.2 – EAF Model Layer

A model layer associated with an EAF is a 9-tuple

$$ML(EAF) = (A, IA, S, M, R, \sigma, H_A, H_S, H_M)$$

where:

- A, IA, S, M, R and σ are defined as for the TLAF Model Layer (see Definition 5.2);
- $H_A \subseteq A \times A$ is an acyclic transitive relation that is established between similar argument types, such that $(a, b) \in H_A$ means that argument-instances of type a are understood as argument-instances of type b. While this function is vaguely similar to the subsumption relation (i.e. subclass between arguments) it does not have the same semantics and it is also constrained to a 1-1 relationship between arguments. In that sense, it is also represented as a partial function $H_A: A \rightarrow A$;
- $H_S \subseteq S \times S$ is an acyclic transitive relation equivalent to the H_A, but between statements instead. It is also represented as a partial function $H_S: S \rightarrow S$;
- $H_M \subseteq M \times M$ is an acyclic transitive relation equivalent to the H_A, but between reasoning mechanisms instead. It is also represented as a partial function $H_M: M \rightarrow M$.

In an EAF instance, arguments, statements and reasoning mechanisms can be arranged in a H relation as specified in H_A, H_S, H_M respectively. Their transitive closure follows the semantics defined in Definition 7.7.

Definition 7.3 – EAF Instance Layer

An instance layer associated with an EAF is a 6-tuple

$$IP(EAF) = (I, instA, instS, instM, \Sigma, sconflict)$$

where I, $instA$, $instS$, $instM$, Σ and $sconflict$ are defined as for the TLAF Instance Layer (see Definition 5.3).

At the Instance Layer, the H relations specified in the EAF Model Layer are exploited to infer that an instance is also of another type (cf. Definition 7.4).

Definition 7.4 – EAF Extended Instance Types

The extension of arguments (A), statements (S) and reasoning mechanisms (M) of an EAF model layer ($ML(EAF)$) are evaluated through the H_A, H_S and H_M relations as follows:

- Function $instA^{H_A}: A \rightarrow 2^I$ such that:

$$instA^{H_A}(a) = instA(a) \cup \left(\bigcup_{\forall x:(x,a) \in H_A} instA^{H_A}(x) \right)$$

- Function $instS^{H_S}: S \rightarrow 2^I$ such that:

$$instS^{H_S}(s) = instS(s) \cup \left(\bigcup_{\forall x:(x,s) \in H_S} instS^{H_S}(x) \right)$$

- Function $instM^{H_M}: M \rightarrow 2^I$ such that:

$$instM^{H_M}(m) = instM(m) \cup \left(\bigcup_{\forall x:(x,m) \in H_M} instM^{H_M}(x) \right)$$

Consequently, the inverse functions are defined as follows:

- $instA^{H_A^-}: AI \rightarrow 2^A$;
- $instS^{H_S^-}: SI \rightarrow 2^S$;
- $instM^{H_M^-}: MI \rightarrow 2^M$.

Because statement-instances used as premises of one argument-instance are also concluded by other argument-instances, an extended notion of premises is defined as follows.

Definition 7.5 – EAF Premises Extended Notion

Let a be an argument-instance. The extended set of premises of a is computed as $ipremise^*: AI \rightarrow 2^{SI}$ where:

$$ipremise^*(a) = ipremise(a) \cup \left(\bigcup_{i \in X(a)} ipremise^*(i) \right)$$

such that:

$$X(a) = \{ i \in AI: i \in instA(x) \wedge a \in instA(y) \wedge (x,y) \in R \wedge iconcl(i) \in ipremise(a) \}$$

Definition 7.6 – Modularization Constraints

Modularization allows the specification of an EAF based on other EAF(s). If an EAF (say EAF) imports another EAF (say EAF_1 with elements marked with subscript 1), that is, if $EAF_1 \in INC(EAF)$, it must satisfy the following modularization constraints:

a) $A_1 \subseteq A,\ IA_1 \subseteq IA,\ S_1 \subseteq S,\ M_1 \subseteq M$;

b) $\forall(a_x, a_y) \in H_{A_1} \Rightarrow (a_x, a_y) \in H_A \vee \{(a_x, a_i), (a_i, a_{i+1}), \dots, (a_{i+n}, a_y)\} \subseteq H_A$, i.e. if a_x "is understood as" a_y through H_{A_1} it must "be understood as" a_y through H_A too, even if indirectly;

c) $\forall(s_x, s_y) \in H_{S_1} \Rightarrow (s_x, s_y) \in H_S \vee \{(s_x, s_i), (s_i, s_{i+1}), \dots, (s_{i+n}, s_y)\} \subseteq H_S$, i.e. if s_x "is understood as" s_y through H_{S_1} it must "be understood as" s_y through H_S too, even if indirectly;

d) $\forall(m_x, m_y) \in H_{M_1} \Rightarrow (m_x, m_y) \in H_M \vee \{(m_x, m_i), (m_i, m_{i+1}), \dots, (m_{i+n}, m_y)\} \subseteq H_M$, i.e. if m_x "is understood as" m_y through H_{M_1} it must "be understood" as m_y through H_M too, even if indirectly;

e) $\forall(a_x, a_y) \in R_1 \Rightarrow (a_x, a_y) \in R \vee \{(a_x, a_i), (a_i, a_{i+1}), \dots, (a_{i+n}, a_y)\} \subseteq R$, i.e. if a R-relation exists in R_1 it must exist in R too, even if indirectly;

f) $\forall a \in A_1 \Rightarrow \sigma(a) = \sigma_1(a)$;

g) $I_1 \subseteq I, AI_1 \subseteq AI, SI_1 \subseteq SI, MI_1 \subseteq MI$;

h) $\forall a \in A_1 \Rightarrow instA_1(a) \subseteq instA^{H_A}(a)$;

i) $\forall s \in S_1 \Rightarrow instS_1(s) \subseteq instS^{H_S}(s)$;

j) $\forall m \in M_1 \Rightarrow instM_1(m) \subseteq instM^{H_M}(m)$;

k) $\forall a \in AI_1 \Rightarrow iconcl(a) = iconcl_1(a)$;

l) $\forall a \in AI_1 \Rightarrow ireason(a) = ireason_1(a)$;

m) $\forall a \in AI_1 \Rightarrow ipremise_1(a) \subseteq ipremise^*(a)$;

n) $\forall s \in SI_1 \Rightarrow sconflict_1(s) \subseteq sconflict(s)$.

The previous constraints guarantee that an EAF importing another EAF contains all the information of the imported EAF such that no information is lost. Relative exceptions are the H_A, H_S, H_M and R relations in the sense that the existing relationships between argumentation elements can be removed, providing that an indirect relation between the same argumentation elements must exist (as suggested by constraints b) to e)).

Definition 7.7 – EAF Interpretation

An interpretation of an EAF is a structure $\mathfrak{I} = (\Delta^{\mathfrak{I}}, A^{\mathfrak{I}}, S^{\mathfrak{I}}, M^{\mathfrak{I}}, I^{\mathfrak{I}})$, where $\Delta^{\mathfrak{I}}$, $A^{\mathfrak{I}}$, $S^{\mathfrak{I}}$, $M^{\mathfrak{I}}$ and $I^{\mathfrak{I}}$ are defined as for TLAF Interpretation (see Definition 5.4). An interpretation is a model of EAF if it satisfies the following properties:

- It satisfies all the properties required to be a model of TLAF;
- $\forall s_1 \forall s_2 : s_1 \in S \wedge s_2 \in S \ \wedge (s_1, s_2) \in H_S \Rightarrow S^{\mathfrak{I}}(s_1) \subseteq_{H_S} S^{\mathfrak{I}}(s_2)$;
- $\forall m_1 \forall m_2 : m_1 \in M \wedge m_2 \in M \ \wedge (m_1, m_2) \in H_M \Rightarrow M^{\mathfrak{I}}(m_1) \subseteq_{H_M} M^{\mathfrak{I}}(m_2)$;
- $\forall a_1 \forall a_2 : a_1 \in A \wedge a_2 \in A \ \wedge (a_1, a_2) \in H_A \Rightarrow \ A^{\mathfrak{I}}(a_1) \subseteq_{H_A} A^{\mathfrak{I}}(a_2) \ \wedge$

  $S^{\mathfrak{I}}(concl(a_1)) \subseteq_{H_S} S^{\mathfrak{I}}(concl(a_2)) \wedge \ M^{\mathfrak{I}}(reason(a_1)) \subseteq_{H_M} M^{\mathfrak{I}}(reason(a_2))$.

An EAF is unsatisfiable if it is does not have a model.

The following information can be inferred from the EAF, since it is true in all models of EAF:

- $H_A^* \subseteq A \times A$ is the transitive closure of the argument's hierarchy and is defined by $A^{\mathfrak{I}}(a_1) \subseteq_{H_A} A^{\mathfrak{I}}(a_2) \Leftrightarrow (a_1, a_2) \in H_A^*$;
- $H_S^* \subseteq S \times S$ is the transitive closure of the statement's hierarchy and is defined by $S^{\mathfrak{I}}(s_1) \subseteq_{H_S} S^{\mathfrak{I}}(s_2) \Leftrightarrow (s_1, s_2) \in H_S^*$;
- $H_M^* \subseteq M \times M$ is the transitive closure of the reasoning mechanisms hierarchy and is defined by $M^{\mathfrak{I}}(m_1) \subseteq_{H_M} M^{\mathfrak{I}}(m_2) \Leftrightarrow (m_1, m_2) \in H_M^*$.

Finally, notice that any TLAF structure can be transformed into an EAF structure straightforwardly. For that, consider that the TLAF structure to be transformed into an EAF structure is defined as $TLAF_1 = (E_1)$. The respective EAF structure is defined as $EAF = (E, INC)$, where:

- $E_1 \subseteq E$;
- $INC = \emptyset$;
- $H_A = \emptyset$;
- $H_S = \emptyset$;
- $H_M = \emptyset$.

7.2 Exploiting the EAF's Features

This section describes and exemplifies how the agents may exploit the modularity and extensibility features introduced by the EAF in the context of the ontology matching argumentation scenario.

Given that a TLAF structure is straightforwardly transformed into an EAF structure, consider the TLAF model presented in section 6.2 as the community's minimal common argumentation model about the ontology matching argumentation scenario, and from now on referred to as EAF_{c_1}.

7.2.1 Privately Extending the Common Argumentation Model

Consider that there is an agent (say Ag_1) belonging to a community of agents (say c_1). Furthermore, consider that the knowledge and needs of agent Ag_1 with respect to ontology matching argumentation scenarios is not completely captured by EAF_{c_1}. To capture the missing knowledge, Ag_1 needs to privately extend the EAF_{c_1} to $EAF_{Ag_1} = (E_{Ag_1}, \{EAF_{c_1}\})$ where $E_{c_1} \subseteq E_{Ag_1}$.

Assume that the missing knowledge of Ag_1 regards the argument type *ExtStructuralArg*. This argument type states that the external structures (or relations) of the source and target matching entities have a similarity relation (e.g. equality) that affects (positively or negatively) the relation between both entities. This is the case of the similarity between the sub-entities (e.g. sub-classes) of each of the entities (i.e. source and target): the more similar the sub-entities of the entities are, the more similar the entities are. Similar cases are the super-entities, siblings and other non-hierarchical relations between entities.

Considering the sub-entities relation only, Ag_1 may specify EAF_{Ag_1} as depicted in Figure 7.1 (elements filled in gray already exist in EAF_{c_1}), such that the argument type *SubEntitiesArg* captures the notion that a relation r (e.g. equality) holds between the sub-entities of the entities of a correspondence. This argument concludes a *SubEntitiesSt* through an *Heuristic* method (not depicted in Figure 7.1).

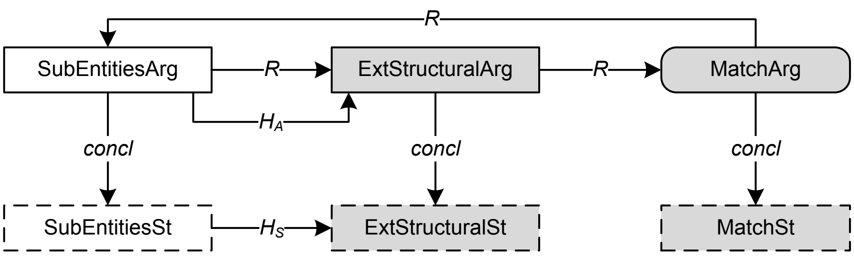

Figure 7.1 – Partial graphical representation of the argumentation entities introduced in EAF_{Ag_1}

In light of the depicted EAF_{Ag_1}, one may say that argument-instances of type *ExtStructuralArg* are affected by argument-instances of type *SubEntitiesArg* ($(SubEntitiesArg, ExtStructuralArg) \in R$) since their conclusions are seen as premises that lead to the conclusion about the relation between the external structure of the two entities. On the other hand, an argument-instance a_1 of type *SubEntitiesArg* is affected by the intention of accepting/rejecting a correspondence (i.e. a *MatchArg*) when the entities related on the *MatchArg* instance are sub-entities of the entities related by a_1. Thus, *MatchArg* affects *SubEntitiesArg* ($(MatchArg, SubEntitiesArg) \in R$). Generalizing, intentional arguments are being used to support/attack other arguments. Yet, it is worth noticing that this feature enables overcoming the third limitation (dependency between correspondences) identified in the literature (cf. section 3.2.2).

Furthermore, to promote the exchange of argument-instances of type *SubEntitiesArg* with the other agents of the community, those instances are understood as instances of type *ExtStructuralArg* in the context of EAF_{C_1}. For that, it has been stated that: $(SubEntitiesArg, ExtStructuralArg) \in H_A$ and $(SubEntitiesSt, ExtStructuralSt) \in H_S$. Because all argument types involved are applying reasoning mechanisms that are classified as Heuristic, no H_M relationship has been defined.

Despite focusing on one argument type only (*ExtStructuralArg*) and on a single ontological notion (sub-entities) the same principles can be employed (i) on other arguments (e.g. *IntStructuralArg*) and (ii) on other ontological notions (e.g. super-entities, siblings, domain and range).

Therefore, the EAF features allow the agents to conceptualize their private argumentation model, maintaining the compatibility and the semantic understanding with the remaining community.

7.2.2 Revising the Argumentation Model Instantiation Process

The agent is requested to revise the instantiation process according to its private argumentation model (e.g. EAF_{Ag_1}), addressing two issues:

- The condition function ($condition(s_a, s_b)$) of the new argument types, that determines the domain conditions under which two statements and argument-instances are related;
- The interpretation function (ψ) that will transform correspondences into argument-instances of the new argument types;

Taking as an example the model of EAF_{Ag_1}, the instantiation process is revised as follows.

7.2.2.1 Revising the Condition Function

Concerning the condition function, to exploit $(MatchArg, SubEntitiesArg) \in R$ in the processes responsible (i) for identifying the argument premises and (ii) for identifying the conflicts between statements, the condition function must also check the existence of the sub-entities relation between the statements' entities. Thus, considering two statements such that $s_a = (G_a, (e_a, e_a^{'}, r_a, n_a), pos_a)$ and $s_b = (G_b, (e_b, e_b^{'}, r_b, n_b), pos_b)$, the new relational condition is defined as:

$$cnd_2(s_a, s_b) = (isSubEntity(e_a, e_b) \wedge isSubEntity(e_a^{'}, e_b^{'}) \wedge r_a \equiv r_b)$$

such that the function $isSubEntity$ returns true iif the first entity is a sub-entity of the second entity.

7.2.2.2 Revising the Interpretation Function

Concerning the interpretation function, since each agent adopts a different interpretation function, each agent should upgrade its own interpretation function to be able to generate the arguments introduced in its private model too (e.g. EAF_{Ag_1}).

Assuming that the interpretation function of Ag_1 is the one introduced in section 6.3.2, Ag_1 would upgrade its interpretation function as depicted in Table 7.1.

Table 7.1 – Upgrading the example interpretation function

Matcher	Correspondence Content			Statement Type	Reasoning Mechanism	tr_+	tr_-
	e	e'	r				
G_1	any			MatchSt	Heuristic	0.90	0.70
G_2	any			TerminologicalSt	Heuristic	0.85	0.80
G_3	any			ExtStructuralSt	Heuristic	0.95	0.75
G_4	any			IntStructuralSt	Heuristic	0.90	0.80
G_5	any			SemanticSt	Semantic	1.00	1.00
G_6	any			SubEntitiesSt	Heuristic	0.60	0.50

7.2.2.3 Instantiation Example

To exemplify this revision, consider:

- The example described along section 6.3;
- The upgraded interpretation function (Table 7.1);
- The correspondences presented in Table 7.2;
- The ontology subsumption diagram depicted in Figure 7.2.

Table 7.2 – Extending the set of data collected by an agent

Generated by Matcher	Correspondence	
	ID	Content
G_6	c_6	$(e_1, e_1', =, 0.49)$
G_1	c_7	$(e_{11}, e_{11}', =, 0.50)$
G_1	c_8	$(e_{12}, e_{12}', =, 0.95)$

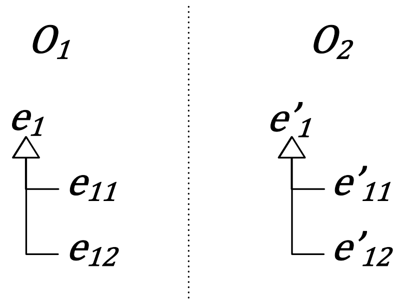

Figure 7.2 – The subsumption relationships between the entities of ontologies $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$

After applying the proposed instantiation process one would obtain the instances represented in Table 7.3, Table 7.4 and Table 7.5.

Table 7.3 – The new statement-instances

Statement-Instances			
ID	**Type**	**Content**	**sconflict**
s_6	SubEntitiesSt	$(G_6, c_6, -)$	
s_7	MatchSt	$(G_1, c_7, -)$	
s_8	MatchSt	$(G_1, c_8, +)$	$\{s_6\}$

Table 7.4 – The new instances of reasoning mechanisms

Reasoning Mechanisms		
ID	**Type**	**Content**
m_6	Heuristic	G_6

Table 7.5 – The new and the updated argument-instances

Argument-Instances				
ID	**Type**	**Premises**	**Concludes**	**Applies**
a_3	ExtStructuralArg	$\{s_6\}$	s_3	m_3
a_6	SubEntitiesArg	$\{s_7\}$	s_6	m_6
a_7	MatchArg	$\emptyset$	s_7	m_1
a_8	MatchArg	$\emptyset$	s_8	m_1

It is worth noticing that the existence of the new statement-instances, namely s_6, caused the identification of a premise of a previous argument-instance (i.e. a_3).

Furthermore, the relational condition cnd_2 allowed the identification of:

- s_7 as premise of a_6;
- s_8 in conflict with s_6.

On the other hand, similarly to a_1, the argument instances a_7 and a_8 may have several premises, but at this point they are unknown. Figure 7.3 depicts all support and attack relationships between argument-instances.

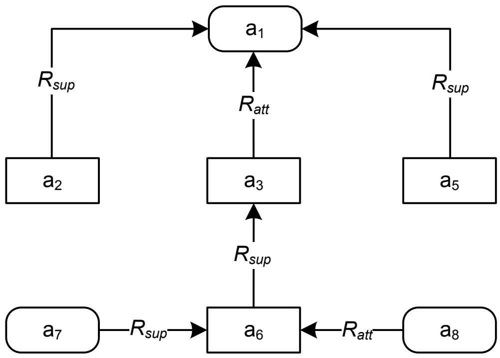

Figure 7.3 – Updated derived support and attack relationships

This section showed how to adapt the agent's internal reasoning process to the changes in its argumentation-model.

7.2.3 Exchanging Arguments

This section demonstrates the use of the EAF extensibility features to support the Persuasion phase (section 6.1.6). Persuasion occurs by exchanging non-intentional arguments that are unknown to the agents receiving them, which contribute to change the agent's position about the intentional arguments.

In the Persuasion phase, for each argument exchanged between two agents, one of four possible scenarios occurs:

- The type of the argument-instance exists in the community's argumentation model and:
  - The receiver agent does not re-interpret the received argument-instance (P1);
  - The receiver agent re-interprets the received argument-instance (P2);
- The type of the argument-instance does not exist in the community's argumentation model and:
  - The sender agent makes use of H_A, H_S and H_M relations to send the argument-instance as the most specific community's argument type (P3);
  - The sender agent is not able to send the argument-instance according to the community's argumentation model (P4).

As a running example, consider agents Ag_1 and Ag_2 both belonging to the c_1 community and that defines EAF_{c_1}. Agent Ag_1 makes use of the EAF_{Ag_1} described in previous section

($EAF_{Ag_1} = (E_{Ag_1}, \{EAF_{c_1}\})$), and agent Ag_2 makes use of $EAF_{c_1} \equiv EAF_{Ag_2}$. Figure 7.4 graphically and partially depicts the EAF_{Ag_1} (elements belonging to EAF_{c_1} are filled in gray).

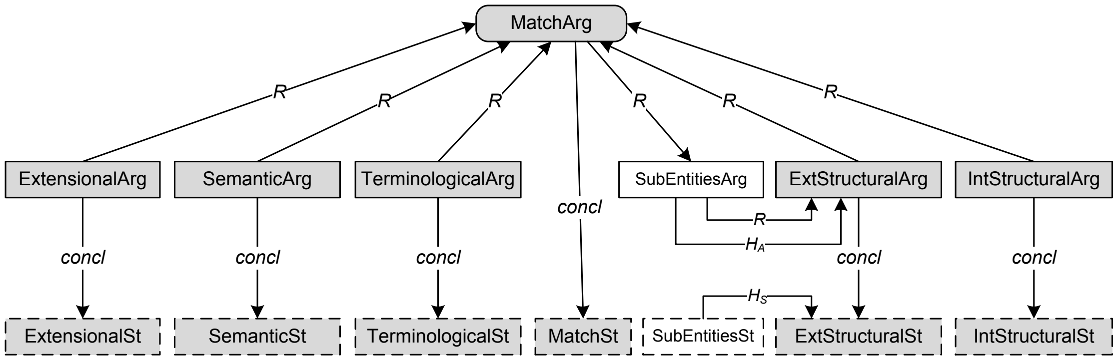

Figure 7.4 – The EAF model adopted by agent Ag_1

Yet, regarding the intentional argument-instance $a_1$consider that agent Ag_1 has the non-intentional argument-instances a_3 and a_6 while agent Ag_2 has the non-intentional argument-instances a_2 and a_5 as previously introduced. Additionally, agent Ag_2 has argument-instance a_9 of type *ExtStructuralArg* such that its conclusion is a statement $s_9 = (G_6, (e_1, e_1', \geq, 0.75), +)$ of type *ExtStructuralSt*. Figure 7.5 graphically and partially depicts this situation.

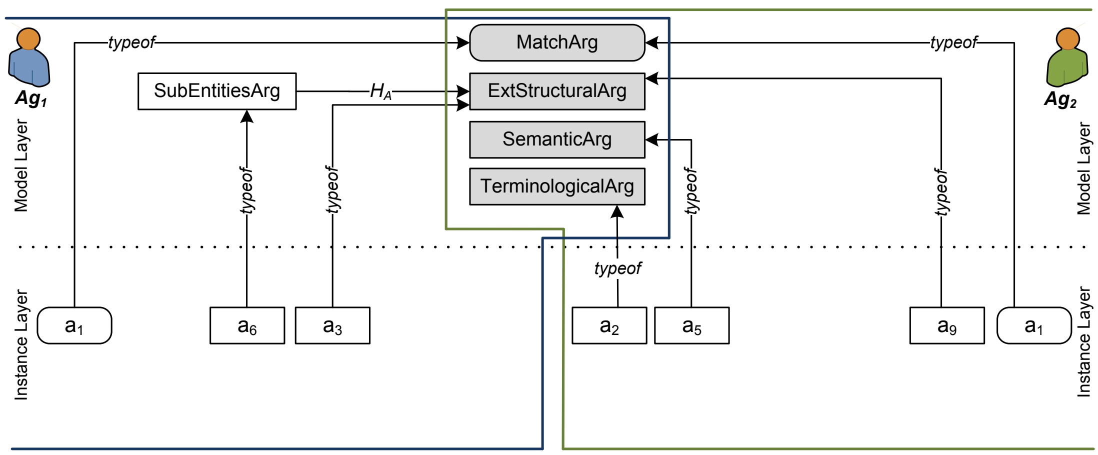

Figure 7.5 – Non-intentional argument-instances of agents Ag_1 and Ag_2

The scenario P1 corresponds to the simplest scenario where argument-instances are straightforwardly exchanged and similarly understood by both agents.

Example 7.1 – Exchanging Arguments in Scenario P1

Agent Ag_2 sends argument-instances a_2 and a_5 to Ag_1. Since Ag_1 is not able to reclassify the received argument-instances to another type, they will be similarly understood by both agents. The same also occurs when agent Ag_1 sends argument-instance a_3 to agent Ag_2. This is graphically depicted in Figure 7.6.

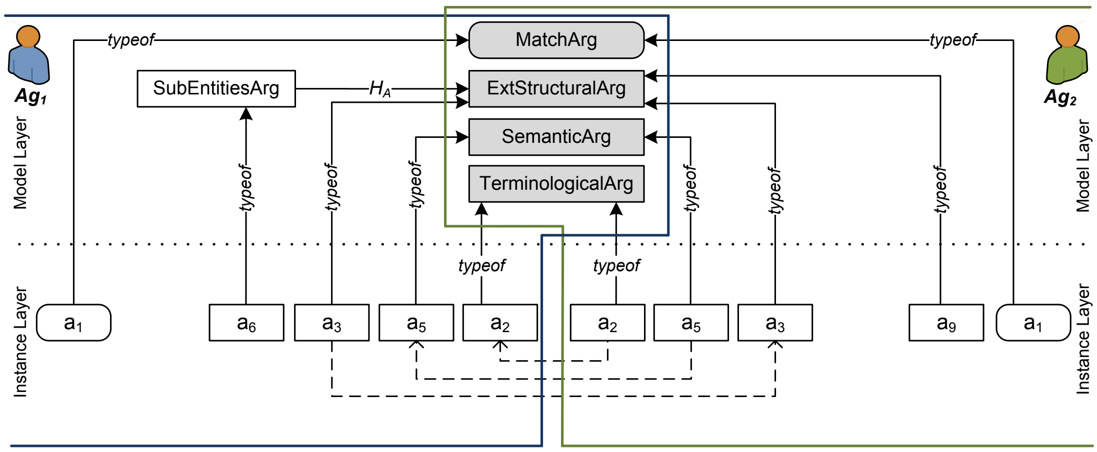

Figure 7.6 – Argument-instances exchanged according to scenario P1

In the scenario P2, argument-instances are also straightforwardly exchanged, but the receiver agent interprets the argument-instances differently than the sender agent. This implies that the receiver agent is able to re-classify the argument-instances to another type.

Example 7.2 – Exchanging Arguments in Scenario P2

Agent Ag_2 sends argument-instance a_9 to Ag_1. Thus, considering:

- The argument-instance content,
- The interpretation function of Ag_1 presented in Table 7.1 which defines that the correspondences generated by matcher G_6 give rise to statements of type *SubEntitiesSt*;
- The H_S-relation defined between *SubEntitiesSt* and *ExtStructuralSt*;
- The H_A-relation defined between *SubEntitiesArg* and *ExtStructuralArg*,

agent Ag_1 is able to re-classify argument-instance a_9 as being of type *SubEntitiesArg*. As a result, agent Ag_1 interprets a_9 as *SubEntitiesArg* instead of *ExtStructuralArg*. This is graphically depicted in Figure 7.7.

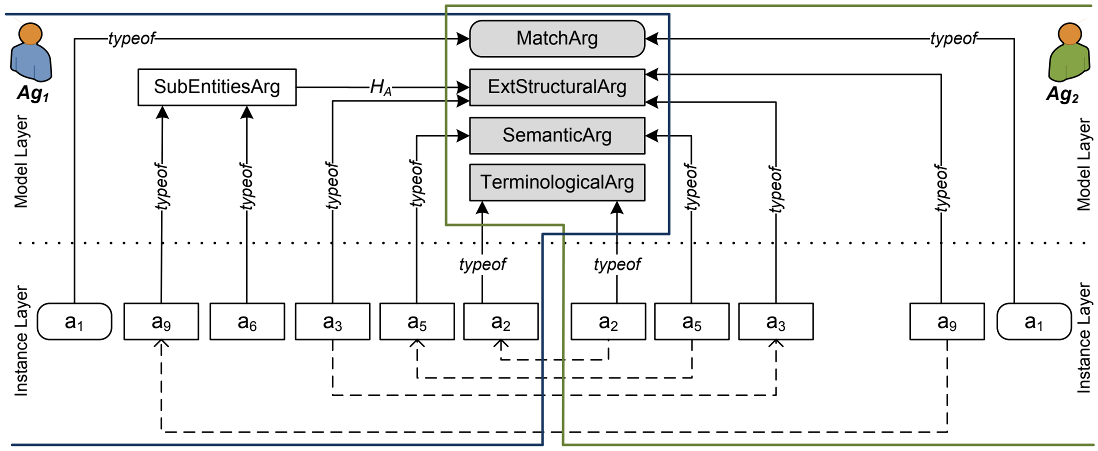

Figure 7.7 – Argument-instance exchanged according to scenario P2

With respect to the scenario P3, the sender realizes that the receiver agent is (probably) not able to understand the argument-instance because it is not (fully or partially) represented according to the common argumentation model. For the arguments exchange purpose, the sender agent reclassifies internally those argument-instances as the most specific common argument type through the existing H_A, H_S and H_M relations.

Example 7.3 – Exchanging Arguments in Scenario P3

This is the case of Ag_1 with respect to argument-instances of type *SubEntitiesArg* since it does not belong to EAF_{c_1} which makes Ag_2 unable to understand such argument-instances. In the case of Ag_1, the most specific common argument of *SubEntitiesArg* is *ExtStrucutureArg*.

Therefore, agent Ag_1 sends argument-instance a_6 to agent Ag_2 as being of type *ExtStructureArg* such that agent Ag_2 is able to understand it. This is graphically depicted in Figure 7.8.

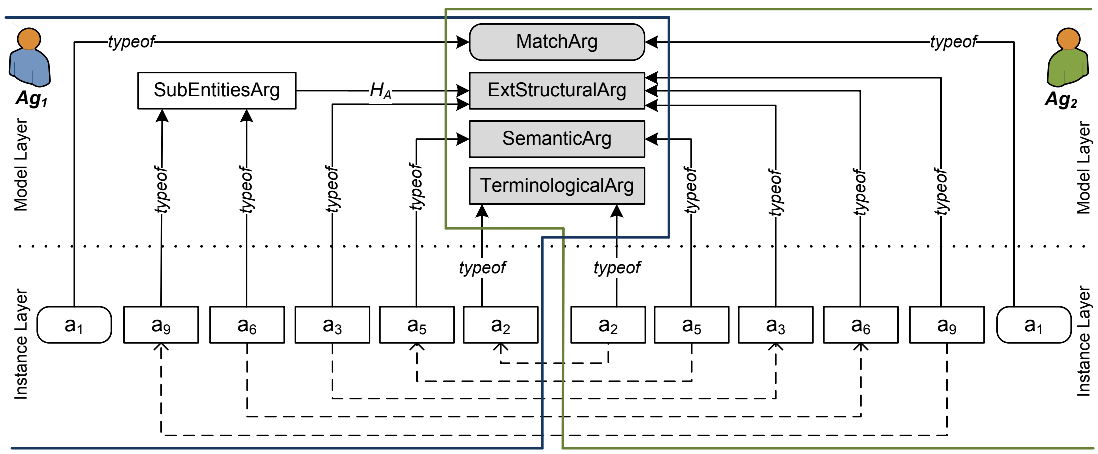

Figure 7.8 – Argument-instance exchanged according to scenario P3

In the scenario P4, the sender agent is not able to reclassify the argument-instances to the community's EAF. In such cases, two mutual exclusive possibilities arise:

- Those argument-instances are not exchanged;
- Those argument-instances are exchanged in a general way (e.g. classified as *Argument* only) expecting that the receiver agent is able to understand them based on their content (similarly to what was described for P2).

7.3 Summary

The Extensible Argumentation Framework was proposed in order to provide extensibility and modularization features to TLAF. The new features extend and drive the agents' capabilities to adapt their internal argument model and maintain compatibility with the community argumentation model.

While improving argumentation and argument modeling capabilities, the adoption of the EAF features implies minimal changes to the internal argumentation process and to the argumentation exchange process.

Because the EAF semantics is domain independent, its adoption to the ontology matching negotiation process was described through a running example.

Chapter 8

EXPERIMENTS

In order to evaluate the effectiveness of the proposed approach an empirical approach was adopted. The experiments aim to:

- Compare the proposed TLAF/EAF-based approach and the MbA/FDO approach (section 8.2);
- Evaluate the agents' ability to capture dependency between intentional arguments in the outcome of the negotiation process (section 8.3);
- Evaluate the relevance of the H relations (H_A, H_S and H_M) in the outcome of the negotiation process (section 8.4).

For that, the experiments are analyzed two-fold:

- Measuring the amount of resolved conflicts and its correctness;
- Measuring the accuracy (in terms of precision, recall and f-measure) of the agreed alignment achieved by the agents through the proposed argumentation process when compared to the agents' initial state, i.e. before the argumentation process.

Before that however, the setup of the experiments is described in the next section.

8.1 Experimental Set-up

This section describes the matching ontologies (section 8.1.1) and the characteristics of the negotiating agents (section 8.1.2) used in the experiments.

8.1.1 The Data Set

To carry out the experiments it is required (i) multiple pairs of ontologies modeling the same domain and (ii) a known reference alignment between each pair of ontologies. For that, seven ontologies representing different theories of a common real-world domain (and therefore reflecting real-world heterogeneity) were taken from the OAEI 2011 Conference Track repository[13]. These ontologies are listed in Table 8.1 together with a brief characterization in terms of number of classes, object properties, data properties and their Description Logic expressivity. Despite other ontologies are available in this repository, they were not used because there is no reference alignment available.

Table 8.1 – The test set of ontologies and their characteristics

Ontology	Named Classes	Object Properties	Data Properties	Expressivity[14]
Cmt	36	49	10	*ALCIN(D)*
Conference	60	46	18	*ALCHIF(D)*
ConfOf	38	13	23	*SIN(D)*
Edas	104	30	20	*ALCOIN(D)*
Ekaw	74	33	0	*SHIN*
Iasted	140	38	3	*ALCIN(D)*
Sigkdd	49	17	11	*ALEI(D)*

Since the ordering of the ontologies in each pair is irrelevant, a total of 21 ontology pairs were identified. Table 8.2 shows those pairs of ontologies together with the number of correspondences existing in the reference alignment. For the sake of brevity and simplicity, the experiment results are presented considering the negotiation of all individual alignments as just one huge alignment. Therefore the reference alignment contains 305 correspondences which correspond to the sum of the number of correspondences of all reference alignments.

Table 8.3 presents the accuracy of the matching systems that participated in the OAEI 2011 Conference Track[15] upon the data set, by presenting the (information retrieval) measures of Precision, Recall and F-Measure[16] [Makhoul et al. 1999].

[13] This repository is available at http://oaei.ontologymatching.org/2011/conference/index.html.
[14] The DL expressivity of an ontology is indicated by the concatenation of letters representing different DL operators [Baader 2003] and it was obtained by the Pellet reasoner [Sirin et al. 2007]
[15] The results are available at http://oaei.ontologymatching.org/2011/results/conference/index.html.
[16] F-Measure is the harmonic mean of Precision and Recall.

Table 8.2 – Pairs of ontologies used in the experiments

Pair ID	Source Ontology	Target Ontology	Nr. Correspondences[17]
1	Cmt	Conference	15
2	Cmt	ConfOf	16
3	Cmt	Edas	13
4	Cmt	Ekaw	11
5	Cmt	Iasted	4
6	Cmt	Sigkdd	12
7	Conference	ConfOf	15
8	Conference	Edas	17
9	Conference	Ekaw	25
10	Conference	Iasted	14
11	Conference	Sigkdd	15
12	ConfOf	Edas	19
13	ConfOf	Ekaw	20
14	ConfOf	Iasted	9
15	ConfOf	Sigkdd	7
16	Edas	Ekaw	23
17	Edas	Iasted	19
18	Edas	Sigkdd	15
19	Ekaw	Iasted	10
20	Ekaw	Sigkdd	11
21	Iasted	Sigkdd	15
		Total:	**305**

Table 8.3 – Accuracy of the OAEI 2011 participants for the data set used in the experiments

Matching System	Accuracy (%)		
	Precision	**Recall**	**F-Measure**
YAM++	78	56	65
CODI	74	57	64
LogMap	84	50	63
AgreementMaker	65	59	62
MaasMatcher	83	42	56
CSA	50	60	55
CIDER	64	45	53
MapSSS	55	47	51
Lily	36	47	41
AROMA	35	46	40
Optima	25	57	35
MapPSO	21	25	23
LDOA	10	56	17
MapEVO	15	2	4

8.1.2 The Agents

Three distinct agents (further referred to as agent A, B and C respectively) have been conceived.

[17] It respects to the number of correspondences existing in the reference alignment that establishes a relation between two entities of the same type. Considered types are Concept, Object Property and Data Property.

Respecting the data acquisition phase, the correspondences between each pair of ontologies were generated using the GECAD Ontology Alignment System (GOALS) [Maio and Silva 2009b; Maio and Silva 2009a]. Each agent used a distinct set of matching algorithms described in Annex 1.

Respecting the argument generation phase, the interpretation functions are also described in Annex 1.

Respecting the argument evaluation phase, two dimensions have to be considered. First, the argument evaluation functions used by the agents are introduced for each group of experiments. Second, regarding the selection of a preferred extension by the agents, a common criterion shared by all agents was defined. This criterion states that if as a result of the argument evaluation process the agent has available more than one preferred extension, the agent must adopt the preferred extension that differs less from the one adopted in the previous iteration of the negotiation process. This criterion turns the agents more consistent with their previous position on the negotiation and, at same time, turns more difficult an agent to change its initial position about a given correspondence.

All the experiments were executed for the agents' pair (A, B) and (A, C).

8.2 Comparison with MbA/FDO Approach

In order to compare the results of the TLAF/EAF-based approach with the results of MbA/FDO approach, several assumptions and constraints were made:

- Concerning the adopted argumentation model:
    - All agents adopt the community's argumentation model without privately extending it;
    - The adopted community argumentation model is partially and graphically represented in Figure 8.1. This argumentation model is a simplified version of the one introduced previously in section 6.2 which is a complete and explicit specification of the implicit argumentation model of the MbA/FDO approach (only the argument-types that will be effectively exploited are represented in Figure 8.1);
- Concerning the argument instantiation process, it follows the data acquisition process and the interpretation function presented in Annex 1. Briefly, each agent:
    - Exploits two matching algorithms: one to generate terminological arguments and another to generate external structural arguments;

    - The intentional arguments (*MatchArg*) are instantiated according to the most preferred argument-type (or value) of the agent;

- Concerning the argument evaluation process:
    - The indefeasible arguments (*TerminologicalArg* and *ExtStructuralArg*) are evaluated by a function returning a constant value k stating that the argument holds;
    - The intentional arguments (*MatchArg*) are evaluated by one of two distinct functions:
        - A function P mimicking the evaluation process followed by the MbA/FDO where the agents express their preferences regarding the argument-types (values in VAF terminology);
        - A function f_1 that exploits the quantitative dimension of the proposed argument evaluation process only (section 6.4).

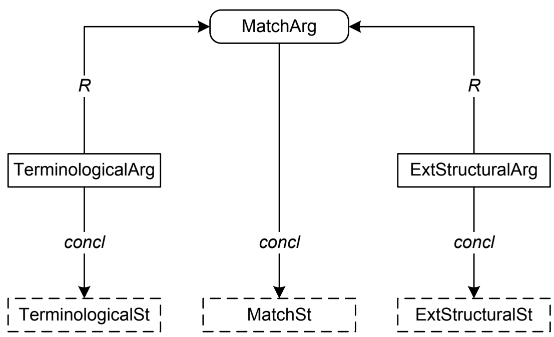

Figure 8.1 – The Model Layer used for comparison with the MbA/FDO approach

Thus, four distinct argumentation scenarios were elaborated (depicted in Table 8.4).

Table 8.4 – The evaluation functions adopted by the agents in the argumentation scenarios

Scenario	Agent A		Agent B		Agent C	
	Evaluation Functions		Evaluation Functions		Evaluation Functions	
	Defeasible	Indefeasible	Defeasible	Indefeasible	Defeasible	Indefeasible
1	$P = \{T, ES\}$	k	$P = \{ES, T\})$	k	$P = \{ES, T\})$	k
2	f_1		$P = \{ES, T\})$		$P = \{ES, T\})$	
3	$P = \{T, ES\}$		f_1		f_1	
4	f_1		f_1		f_1	

The first scenario mimics the MbA/FDO approach such that:

- Agent A prefers terminological arguments (T) to external structural arguments (ES). In MbA/FDO this is represented by stating $P = \{T, ES\}$;

- Agents B and C prefer external structural argument to terminological arguments, i.e. $P = \{ES, T\}$.

The other scenarios exploit the feature of the TLAF/EAF-based approach concerning the adoption of argument evaluation functions instead of preferences on argument-types. In these scenarios, at least one of the agents make use of the quantitative function f_1 to evaluate the intentional arguments (IA). This function counts the number of support relationships (n_{sup}) and the number attack relationships (n_{att}) of the argument-instance being evaluated (x), such that:

$$f_1(x, mapV_{i-1}) = \begin{cases} 1 & : n_{sup}(x) > n_{att}(x) \\ -1 & : n_{att}(x) > n_{sup}(x) \\ 0 & : otherwise \end{cases}$$

Yet, it is worth noticing that since the MbA/FDO approach does not have the notion of intentional argument, the argument-instantiation process has been adapted to instantiate the intentional arguments with the value of the most preferred argument-type of each agent. This guarantees that every intentional argument-instance is supported by the most preferred argument-instance. This constraint is depicted in Figure 8.2a (for agent A) and in Figure 8.2b (for agents B and C) where the elements filled in gray means they are instantiated similarly.

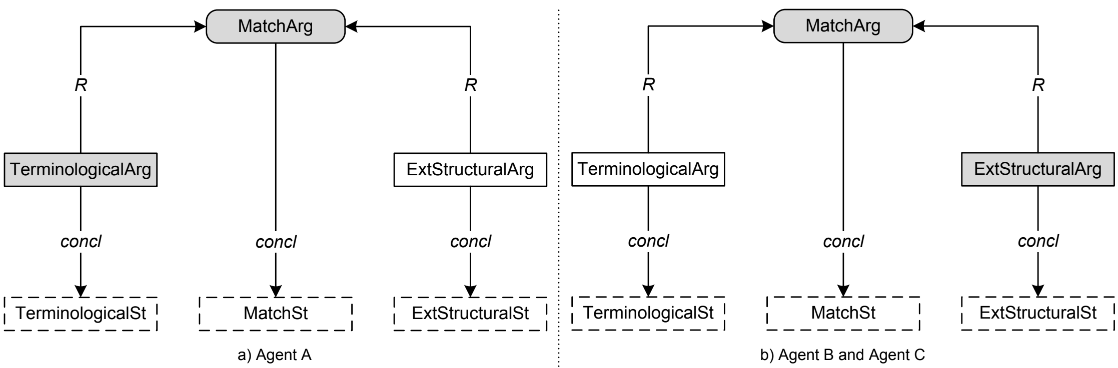

Figure 8.2 – Instantiation of intentional arguments according to the agents' preferences

8.2.1 Conflict Resolution

8.2.1.1 Results

In terms of conflict resolution, the result of the argument-based negotiation for the agents' pair (A, B) and (A, C) are depicted in Table 8.5 and Table 8.6 respectively. Each table shows: (i) the initial amount of conflict existing between the two agents before the argumentation process runs, (ii) the amount of conflict resolved during the argumentation process, (iii) the amount of remaining conflict after the argumentation process, (iv) the percentage of resolved conflicts and

(v) the percentage of conflict correctly resolved and (vi) badly resolved, both regarding the amount of resolved conflicts.

Table 8.5 – Analysis of the conflicts between agent A and B (S1 to S4)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
1	1319	0	1319	0.00	0.00	0.00
2		88	1231	6.67	67.05	32.95
3		170	1149	12.89	98.82	1.18
4		258	1061	19.56	87.98	12.02

Table 8.6 – Analysis of the conflicts between agent A and C (S1 to S4)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
1	493	0	493	0.00	0.00	0.00
2		51	442	10.34	78.43	21.57
3		75	418	15.21	98.67	1.33
4		126	367	25.56	90.48	9.52

8.2.1.2 Analysis and Discussion

The examination of these results shows that in the MbA/FDO approach (scenario 1) the agents were not able to resolve any conflict. This occurs due to the argument evaluation process of MbA/FDO approach because whenever an agent is able to generate an argument-instance of its preferred argument-type, the best the opponent agent can do is send an argument-instance of the same type but with an opposite position. I.e. an agent says a and the opponent agent negates a ($\neg a$). In this case, the agent would obtain two possible preferred extensions. Due the settled criterion on the preferred extension selection the agent opts with the preferred extension that maintains its previous position.

On the contrary, in the other three scenarios the agents were always able to resolve some conflicts. This occurs because the agents adopting the evaluation function f_1 evaluate their positions grounded in the amount of evidence (argument-instances) in favor and against the argument instead of considering a preferred argument-type only. Consequently, adopting the evaluation function f_1, an agent changes its initial position about a correspondence if the opponent agent is able to provide more evidences against that position than those the agent itself is able to provide in favor of that position. Thus, multiple preferred extensions occur when the amount of evidence in favor and against a correspondence is the same.

Additionally, it is worth noticing that despite the amount of resolved conflicts being relatively small (around 25% at the best case) the percentage of conflicts correctly resolved is very high (around 69% at the worst case).

8.2.2 Alignment Accuracy

8.2.2.1 Results

Table 8.7 summarizes and characterizes the accuracy of the following alignments:

- The alignment generated by each agent before the argumentation process. Notice that the agents' alignment before the argumentation process is the same independent of the argumentation scenario, because the argumentation model, data acquisition and argument generation are invariants for these four scenarios;
- For each agents' pair (A vs. B and A vs. C), the alignment resulting from:
  - The union of the agents' alignment generated before the argumentation process;
  - The intersection of the agents' alignment generated before the argumentation process;
  - The agreement obtained in each scenario after the argumentation process.

Table 8.7 – Summary and characterization of the alignments (S1 to S4)

Alignment		Correspondences		Accuracy(%)		
		Proposed	Correct	Precision	Recall	F-Measure
Agents Initial Proposal						
Agent A		305	176	57.70	57.70	57.70
Agent B		1450	172	11.86	56.39	19.60
Agent C		702	193	27.49	63.28	38.33
Agent A vs. Agent B						
Union		1537	199	12.95	65.25	21.61
Intersection		218	149	68.35	48.85	56.98
Agreed	**Scenario 1**	218	149	68.35	48.85	56.98
	Scenario 2	221	150	67.87	49.18	57.03
	Scenario 3	220	149	67.73	48.85	56.76
	Scenario 4	223	150	67.26	49.18	56.82
Agent A vs. Agent C						
Union		750	202	26.93	66,23	38.29
Intersection		257	167	64.98	54.75	59.43
Agreed	**Scenario 1**	257	167	64.98	54.75	59.43
	Scenario 2	260	168	64.62	55.08	59.47
	Scenario 3	257	167	64.98	54.75	59.53
	Scenario 4	260	168	64.62	55.08	59.47

Each alignment is characterized two-fold:

- Quantitatively, by presenting the number of proposed correspondences and the number of correct correspondences (i.e. the ones existing in the reference alignment);
- Qualitatively, by presenting the accuracy measures Precision, Recall and F-Measure.

Figure 8.3 and Figure 8.4 graphically complement the characterization of the alignments with respect to agents' pair (A, B) and (A, C) respectively.

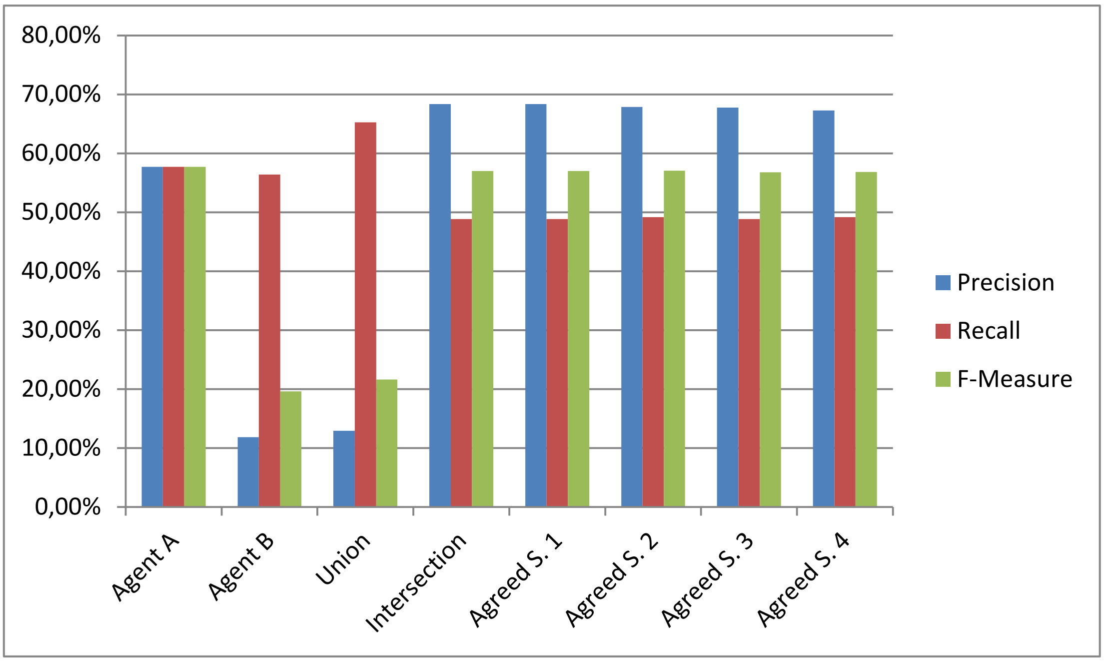

Figure 8.3 – Characterization of the alignments for the agents' pair A-B (S1 to S4)

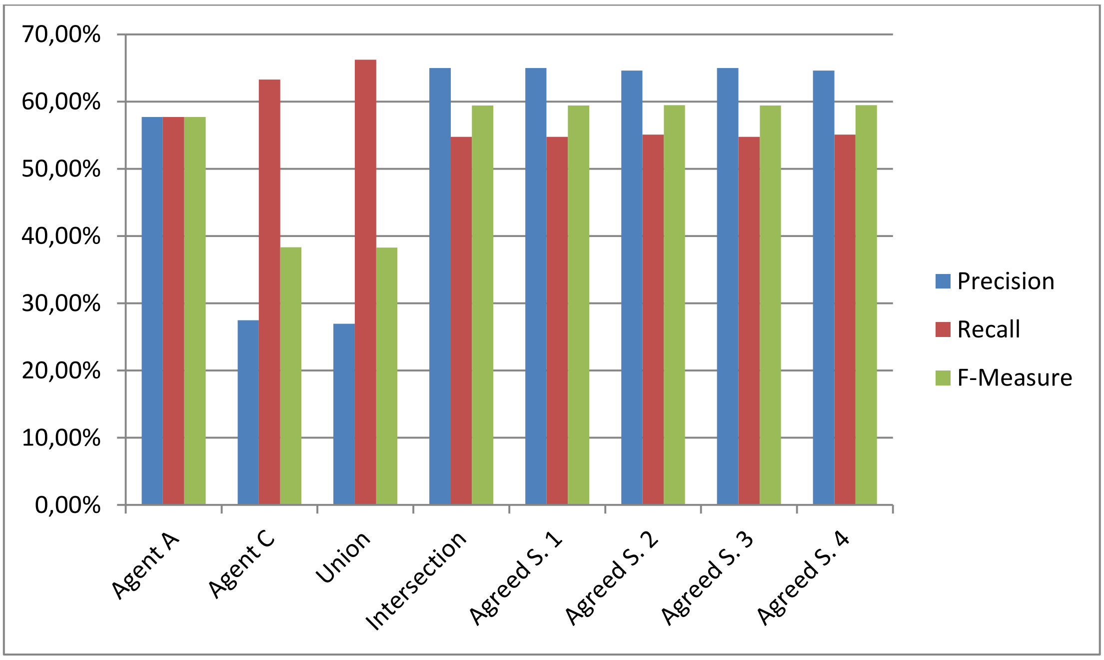

Figure 8.4 – Characterization of the alignments for the agent's pair A-C (S1 to S4)

8.2.2.2 Analysis and Discussion

The examination of these results shows that in the MbA/FDO approach (scenario 1) the agreed alignment corresponds exactly to the intersection of the alignments devised by the agents before the argumentation process runs. This occurs because none of the conflicts between the agents is resolved during the argumentation.

In the other scenarios, the alignment is slightly different from the alignment resulting from the intersection of the alignments devised by the agents before the argumentation process. These differences result from the resolved conflicts in which an agent was able to persuade its opponent to accept the inclusion of correspondences in the agreement. These small differences allow the perception of two issues:

- The high difficulty that an agent has to persuade its opponent to accept the inclusion of a correspondence in the agreement;
- The lack of evidence supporting the inclusion of a correspondence in the agreement in contrast with the evidences against such inclusion.

Comparing the alignment devised individually by the agents with the agreed alignment, it becomes clear that agents profit from the argumentation process:

- Agent A agreed an alignment with Agent B which is, in terms of f-measure, around 1% worse than the one devised by itself, but it has improved almost 2% in the agreed alignment with agent C;
- Agent B improved its alignment whose f-measure is around 20% to an agreed alignment whose f-measure is almost 57%;
- Agent C improved its alignment whose f-measure is around 38% to an agreed alignment whose f-measure is almost 59%.
- These f-measure improvements happened at the same time that conflicts are resolved.

As an overall comment, this set of experiments show that contrary to the MbA/FDO approach, the proposed EAF/TLAF-based approach is able to resolve conflicts even when the argumentation skills of the agents is very limited (only two kinds of argument exist). Simultaneously to the conflict resolution, the accuracy of the agreed alignment improves.

Further experiments will exploit other features of the proposed TLAF/EAF-based approach which are not supported by the MbA/FDO approach. In this sense, no more comparisons with MbA/FDO are provided.

8.3 Dependency between Correspondences

The experiments to evaluate the effect of the agents' ability to capture dependency between correspondences rely on the following assumptions and constraints:

- Concerning the adopted argumentation model:

- The community argumentation model is the same as used in the previous experiments (depicted in Figure 8.1), and from now on denominated as EAF_{FDO};
- To capture the dependency between correspondences, the agents privately extend EAF_{FDO} as depicted in Figure 8.5. The extended model (EAF_{DC}) adds the $(MatchArg, ExtStructuralArg) \in R$ to EAF_{FDO} such that an argument-instance of type *MatchArg* may affect an argument-instance of type *ExtStructuralArg* if there is a subsumption relationship (either *subClassOf* or *superClassOf*) between the concepts of the statements concluded by both argument-instances;

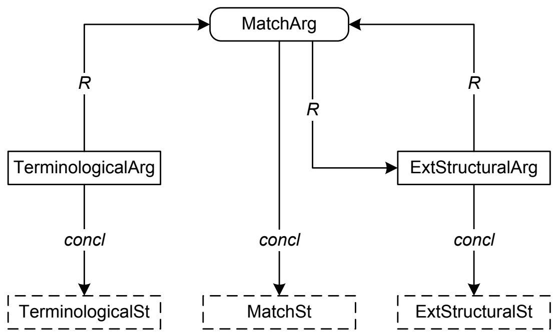

Figure 8.5 – The Model Layer used to study dependency between correspondences (EAF_{DC})

- Concerning the argument instantiation process, it follows the data acquisition process and the interpretation function presented in Annex 1. Briefly, each agent:
    - Exploits the same two matching algorithms as previously in scenarios 1 to 4 to instantiate the terminological and the external structural arguments;
    - Exploits a third matching algorithm to instantiate the intentional argument (*MatchArg*);
- Concerning the argument evaluation process:
    - The agents do not have any preferred argument-type;
    - The indefeasible arguments are evaluated by a function returning a constant value k stating that the argument holds;
    - The defeasible arguments are evaluated by the quantitative function f_2.

It is worth noticing that in EAF_{FDO} the *ExtStructuralArg* argument is indefeasible while in EAF_{DC} it is defeasible.

Thus, four distinct argumentation scenarios were elaborated (depicted in Table 8.8).

Table 8.8 – Argumentation scenarios to study dependency between correspondences

Scenario	Agent A			Agent B			Agent C		
	Arg. Model	Evaluation Functions		Arg. Model	Evaluation Functions		Arg. Model	Evaluation Functions	
		Dfsble	Indfsble		Dfsble	Indfsble		Dfsble	Indfsble
5	EAF_{FDO}	f_2	k	EAF_{FDO}	f_2	k	EAF_{FDO}	f_2	k
6	EAF_{DC}			EAF_{FDO}			EAF_{FDO}		
7	EAF_{FDO}			EAF_{DC}			EAF_{DC}		
8	EAF_{DC}			EAF_{DC}			EAF_{DC}		

Function f_2 returns the weighted average between (i) the strength value of the argument-instance being evaluated (x) and (ii) the normalized difference between the sum of the strength value of all argument-instances supporting it and the sum of the strength value of all argument-instances attacking it, such that:

$$f_2(x, mapV_{i-1}) = \frac{1}{3} mapV_{i-1}^{x} + \frac{2}{3}\left(\left(\sum_{yR_{sup}x} mapV_{i-1}^{y} - \sum_{yR_{att}x} mapV_{i-1}^{y}\right) \Big/ |yRx|\right)$$

Because in scenario 5 all the agents make use of EAF_{FDO}, none of the agents exploit the dependency between correspondences. On the contrary, in the other scenarios at least one of the agents is exploiting the dependency between correspondences (using EAF_{DC}). In that sense, scenario 5 serves as basis of comparison to the other scenarios.

8.3.1 Conflict Resolution

8.3.1.1 Results

The results of the experiments respecting the resolved conflicts are quantitatively and qualitatively summarized in Table 8.9 and Table 8.10 respectively.

Table 8.9 – Analysis of the conflicts between agent A and B (S5 to S8)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
5	995	769	226	77.29	93.76	6.24
6		742	253	74.57	95.15	4.85
7		740	255	74.37	95.27	4.73
8		757	238	76.08	94.06	5.94

Table 8.10 – Analysis of the conflicts between agent A and C (S5 to S8)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
5	360	221	139	66.39	90.50	9.50
6		213	147	59.17	94.37	5.63
7	356	206	150	57.87	93.69	6.31
8		219	137	61.52	93.15	6.85

8.3.1.2 Analysis and Discussion

Before addressing the dependency between the correspondences issue, it is important to roughly compare these results with the results obtained previously in scenarios 1 to 4. From this comparison, two issues become evident:

- The amount of initial conflict decreased from: (i) 1319 to 995 (for agent A vs. B) and (ii) from 493 to 360 (for agent A vs. C). This is justified by the use of a third matching algorithm to instantiate the intentional arguments in combination (through f_2) with the other arguments (terminological or external structural) supporting the intentional argument. In the cases where none of those arguments support the intentional argument, the evaluation may lead the agent to remove the intentional argument from its initial proposal. This is also the case in scenarios 7 and 8 where agent C is using EAF_{DC}, which lead to an additional retraction of four correspondences from its previous initial alignment;
- The percentage of resolved conflicts is much higher in these argumentation scenarios (57% in the worst case) than in the argumentation scenarios 1 to 4 (25% at the best case). This is also justified by the introduction of a third matcher together with the use of function f_2 instead of function f_1.

These two reasons are proved by analyzing the differences between the argumentation scenarios, namely scenario 4 with 5, where f_1 was replaced by f_2 and the constraint on the instantiation of the intentional arguments was removed.

Comparing the results of scenario 5 where none of the agents exploit the dependency feature with the scenarios where at least one of the agents exploits such feature (scenario 6 to 8), it is perceivable that:

- The number of resolved conflicts slightly decreased;
- The percentage of conflicts correctly resolved slightly increased.

The combination of these two facts allows concluding that the dependency feature helps improve the quality of the resolved conflicts.

8.3.2 Alignment Accuracy

8.3.2.1 Results

Table 8.11 summarizes and characterizes the alignments in terms of accuracy.

Notice that the alignment initially proposed by Agent C differs from scenarios 5 and 6 to scenarios 7 and 8. This difference was already justified previously in section 8.3.1.2. This also has impact in the analysis of the alignment resulting from the union of the alignments initially proposed by Agent A and Agent C. On the contrary, the alignment resulting from the intersection of the alignments initially proposed by Agent A and Agent C did not suffer any change.

Table 8.11 – Summary and characterization of the alignments (S5 to S8).

Alignment		Correspondences		Accuracy(%)		
		Proposed	Correct	Precision	Recall	F-Measure
Agents Initial Proposal						
Agent A		297	172	57.91	56.39	57.14
Agent B		1204	222	18.44	72.79	29.42
Agent C (S5/S6)		501	176	35.13	57.70	43.67
Agent C (S7/S8)		497	175	35.21	57.38	43.64
Agent A vs. Agent B						
Union		1248	228	18.27	74.75	29.36
Intersection		253	166	65.61	54.43	59.50
Agreed	**Scenario 5**	259	169	65.25	55.41	59.93
	Scenario 6	254	166	65.35	54.43	59.39
	Scenario 7	259	169	65.25	55.41	59.93
	Scenario 8	276	176	63.77	57.70	60.59
Agent A vs. Agent C						
Union (S5/S6)		579	202	34.89	66.23	45.70
Union (S7/S8)		575	201	34.96	65.90	45.68
Intersection		219	146	66.67	47.87	55.73
Agreed	**Scenario 5**	224	148	66.07	48.52	55.95
	Scenario 6	227	152	66.96	49.84	57.14
	Scenario 7	224	148	66.07	48.52	55.95
	Scenario 8	237	159	67.09	52.13	58.67

Figure 8.6 and Figure 8.7 graphically complement the characterization of the alignments with respect to agents' pair (A, B) and (A, C) respectively.

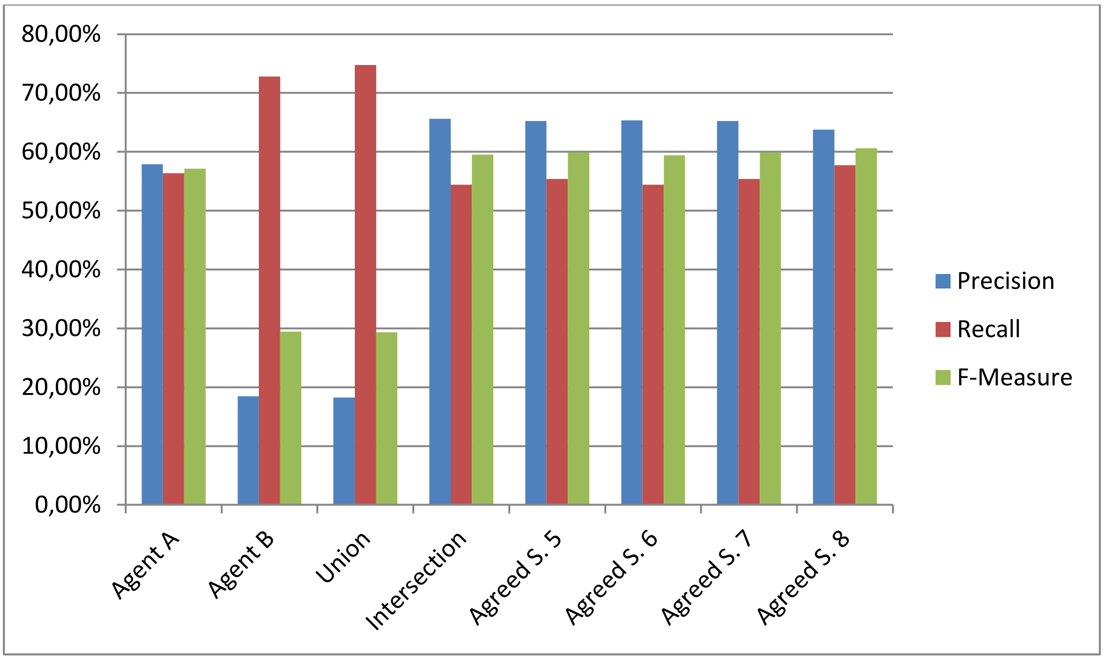

Figure 8.6 – Characterization of the alignments for the agent's pair A-B (S5 to S8)

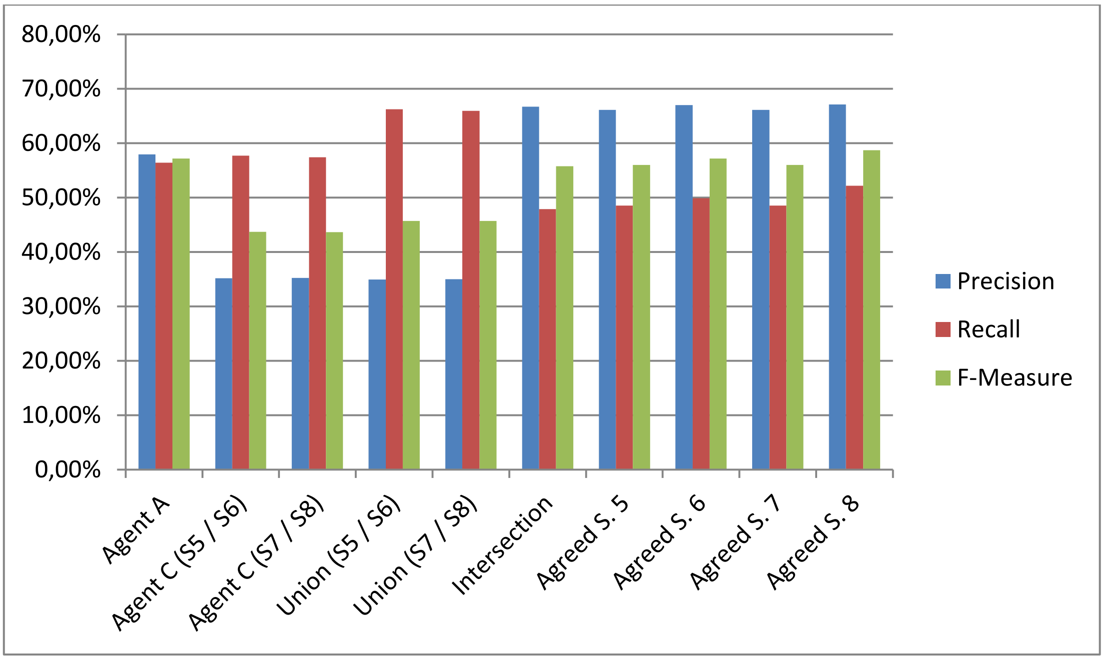

Figure 8.7 – Characterization of the alignments for the agent's pair A-C (S5 to S8)

8.3.2.2 Analysis and Discussion

The examination of previous results (scenarios 5-8) shows that comparing with scenarios 1-4, the accuracy of the alignments before argumentation significantly improved in terms of f-measure (around 10% for agent B and 5% for agent C).

The agreed alignment outperforms (in terms of f-measure) the alignment resulting from the intersection of the alignments initially proposed by the participating agents. Furthermore, it is perceived that the scenarios where at least one of the agents exploited the dependency feature (scenarios 6 to 8) the f-measure is greater than in the scenario 5 where none of the agents exploited such features. This is even more evident in scenario 8 where the two agents are exploiting simultaneously the dependency feature.

Therefore, it can be concluded that the dependency feature helps improve the alignment accuracy while it also helps improve the quality of the resolved conflicts (as seen in section 8.3.1.2).

8.4 The H Relations

In order to show the relevance of the features introduced by the EAF, namely the H relations (H_A, H_S and H_M), in the proposed argument-based negotiation approach, it has been decided to use the EAF_{DC} as the common argumentation model to all agents, and extended it differently and privately for each agent. Agents A, B and C use the argumentation model graphically and partially depicted in Figure 8.8, Figure 8.9 and Figure 8.10 respectively. These argumentation models are further referred to as EAF_A, EAF_B and EAF_C respectively. In each of these argumentation models, the elements belonging to the common argumentation model (EAF_{DC}) are filled in gray.

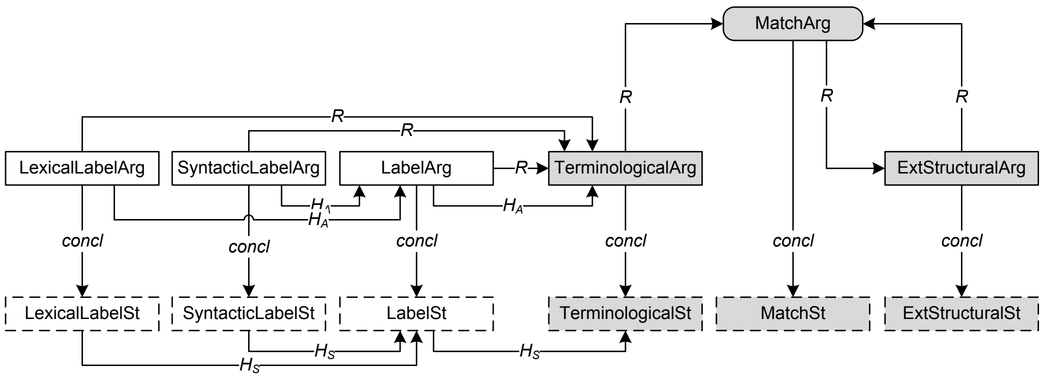

Figure 8.8 – The argumentation model internally adopted by Agent A (EAF_A)

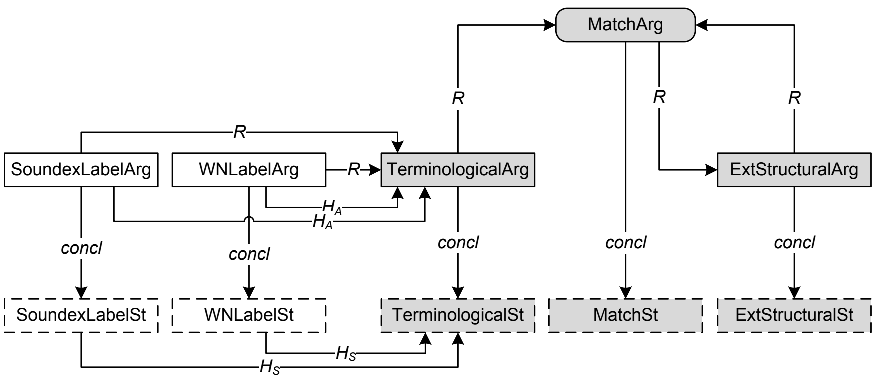

Figure 8.9 – The argumentation model internally adopted by Agent B (EAF_B)

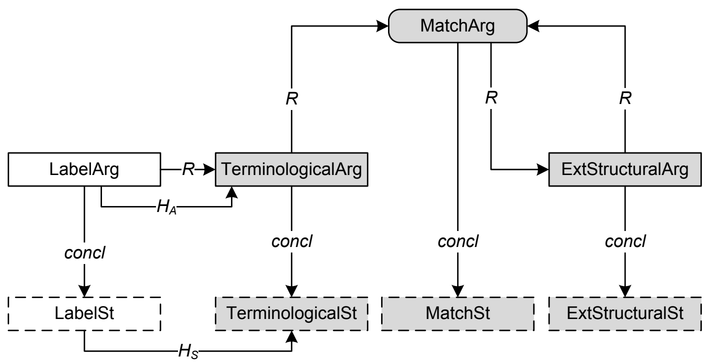

Figure 8.10 – The argumentation model internally adopted by Agent C (EAF_C)

Contrary to the EAF_{DC} where the *TerminologicalArg* argument is indefeasible, in these three argumentation models the *TerminologicalArg* argument become defeasible. For that, each agent introduced new arguments affecting it. E.g. in the EAF_C the terminological argument is affected by a label argument, i.e. $(LabelArg, TerminologicalArg) \in R$. On the other hand, the introduced arguments are indefeasible.

With respect to the new arguments, they must be interpreted as follows:

- *LabelArg* (either in EAF_A or EAF_C) is an argument stating the similarity of the labels associated with two ontological entities;
- *SyntacticLabelArg* is a specialization of the *LabelArg* argument, considering a syntactic technique is applied to evaluate the similarity of the labels;
- *LexicalLabelArg* is also a specialization of the *LabelArg* argument that exploits a lexical resource (e.g. WordNet [Fellbaum 1998]);
- *WNLabelArg* is an argument similar to the *LexicalLabelArg* which has been named differently by agent C. Notice that despite the grounds of these two arguments being the same, the conceptualization made by the agents A and C regarding these arguments differs. This is possible because each of these conceptualizations is made privately by the respective agent;
- *SoundexLabelArg* is an argument stating the similarity of the labels considering the sound of the labels when pronounced [Russell 1918; Russell 1922].

Consequently EAF_A, EAF_B and EAF_C reflect some heterogeneity at the conceptual level of the ontology matching knowledge concerning the terminological similarity.

To foster the exchange of arguments H_A and H_S relations have been established between the new elements (i.e. arguments and statements) and the elements of the common argumentation model. Given that, instances of these new arguments might be exchanged as instances of the terminological argument (*TerminologicalArg*).

Further, those argument-instances may be reclassified internally by the agents as depicted in Table 8.12. Notice that this table reflects the agents' internal and private knowledge, thus an agent does not know the reclassification rules of the other agents.

Table 8.12 – Reclassification of argument-instances exchanged as terminological

Original Argument-Type Sent \ Reclassified as		EAF_A			EAF_B		EAF_C
		LabelArg	***Syntatic LabelArg***	***Lexical LabelArg***	***Soundex LabelArg***	***WN LabelArg***	***LabelArg***
EAF_A	***LabelArg***	n/a					X
	SyntaticLabelArg						X
	LexicalLabelArg					X	X
EAF_B	***SoundexLabelArg***	X			n/a		
	WNLabelArg			X			
EAF_C	***LabelArg*** G_{C1}	X					n/a
	LabelArg G_{C2}			X			

As an example, argument-instances of type $EAF_A\colon LexicalLabelArg$ are exchanged as $EAF_{DC}\colon TerminologicalArg$ instances which are further reclassified as (i) instances of $EAF_B\colon WNLabelArg$ by agent B and (ii) as instances of $EAF_C\colon LabelArg$ by agent C.

Concerning the argument instantiation process, it follows the data acquisition process and the interpretation function presented in Annex 1. Briefly, each agent exploits a single matching algorithm by argument-type of the adopted argumentation model. An exception regards the argument-type $EAF_C\colon LabelArg$ which is instantiated by agent C based on two matching algorithms: G_{C1} and G_{C2} (cf. Annex 1 for details).

Concerning the argument evaluation process, indefeasible arguments are evaluated through a function returning a constant value k stating that the argument holds. The defeasible arguments are evaluated through the function f_2 previously introduced (cf. section 8.3).

Accordingly, based on (i) the agents ability to exchange arguments based on the H relations and (ii) on the agents ability to reclassify terminological argument-instances, five new argumentation scenarios have been delineated (depicted Table 8.13).

Table 8.13 – Argumentation scenarios to study the relevance of H relations

Scenario	Agent A			Agent B			Agent C		
	Arg. Model	**H's**	**Reclas.**	**Arg. Model**	**H's**	**Reclas.**	**Arg. Model**	**H's**	**Reclas.**
9	EAF_A	No	No	EAF_B	No	No	EAF_C	No	No
10		Yes	No		No	Yes		No	Yes
11		No	Yes		Yes	No		Yes	No
12		Yes	Yes		Yes	Yes		Yes	Yes
13		Yes	No		Yes	No		Yes	No

Scenario 9 serves as basis of comparison to the other scenarios since none of the agents exploit both (i) the H relation and (ii) the reclassification feature. Regarding scenario 13, the outcome will be the same as scenario 9 because for the same correspondence an agent sends through the H relations several terminological argument-instances which are not further reclassified by the receiving agent and therefore are seen as duplicated argument-instances and, thus, discarded. Given this, scenario 13 is no longer considered.

8.4.1 Conflict Resolution

8.4.1.1 Results

Table 8.9 and Table 8.10 summarize quantitatively and qualitatively the conflicts resolved by the argument-based negotiation for the agents' pair (A, B) and (A, C) respectively.

Table 8.14 – Analysis of the conflicts between agent A and B (S9 to S12)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
9	293	3	290	1.02	100.00	0.00
10		243	50	82.94	89.30	10.70
11		29	264	9.90	75.86	24.14
12		257	36	87.71	89.88	10.12

Table 8.15 – Analysis of the conflicts between agent A and C (S9 to S12)

Scenario	Number of Conflicts			Conflicts Resolved (%)		
	Initial	Resolved	Remain	Total	Correctly	Badly
9	50	24	26	48.00	75.00	25.00
10		50	0	100.00	66.00	34.00
11		37	13	74.00	67.57	32.43
12		38	12	76.00	71.05	28.95

8.4.1.2 Analysis and Discussion

Concerning the initial amount of conflict it is perceivable that as long as the agents argumentation model evolves (from EAF_{FDO} to EAF_{DC} and from EAF_{DC} to EAF_A, EAF_B and EAF_C) and therefore the agents matching capability improves, the initial amount of conflict between agents decrease as follows:

- Agent A vs. Agent B: from 1319 initial conflicts to 995 and from 995 to 293 initial conflicts;
- Agent A vs. Agent C: from 493 initial conflicts to 360 and from 360 to 50 initial conflicts.

Despite this fact, the rate of resolved conflicts is in most of the scenarios high. This assumes even more relevance by the fact that at least 66% of the resolved conflicts were correctly resolved.

Small exceptions in the high rates of the resolved conflicts are the scenarios 9 and 11 for the agents' pair (A, B). Regarding scenario 9, this is plausibly justified by the fact that both agents are using sets of arguments internally that are not further exchanged between them since none of the agents is exploiting the defined H relations. Given that, the arguments exchanged through the common argumentation model were insufficient to successfully persuade the opponent agent. Regarding scenario 11, agent B is exploiting the H relations which allows agent A to reclassify all argument-instances exploited privately by agent B. This has granted that the amount of resolved conflicts grow from 1.02% to 9.90% only. This allows concluding that the persuasiveness of agent B is inefficient when compared to the persuasiveness of agent A in the opposite scenario (i.e. in scenario 11) where the amount of resolved conflicts grow from 1.02% to 87.71%.

By comparing the amount of resolved conflicts and its correctness between scenario 9 (where none of the agents exploit the H relations neither the arguments reclassification feature) and scenario 12 (where both features are exploited), the usefulness of these two features regarding the conflict resolution issue becomes evident. The results of scenarios 10 and 11 allow us to conclude on the persuasiveness of each agent. Hence, agent A was very persuasive against both agent B and agent C. Instead, agent C was also very persuasive (but less than agent A) while agent B was inefficient as debated.

8.4.2 Alignment Accuracy

8.4.2.1 Results

Table 8.16 summarizes and characterizes the alignments in terms of accuracy.

Table 8.16 – Summary and characterization of the alignments (S9 to S12)

Alignment		Correspondences		Accuracy(%)		
		Proposed	Correct	Precision	Recall	F-Measure
Agents Initial Proposal						
Agent A		198	151	76.26	49.51	60.04
Agent B		455	176	38.68	57.70	46.32
Agent C		176	134	76.14	43.93	55.72
Agent A vs. Agent B						
Union		473	182	38.48	59.67	46.79
Intersection		180	145	80.56	47.54	59.79
Agreed	**Scenario 9**	183	148	80.87	48.52	60.66
	Scenario 10	182	147	80.77	48.20	60.37
	Scenario 11	191	155	81.15	50.82	62.50
	Scenario 12	185	150	81.08	49.18	61.22
Agent A vs. Agent C						
Union		212	157	74.06	51.48	60.74
Intersection		162	128	79.01	41.97	54.82
Agreed	**Scenario 9**	186	146	78.49	47.87	59.47
	Scenario 10	200	153	76.50	50.16	60.59
	Scenario 11	186	146	78.49	47.87	59.47
	Scenario 12	188	147	78.19	48.20	59.63

Figure 8.11 and Figure 8.12 graphically complement the characterization of the alignments with respect to agents' pair (A, B) and (A, C) respectively.

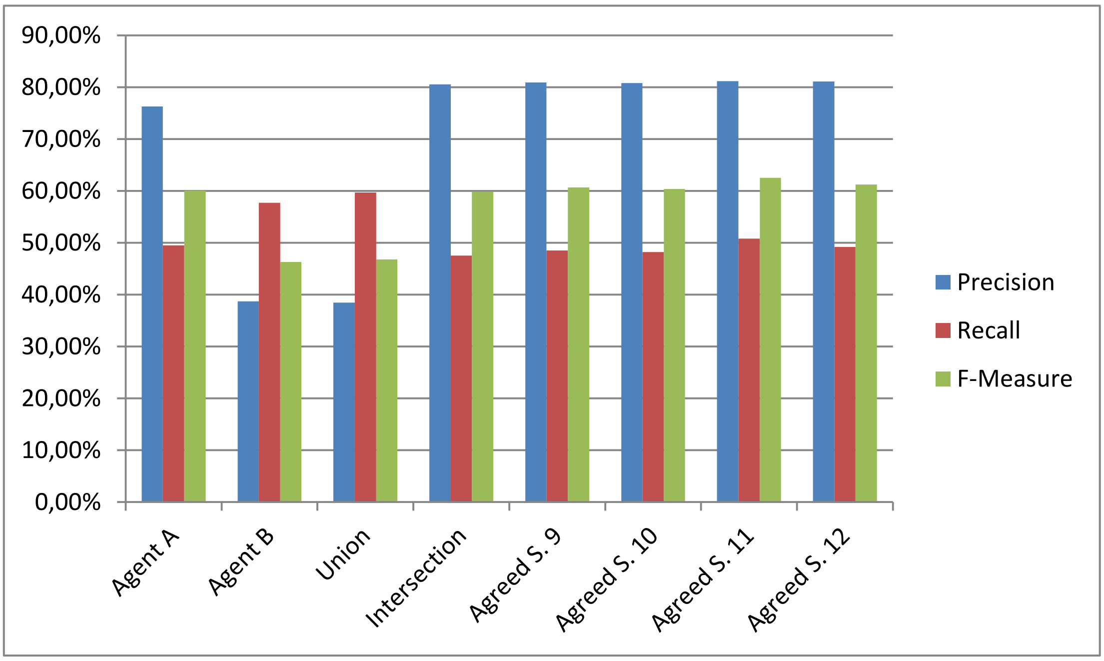

Figure 8.11 – Characterization of the alignments for the agent's pair A-B (S9 to S12)

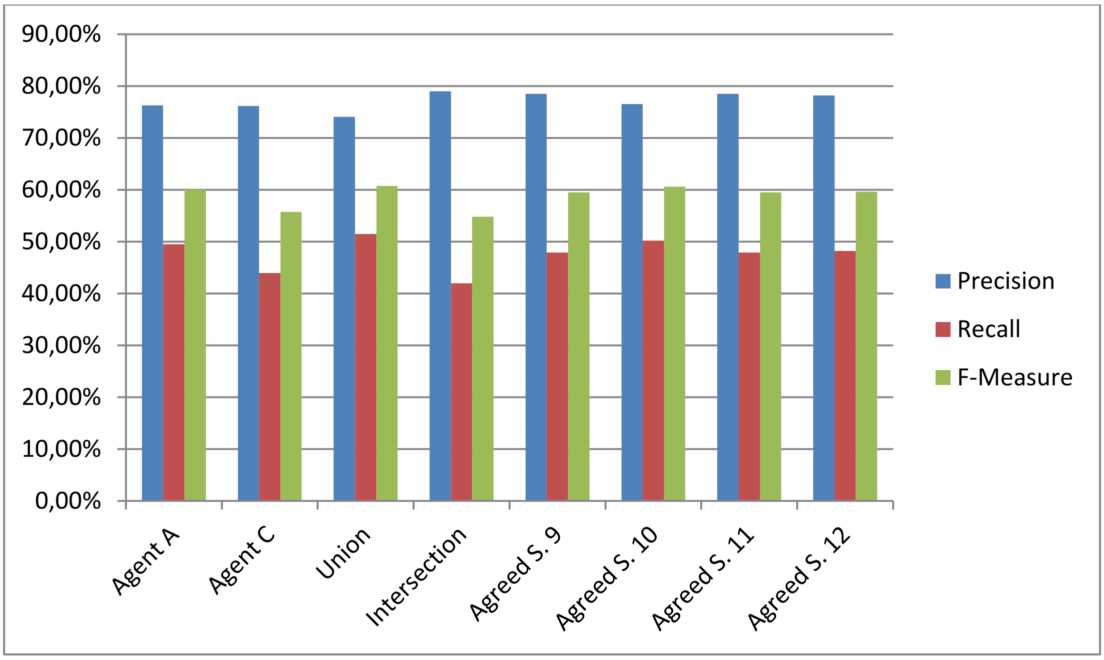

Figure 8.12 – Characterization of the alignments for the agent's pair A-C (S9 to S12)

8.4.2.2 Analysis and Discussion

The examination of these results confirms the idea suggested in the previous section by the decreasing amount of initial conflict that the accuracy of the alignments initially proposed by each agent improves as long as the agents argumentation model evolves (from EAF_{FDO} to EAF_{DC} and from EAF_{DC} to EAF_A, EAF_B and EAF_C).

In scenarios 10 to 12 where the H relations and the reclassification feature are exploited by the agents, in terms of f-measure, the agreed alignment is better or equal to the agreed alignment in scenario 9 where those features are not exploited. The exception is scenario 10 for the agents' pair (A, B).

Another issue that must be highlighted concerns the agents' pair (A, C): the alignment resulting from the union of the agents initial proposal ("union alignment") outperforms (with respect to f-measure) the alignment resulting from the intersection ("intersection alignment"). In all the other scenarios (1 to 8) the "intersection alignment" outperformed the "union alignment" and, therefore, the agreed alignments were closer or better than the "intersection alignment". However, in this case where the "union alignment" outperforms the "intersection alignment" the agreed alignments become closer to the "union alignment" instead of the "intersection alignment". This fact suggests that the agreed alignment tends to follow (and outperform) the best f-measure resulting from either the union or the intersection of the agents' initial proposal.

Considering the accuracy of the agreed alignments and the quantity and quality of the resolved conflicts, one may conclude that establishing H relations in the agents' private argumentation model is usefulness in the case where the opponent agent is able to reclassify the argument-instances exchanged based of that feature. This might be also seen by the agents as an indication to refine the community argumentation model as foreseen in the ANP.

8.5 Summary

The experiments demonstrate the effectiveness of the proposed TLAF/EAF-based approach through the following issues:

- It is possible to mimic the MbA/FDO approach in the TLAF/EAF-based approach by using the appropriate argument evaluation functions and constraining the argument instantiation process;
- By changing the argument evaluation functions only the TLAF/EAF-based approach outperforms the MbA/FDO approach both in number of (correctly) resolved conflicts and alignment accuracy;
- The agents ability to consider the dependency between correspondence improves the correctness of the resolved conflicts and the alignment accuracy;
- The argumentation process profits from the agents ability to privately extend the common argumentation model since it helps improve the alignment accuracy of the agents' initial proposal and, therefore, to reduce the amount of conflict to resolve;
- The ability to exchange arguments exploiting the H relation defined during the extension of the common argumentation model combined with the ability to reclassify the received argument fosters the persuasiveness of the agents while improving the correctness of the resolved conflicts and the alignment accuracy.

THIRD PART

Chapter 9

CONCLUSIONS

The initial and primary emphasis of the research presented in this document focuses on proposing and/or developing argument-based negotiation solutions enabling two or more agents wishing to resolve ontology matching divergences.

Despite that, as long as the research work evolved, the proposed contributions become generic (i.e. domain independent) and applicable to a set of other scenarios where conflict resolution is also required. In that sense, the emphasis on the conflicts about ontology matching correspondences is seen and became the scenario on which the generic contributions were applied, thus demonstrating the capability of the generic proposals to overcome the limitations identified in the state-of-the-art. Still, several other contributions arose from applying the generic contributions to this concrete scenario.

Consequently, conclusions on the applicability of the proposed research ideas must be drawn considering the dichotomy and the symbiosis between the generic contributions and domain specific contributions.

Even if a formal conclusion is difficult in this context, comparing the limitations of the state-of-the-art with the proposed contributions provides evidences concerning the validity of the thesis. This comparison is provided in section 9.2, but before that a summary of the contributions is presented.

9.1 Summary of Contributions

This section systematizes and summarizes the contributions made in the scope of this thesis:

- The Argument-based Negotiation Process (ANP):
    - Systematizes and organizes the phases of a negotiation based on arguments from the perspective of an agent and, therefore, the phases become task-oriented;
    - Clearly identifies the actors and their roles in the negotiation process and namely on which phases they act and interact;
    - Identifies the main blocks of data/information used as input and as output of the phases of the process;
    - Advocates an iterative and incremental flow of the results between the phases of the negotiation process;
    - Defines the notion of argumentation model and its purpose;
    - Advocates four characteristics to any argumentation model: sharable, reusable, extensible and modular;
    - Suggests and promotes the idea that the negotiation participants have distinct knowledge and perspectives about the domain application being argued;
    - Requires that despite the negotiation participants having distinct knowledge and perspectives about the domain application, a part of that knowledge is shared to foster the exchanged argument understanding;
- The Three-Layer Argumentation Framework (TLAF):
    - Suggests and promotes the idea of "explicit and formal specification of the agents' conceptualization of the arguments";
    - Is a less abstract argumentation framework than AF, BAF and VAF but suitable for representing many different situations without being committed to any domain of application;
    - Reduces the gap between argumentation systems and the abstract argumentation frameworks;
    - Satisfies the argumentation model purpose through its model layer;
    - Allows different models of the same argumentation application;

  - Constrains and drives the modeling effort during the process of capturing and representing the knowledge of a concrete argumentation scenario;
  - Through the information captured in the model layer the argumentation system is able to:
    - Constrain, simplify and validate the argument-instance generation;
    - Automatically infer the support and attack relationships between argument-instances;
  - Advocates and promotes the adoption of a widely accepted argument structure by argumentation systems;
  - Captures the application domain data by means of statements and reasoning mechanisms which permit a clear distinction and separation from the argumentation data;
  - Provides the ability to represent the information captured in the TLAF instance layer in an abstract formalism such as AF and BAF;
  - Argumentation systems adopting TLAF are still able to exploit the abstract argumentation frameworks in the same way and purpose they are used to (i.e. to infer about the arguments' acceptability);
- The Extensible Argumentation Framework (EAF):
  - Preserves all the TLAF characteristics;
  - Has/satisfies the extensibility and modularity features advocated by the thesis statement and the ANP;
- Argument-based Negotiation Approach for Ontology Matching (ANP-OM):
  - Adopts the ANP and its phases and, therefore, demonstrates its applicability in a concrete scenario;
  - Adopts either the TLAF or the EAF as the artifact that meets the notion of argumentation model and, therefore, demonstrates their suitability to represent a concrete scenario;
  - Defines the structure of the statements and reasoning mechanisms for the ontology matching negotiation scenario left open by the TLAF/EAF;
  - Identifies and defines the ANP actors in the scope of the ontology matching negotiation;

    - Allows negotiating participants to freely select their information source according to their preferences and interests;
    - Defines the content and the structure of the output of the Data Acquisition phase;
    - Has an intrinsic ability to evolve as long as the participants' knowledge of the matching process evolves;
- Argument Generation Process of ANP-OM:
    - Specifies an automatic configurable approach to generate argument-instances according to a given TLAF/EAF model;
    - Generates argument-instances from correspondences generated by third-part ontology matching algorithms;
    - Specifies a configurable approach to interpret the matchers' position about correspondences;
    - Allows different interpretation of the same correspondence provided by two different matchers and, therefore, give rise to distinct argument-instance;
    - Specifies a configurable approach based on a set of conditions:
        - Finds the premises of arguments even when the matcher is not able to justify the correspondences that give raise such arguments;
        - Detects and establish conflicts between statement-instances;
- Argument Evaluation Process of ANP-OM:
    - Grounds on a set of argument evaluation functions responsible for evaluating the arguments strength as suggested by the TLAF;
    - Identifies, describes and discusses three evaluation dimensions (i.e. strength, quantitative and qualitative dimensions);
    - Reuses and exploits the information of the arguments generation process to provide an initial strength for every argument-instance which is used to induce the evaluation functions;
    - Defines a set of principles/rules to generate the arguments initial strength;
    - Supports explicit cyclic and mutual dependency between argument-instances.

While the critical phases and processes of the ANP-OM have been extensively addressed along this thesis to demonstrate the applicability and usefulness of the proposed ideas, and namely to provide a clear perception (to the reader) of how the limitations found on the literature are overcome, several other relevant aspects/issues were slightly referred only or even not mentioned (e.g. implementation and performance issues). Despite their relevance, they are deemed trivial (e.g. implementation) or out of focus and thus less relevant (e.g. performance). Although, most of them are at least addressed by the described work.

9.2 Contributions and the State-of-the-Art Limitations

This section aligns the limitations identified for the state-of-the-art with the proposed contributions:

1. Collection and Classification of Matchers and Correspondences are overcome because:
   - Each agent is able to collect correspondences and other matching information from any sources it knows and trusts, instead of a single OMR;
   - Each agent has the internal ability to interpret distinctively the collected correspondences, through the interpretation functions. This means that the same correspondence may be interpreted differently by the agents;
2. Private arguments. The ANP-OM approach suggests and encourages agents to employ private arguments in their internal reasoning process by letting them privately extend the public argumentation model. The basis for overcoming this limitation is already present in TLAF, but the EAF provides specific constructs and semantics to fulfill this limitation;
3. Dependency between correspondences under negotiation is now possible because TLAF adopts the concept of intentional argument. Intentional argument is a full-fledged argument that also corresponds to an object under negotiation. This feature combined with the TLAF R relation allows capturing and exploiting the dependency between correspondences under negotiation;
4. Preferences. The limitations of the MbA/FDO preferences are overcome in TLAF by the modeling features and by the ability to apply arbitrarily complex, domain dependent or independent evaluation functions that will typically exploit the adopted argumentation model, namely the R relation;
5. Rebuttal, Undercut and Undermining arguments. Undercut and undermining arguments are allowed (even promoted) by the adoption of TLAF. TLAF allows arbitrary definition of R relations between arguments, thus forming arbitrary cyclic, graph-shaped structures encompassing undercut and undermining arguments;

6. Symmetric attacks must be explicitly defined in the TLAF model layer. As a consequence TLAF provides a modeling construct instead of a modeling constraint. Because in TLAF attacks are always unidirectional, TLAF models symmetric attacks by the combination of two unidirectional attacks;
7. Multiple preferred extensions. The multiple preferred extensions were not directly addressed, but this was indirectly addressed by the adoption of arbitrarily complex, domain dependent and independent evaluation functions. The presented experiments proved this claim.

Considering the previous exposition, it is believed that the proposed contributions overcome the identified limitations of the state-of-the-art on ontology matching argument-based negotiation approaches.

Additionally, the presented experiments proved that the proposed argument-based negotiation process performs better than the state-of-the-art on ontology matching argument-based negotiation approaches both quantitatively and qualitatively regarding the resolved conflicts and the accuracy of the agreed alignment. Moreover, the usefulness of the overcome limitations has been demonstrated.

9.3 Thesis Validation

Finally, aligning the thesis statement with the proposed contributions, allows argument supporting the validity of the thesis statement. This thesis advocates:

- "*the needs and benefits of adopting an explicit, formal and extensible specification of a shared argumentation model between argument-based negotiating agents*":

  Both EAF and TLAF adopt and promote the notion of "explicit, formal and extensible specification of a shared argumentation model", which simplifies and generalizes the design and development of argument-based negotiation scenarios.

- "*in order to resolve conflicts and achieve better agreements and in particular in the scope of ontology matching*".

  As the experiences demonstrated, the proposed contributions allow the achievement of better agreements though the results depend on many factors that are out of scope of this thesis (e.g. matchers, argumentation modeling methodologies and evaluation functions design).

Accordingly, even if future work is necessary in this subject, the proposed thesis is deemed valid.

Chapter 10

ONGOING AND FUTURE RESEARCH

During this thesis, as the restrictions were observed and discussed other research topics were mentioned. Based on that, this chapter describes four ongoing and future research directions:

- ANP-OM open issues;
- TLAF and EAF modeling methodologies;
- Combination of the relaxation and argumentation approaches on the ontology matching negotiation problem;
- Argumentation as a reasoning mechanism for ontology matching.

10.1 ANP-OM Open Issues

The proposed argument-based negotiation approach for the ontology matching scenario comprehends nine task-oriented phases. While a conceptual description and guidelines were provided for all phases of the ANP, concrete formal guidelines and algorithms were proposed only for the Instantiation and Evaluation phases. Given that, each phase is now revisited in order to identify the required but missing processes and methods.

10.1.1 Data Acquisition Phase

In the Data Acquisition phase, the main open question concerns the selection of the matching algorithms or matching agents from which an agent collects the correspondences to generate arguments and, therefore, to devise an alignment proposal. In fact, there is no easy answer to this pertinent question neither in this approach nor in the ontology matching domain. Currently, it is commonly accepted that matching algorithms/system should be selected based on two factors:

- The kind of data captured by the ontologies; and
- The problem to be solved.

Therefore, and according to [Shvaiko and Euzenat 2012], this question demands particular attention and research.

10.1.2 Instantiation Phase

Regarding the Instantiation phase, the proposed process is robust, well defined and consistent. However, it relies on the notion of interpretation function whose content is subjective and configurable. As that, and similarly to the Setup phase, expertise on ontology matching is required to pre-define such functions. Further, the improvement and refinement of such functions may be done dynamically based on the analysis of past experiences.

10.1.3 Evaluation Phase

The process proposed for the Evaluation phase relies on a set of evaluation functions whose ultimate output is a value stating which arguments hold and which do not hold. In this process, two open issues were identified:

- The first issue regards the concrete definition of the evaluation functions. While the evaluation dimensions have been identified and described, no research was done about their systematic application and combination. The definition of these evaluation functions may profit from utility functions and more concretely, on multi-attribute utility functions;
- The second issue regards the selection of a single preferred extension when more than one is possible/available. The adoption of strategies is a natural approach. The first step in that direction consists of identifying and characterizing the information on which the strategies will rely on. Yet, it is perceivable that while some of that information is orthogonal to many domains, other information exists and is relevant in some concrete domains only.

10.1.4 Agreement Attempt Phase

In the Agreement Attempt phase, each participant agent evaluates its level of satisfaction with the current candidate agreement leading the negotiation to the Persuasion phase or to the Settlement phase or even to end unsuccessfully. No contributions were made respecting this process during this thesis.

While an approach based on utility functions for measuring the pros and cons of each alternative is acceptable, an approach based on pragmatic reasoning focused on the interoperability task in hand exhibits great potential. While the adoption of strategies will complement the pragmatic reasoning approach, the combination of all these approaches show even greater potential, at the expense of complexity.

10.1.5 Persuasion Phase

In the Persuasion phase, the open issue concerns the selection of which and when arguments should be put forward to support the position taken and also to persuade the opponent to change their position. It is worth noticing that an argument can be put forward iif it belongs to the current preferred extension. Thus, the question here is more related to the attitude of the agent to resolve the divergences. An agent may be more or less (in a spectrum):

- Reactive or proactive: arguments are put forward proactively or just as a reaction of arguments put forward by the opponents;
- Concise or sparse: the agent is brief putting forwarding arguments (e.g. only the best argument at the time is put forward and the others are kept for another opportunity) or it puts forwards all know arguments.

Again, strategies are seen as an advisable and applicable approach.

10.1.6 Argumentation Model Refinement Phase

In the Argumentation Model Refinement phase, the underlying issues concern how the agents evolve and acquire new knowledge. This is seen as complementary process addressed in other fields of research [Kozareva and Hovy 2010; Bratko, Žabkar, and Možina 2009; Staab et al. 2002; Aberer et al. 2004].

10.2 TLAF/EAF model layers

The TLAF/EAF models developed in the scope of this thesis adopted empirical knowledge and competencies gained from many disciplines studied and applied through the years, namely:

- Adoption and modeling data structures;
- Knowledge specification through ontologies;
- Ontology matching.

While these competencies are common among informatics engineers and argumentation is a common sense human capability, argumentation modeling is not. Accordingly, it is necessary to adapt, devise and propose methodologies capable of guiding the argumentation modeling efforts for a diversity of stakeholders.

10.3 Combination of Argument-based and Relaxation-based Approaches

There are two types of approaches tackling the ontology matching negotiation problem: relaxation-based approaches and argument-based approach (cf. Chapter 3).

Assuming that some of the advantages of one are disadvantages of the other and vice-versa, it is foreseen that a hybrid negotiation approach might mitigate the disadvantages and enhance the advantages.

Yet, the phase(s) where the relaxation-based approach is applicable in scope of the ANP is still to be researched.

10.4 Argumentation as a Reasoning Mechanism

Argumentation was adopted in this research as a negotiation mechanism. However, argumentation can be also seen as a reasoning mechanism too, adopted by the agent to determine its beliefs (i.e. theoretical reasoning) and what to do (i.e. practical reasoning).

Thus, this section aims to expose some points concerning the ontology matching process in which the adoption of argumentation as a reasoning mechanism might be advisable.

With a few exceptions (see [Shvaiko et al. 2005; Deerwester et al. 2004]), most of the current matching algorithms and systems lack justifications or explanations to the proposed correspondences. However, having this ability matching systems can profit twofold:

- To gain a wider acceptance by users, since most of the alignments produced may not be intuitively obvious to human users;
- Since the users are able to understand the correspondences produced by the system (through the justifications), users can be involved in and grant benefits to the matching process [Zhdanova and Shvaiko 2006; McCann, Shen, and Doan 2008; Luz et al. 2012].

According to [Shvaiko and Euzenat 2008], the key issue is to represent explanations in a simple and clear way to the user in order to facilitate informed decision making. Yet, the standardization of explanations of correspondences would facilitate the interaction of matching systems with other programs. This includes scalability of visualization [Robertson, Czerwinski, and Churchill 2005] and better user interfaces in general, which are expected to bring big productivity gains.

The TLAF and EAF provide relevant modeling constructs (e.g. premises and the relationships between arguments) to capture and provide explanations in an iterative and incremental way (by walking through the tree/graph of arguments).

Another critical limitation on the state-of-the-art matching algorithms is that most of them are able to produce simple correspondences only. However, applications, such as Information Integration [Halevy et al. 2005] and E-Business [Fensel 2001], require alignments with more complex correspondences. Consequently, the ontology matching process is performed manually or semi-automatically at design time where the matching systems output serves as input information to the user only.

A potential approach to produce complex correspondences may rely on the adoption of matching patterns [Scharffe and Fensel 2008] and the conditions under which those patterns manifest themselves [Ritze, Völker, and Meilicke 2010] over the simple correspondences. As advocated in [Luz et al. 2012], the TLAF and EAF model layer seems to have the necessary flexibility to capture and represent those patterns and conditions. For that, several tasks are envisaged:

- To model the correspondence patterns and their dependencies from simple correspondences;
- To identify and distinguish between different types of statements to describe required expressions (e.g. correspondence statements and ontological statements);
- To identify and distinguish between different kinds of reasoning mechanisms to capture the semantics of the processes employed not only by matchers but also by analyzers (e.g. nominalization);
- To revise the defined statements' structure and reasoning mechanisms' structure to accomplish upcoming needs such as embed description logic or more expressive expressions (e.g. rules);
- To revise the arguments instantiation process proposed.

10.5 Outlook

Despite the contributions beyond the state-of-the-art proposed during this thesis, both ontology matching negotiation and argumentation mechanisms *per se* have considerable limitations and lack substantial features.

Accordingly, the ideas just proposed are seen as the starting point for future rich research.

FOURTH PART

Annex 1

DATA ACQUISITION'S INTERPRETATION FUNCTIONS

This annex complements the description of the experiments described previously in Chapter 8. In concrete, this annex details:

- The data acquisition process carried out by the three agents (A, B and C);
- The interpretation function (ψ) that enables the agents to generate arguments based on the acquired correspondences.

Since the data acquisition process relies on the GECAD Ontology Alignment System (GOALS) [Maio and Silva 2009b; Maio and Silva 2009a], this ontology matching tool is described first (section A 1.1). Further, section A 1.2 describes the GOALS configuration made for every agent together with the interpretation function adopted (ψ).

A 1.1 The Ontology Matching Tool

GOALS is a test-bed system developed to ease the creation of new matching systems. It promotes the reusability, combination and reconfiguration of existing matching techniques. Further, by exploiting existing methods and algorithms, it encourages the development of new matching algorithms for specific requirements.

GOALS has four core concepts:

- Data-entities: these represent any kind of data structures that any component manipulates as input and/or output. Currently, the system supports two different data-structure: (i) an "OntModel" which corresponds to one ontology and (ii) an "Alignment" (or "Matrix") which corresponds to a set of correspondences;
- Components: are objects acting as black boxes that play one or more roles in the ontology matching process (e.g. a matching algorithm, an alignment extractor, a threshold-based filter). Each component explicitly defines a set of data entities as inputs and outputs and a specific functionality. Particularities of each component are configurable through a set of parameters;
- Workflow specification: concerns the selection of the components taking part in the matching process, its parameters and roles and how the data entities flow between the components. The result of a workflow specification is a new complex matching algorithm;
- Execution engine: is the unit responsible for tackling a workflow specification and automatically executing it.

Two relevant features of the system are:

- The ability to encapsulate ontology matching tools developed externally as components. External tools are encapsulated by means of adapters [Gamma et al. 1994] that meet the GOALS API. In fact, most of the available components in GOALS are implemented through this feature;
- The ability to transform a workflow specification into a component that can be further reused into another workflow specification. These components are further referred to as workflow-based components.

All matching algorithms mentioned in the following section are available in GOALS through one of these two features.

A 1.2 Agents Configuration

The following sub-sections describe the matching algorithms used by each agent (A, B and C) and presents the adopted interpretation function (ψ) regarding the most specific argumentation model adopted by the agents.

A 1.2.1 Agent A

During the experiments, the most specific argumentation model adopted by agent A is EAF_A extending EAF_{DC} which in turn extends EAF_{FDO}. Figure A 1.1 partially and graphically depicts EAF_A as previously described in section 8.4. The elements filled in gray are those belonging to both EAF_{DC} and EAF_{FDO}.

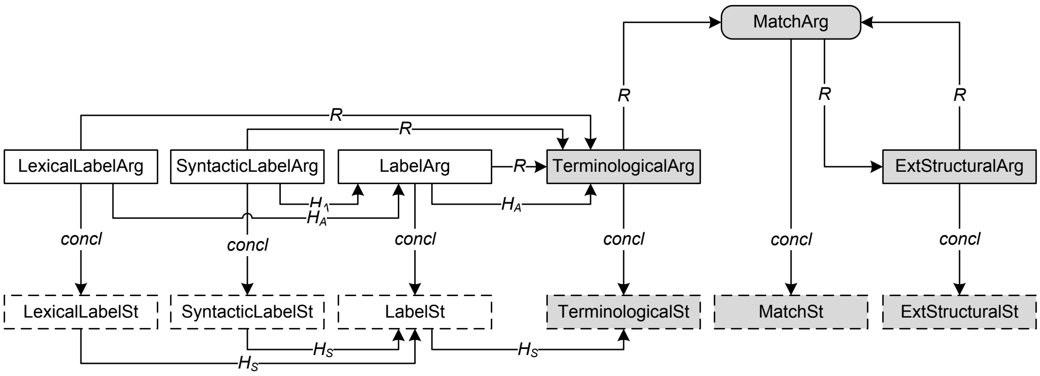

Figure A 1.1 – The most specific argumentation model used by agent A

These argumentation models were instantiated through the interpretation function (ψ) depicted in Table A 1.1. Notice that, correspondences are only collected from the mentioned matching algorithms when the argumentation model being used in the argumentation scenario contemplates the consequent statement-type.

Table A 1.1 – The interpretation function of agent A

Matcher	Correspondence Content			Statement Type	Reasoning Mechanism	tr_+	tr_-
	e	e'	r				
G_{A1}	any			LexicalLabelSt	Heuristic	1.00	1.00
G_{A2}	any			SyntacticalLabelSt	Heuristic	0.75	0.75
G_{A3}	any			LabelSt	Heuristic	0.70	0.70
G_{A4}	any			TerminologicalSt	Heuristic	0.80	0.80
G_{A5}	any			ExtStructuralSt	Heuristic	0.50	0.50
G_{A6}	any			MatchSt	Heuristic	0.70	0.70

In the argumentation scenarios 1 to 4 (cf. section 8.2) the matching algorithm G_{A6} was not used to instantiate *MatchSt*, but instead G_{A4}.

The matching algorithms mentioned in Table A 1.1 correspond to the following matching techniques:

- G_{A1} corresponds to the standard WordNet-based matching algorithm (WNMatcher) available in the CROSI Mapping System (CMS) [Kalfoglou et al. 2005];
- G_{A2} corresponds to the string-based matching algorithm available in the FALCON-AO [Jian et al. 2005];

- G_{A3} corresponds to the V-Doc [Qu, Hu, and Cheng 2006] matching algorithm also available in the FALCON-AO. It discovers correspondences by revealing the context of domain entities in ontologies;
- G_{A4} corresponds to the aggregation of the alignments generated by the matching algorithms G_{A1} and G_{A2} through the maximum function (graphically depicted in Figure A 1.2);

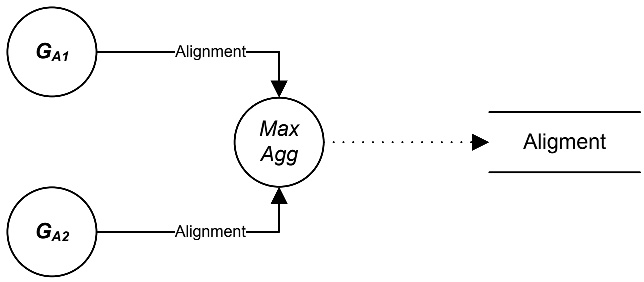

Figure A 1.2 – Graphical representation of the complex matching algorithm G_{A4}

- G_{A5} corresponds to the GMO [Hu et al. 2005] matching algorithm also available in the FALCON-AO. It measures the structural similarity of ontological entities;
- G_{A6} corresponds to the aggregation of the alignments generated by the matching algorithms G_{A1}, G_{A2} and G_{A3} through the maximum function (graphically depicted in Figure A 1.3).

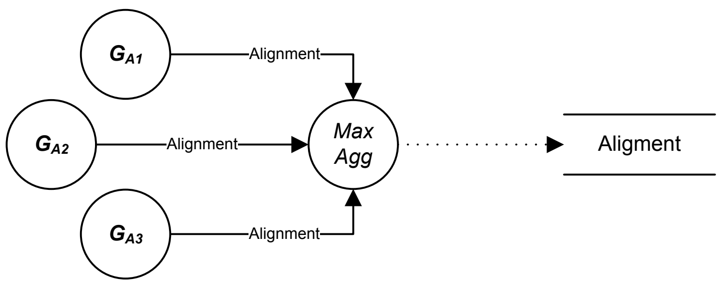

Figure A 1.3 – Graphical representation of the complex matching algorithm G_{A6}

A 1.2.2 Agent B

During the experiments, the most specific argumentation model adopted by agent B is EAF_B extending EAF_{DC} which in turn extends EAF_{FDO}. Figure A 1.4 partially and graphically depicts EAF_B as previously described in section 8.4. The elements filled in gray are those belonging to both EAF_{DC} and EAF_{FDO}.

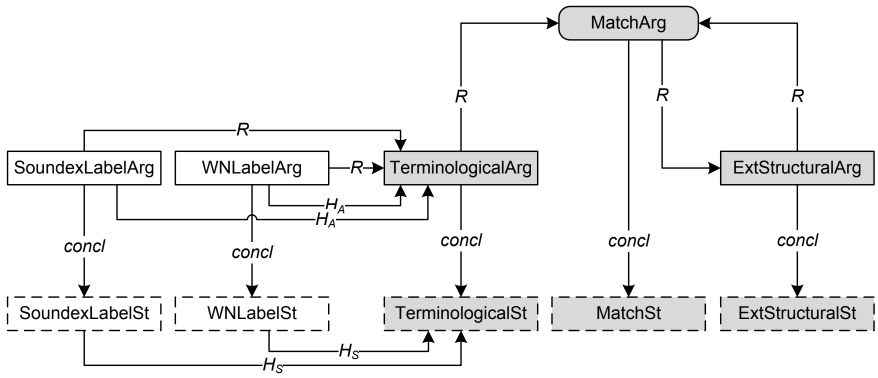

Figure A 1.4 – The most specific argumentation model used by agent B

These argumentation models were instantiated through the interpretation function (ψ) depicted in Table A 1.2. Notice that, correspondences are only collected from the mentioned matching algorithms when the argumentation model being used in the argumentation scenario contemplates the consequent statement-type.

Table A 1.2 – The interpretation function of agent B

Matcher	Correspondence Content			Statement Type	Reasoning Mechanism	tr_+	tr_-
	e	e'	r				
G_{B1}	any			SoundexLabelSt	Heuristic	0.75	0.75
G_{B2}	any			WNLabelSt	Heuristic	1.00	1.00
G_{B3}	any			TerminologicalSt	Heuristic	0.60	0.60
G_{B4}	any			ExtStructuralSt	Heuristic	0.70	0.70
G_{B5}	any			MatchSt	Heuristic	0.25	0.25

In the argumentation scenarios 1 to 4 (cf. section 8.2) the matching algorithm G_{B5} was not used to instantiate *MatchSt*, but instead G_{B4}.

The matching algorithms mentioned in Table A 1.2 correspond to the following matching techniques:

- G_{B1} corresponds to the string-based matching algorithm available in the SimMetrics project[18] that exploits the phonetic algorithm Soundex [Russell 1918; Russell 1922];
- G_{B2} corresponds to an improved WordNet-based matching algorithm (WNPlusMatcher) available in the CMS [Kalfoglou et al. 2005];
- G_{B3} corresponds to the aggregation of the alignments generated by the following matching algorithms:
    - G_{B1} and G_{B2} as described above;

[18] This project is available at http://sourceforge.net/projects/simmetrics/

   - G_{B6}, which corresponds to the string-based matching algorithm available in the SimPack [Bernstein et al. 2005] that exploits the frequency of substrings with length 2 in a given string (BiGram).

  The alignments generated by G_{B1}, G_{B2} and G_{B6} are aggregated through the OWA operator [Ji, Haase, and Qi 2008] (graphically depicted in Figure A 1.5);

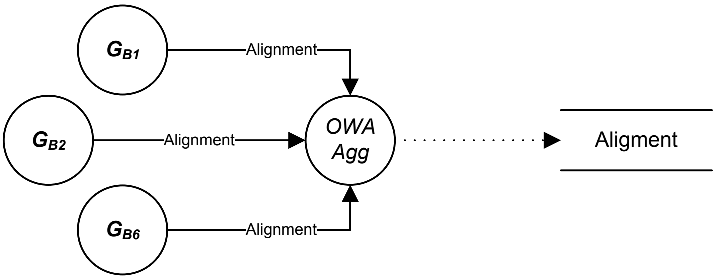

Figure A 1.5 – Graphical representation of the complex matching algorithm G_{B4}

- G_{B4} corresponds to the standard structure-based matching algorithm (StructureMatcher) available in the CMS [Kalfoglou et al. 2005];

- G_{B5} corresponds to the aggregation of the alignments generated by the following matching algorithms:

   - G_{B2} as described above;

   - G_{B7}, which corresponds to the string distance matching algorithm SMOA [Stoilos, Stamou, and Kollias 2005] available in the Alignment API implementation[19].

  The alignments generated by G_{B2} and G_{B7} are aggregated through the maximum function (graphically depicted in Figure A 1.6).

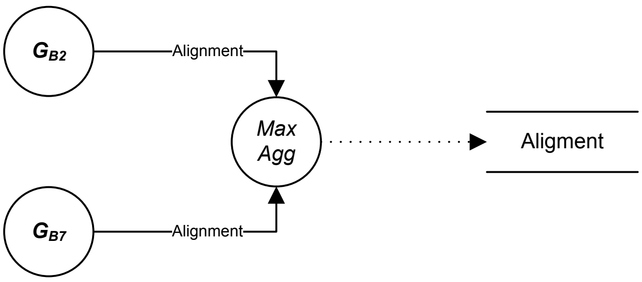

Figure A 1.6 – Graphical representation of the complex matching algorithm G_{B5}

[19] This Java API is available at http://alignapi.gforge.inria.fr/

A 1.2.3 Agent C

During the experiments, the most specific argumentation model adopted by agent C is EAF_C extending EAF_{DC} which in turn extends EAF_{FDO}. Figure A 1.7 partially and graphically depicts EAF_C as previously described in section 8.4. The elements filled in gray are those belonging to both EAF_{DC} and EAF_{FDO}.

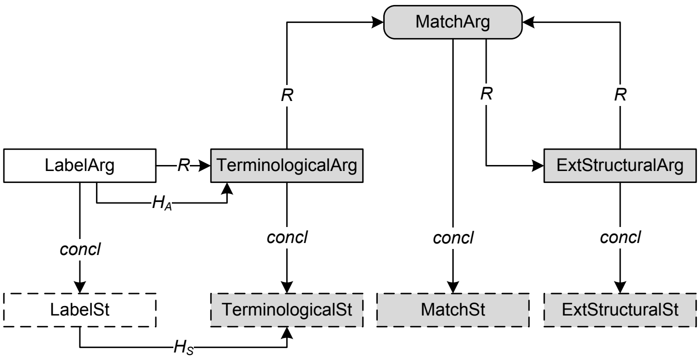

Figure A 1.7 – The most specific argumentation model used by agent C

These argumentation models were instantiated through the interpretation function (ψ) depicted in Table A 1.3. Notice that, correspondences are only collected from the mentioned matching algorithms when the argumentation model being used in the argumentation scenario contemplates the consequent statement-type.

Table A 1.3 – The interpretation function of agent C

Matcher	Correspondence Content			Statement Type	Reasoning Mechanism	tr_+	tr_-
	e	e'	r				
G_{C1}	any			LabelSt	Heuristic	0.75	0.75
G_{C2}	any			LabelSt	Heuristic	1.00	1.00
G_{C3}	any			TerminologicalSt	Heuristic	0.70	0.70
G_{C4}	any			ExtStructuralSt	Heuristic	0.80	0.80
G_{C5}	any			MatchSt	Heuristic	0.25	0.25

In the argumentation scenarios 1 to 4 (cf. section 8.2) the matching algorithm G_{C5} was not used to instantiate *MatchSt*, but instead G_{C4}.

The matching algorithms mentioned in Table A 1.3 correspond to the following matching techniques:

- G_{C1} corresponds to the Levenshtein string distance matching algorithm [Levenshtein 1965] available in the SimMetrics project;
- G_{C2} corresponds to the matching algorithm G_{B2} (WNPlusMatcher) as described previously for agent B;

- G_{C3} corresponds to the aggregation of the alignments generated by the following matching algorithms:
    - G_{C1} and G_{C2} as described above;
    - G_{C6}, which corresponds to the matching algorithm G_{B7} (SMOA) as described previously for agent B;

  The alignments generated by G_{C1}, G_{C2} and G_{C6} are aggregated through the linear average function (graphically depicted in Figure A 1.8);

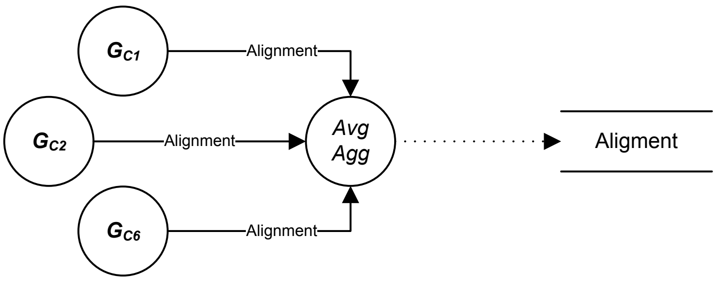

Figure A 1.8 – Graphical representation of the complex matching algorithm G_{C3}

- G_{C4} corresponds to the aggregation of the alignments generated by the following matching algorithms:
    - G_{C6} as described above;
    - G_{C7}, which corresponds to the output of an improved structure-based matching algorithm (StructurePlusMatcher) available in the CMS [Kalfoglou et al. 2005] (referred to as G_{C8}) filtered by the Hungarian method [Munkres 1957] to global optimize the generated alignment (graphically depicted in Figure A 1.9). The resulting alignment is injective (cf. section 2.2.1);

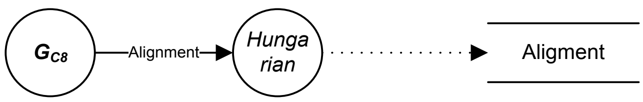

Figure A 1.9 – Graphical representation of the complex matching algorithm G_{C7}

  The alignments generated by G_{C6} and G_{C7} are aggregated through the linear average function (graphically depicted in Figure A 1.10);

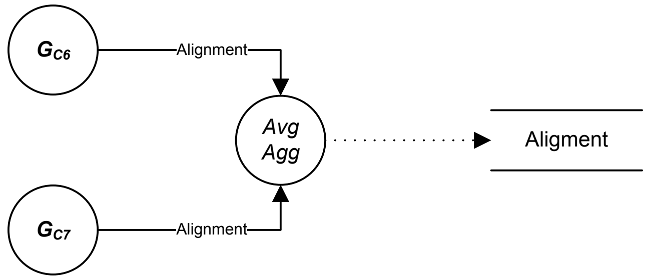

Figure A 1.10 – Graphical representation of the complex matching algorithm G_{C4}

- G_{C5} corresponds to the aggregation of the alignments generated by the matching algorithms G_{C2}, G_{C6} and G_{C7} through the maximum function. The resulting alignment is further global optimized by applying the Hungarian method [Munkres 1957] (graphically depicted in Figure A 1.11).

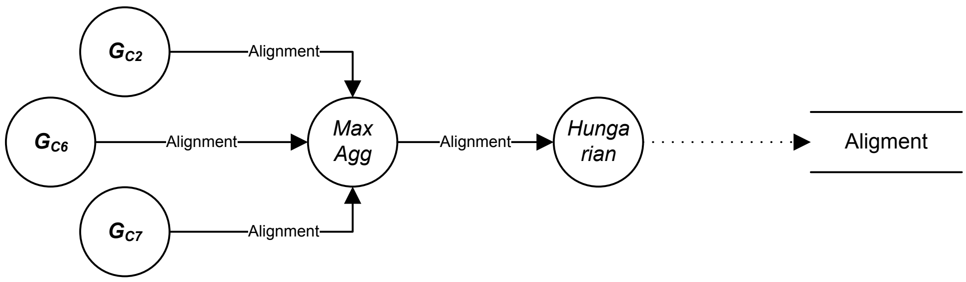

Figure A 1.11 – Graphical representation of the complex matching algorithm G_{C5}

BIBLIOGRAPHY

Aberer, K., P. Cudré-Mauroux, A. Catarci, M. Hacid, A. Illarramendi, V. Kashyap, M. Mecella, E. Mena, and J. Neuhold. 2004. '*Emergent semantics principles and issues*'. In Database Systems for Advanced Applications, LNCS 2973: 25-38. Springer.

Ackoff, R. 1989. '*From Data to Wisdom*'. In Journal of Applied System Analysis 16(1): 3-9.

Agrawal, R., and R. Srikant. 2001. '*On integrating catalogs*'. In Proc. 10th International World Wide Web Conference (WWW), 603-612. Hong Kong (CN).

Amgoud, L., C. Cayrol, M. Lagasquie-Schiex, and P. Livet. 2008. '*On bipolarity in argumentation frameworks*'. In International Journal of Intelligent Systems, 23(10): 1062-1093. Wiley.

Baader, F. 2003. '*The description logic handbook: theory, implementation, and applications*'. Cambridge University Press.

Bailin, S., and W. Truszkowski. 2003. '*Ontology Negotiation: How Agents Can Really Get to Know Each Other*'. In Innovative Concepts for Agent-Based Systems, LNCS 2564: 320-334. Springer.

Baldauf, M., S. Dustdar, and F. Rosenberg. 2007. '*A survey on context-aware systems*'. In International Journal of Ad Hoc and Ubiquitous Computing, 2(4): 263-277. Inderscience Enterprises Ltd.

Baroni, P., and M. Giacomin. 2009. '*Semantics of Abstract Argument Systems*'. In Argumentation in Artificial Intelligence, 25-44. Springer.

Batini, C., M. Lenzerini, and S. Navathe. 1986. '*A Comparative Analysis of Methodologies for Database Schema Integration*'. In ACM Computing Surveys, 18(4): 323-364. ACM.

Bellahsene, Z., A. Bonifati, and E. Rahm. 2011. '*Schema Matching and Mapping*'. Springer.

Bellinger, G., D. Castro, and A. Mills. 2004. '*Data, information, knowledge, and wisdom*'. http://www.systems-thinking.org/dikw/dikw.htm.

Bench-Capon, T. 2003. '*Persuasion in Practical Argument Using Value-based Argumentation Frameworks*'. In Journal Logic Computation, 13(3): 429-448. Oxford University Press.

Benerecetti, M., P. Bouquet, and C. Ghidini. 2001. '*On the dimensions of context dependence: partiality, approximation, and perspective*'. In Modeling and Using Context, LNCS 2116: 59-72. Springer.

Beneventano, D., S. Bergamaschi, F. Guerra, and M. Vincini. 2001. '*The MOMIS Approach to Information Integration*'. In Proc. International Conference on Enterprise Information Systems, 194-198. Setúbal (PT).

Berners-Lee, T., and M. Fischetti. 2000. '*Weaving the Web The Original Design and Ultimate Destiny of the World Wide Web*'. HarperInformation.

Berners-Lee, T., J. Hendler, and O. Lassila. 2001. '*The Semantic Web*'. In Scientific American, 284(5): 34-43. Scientific American.

Bernstein, A, E Kaufmann, C Kiefer, and C Burki. 2005. '*SimPack: A Generic Java Library for Similarity Measures in Ontologies*'. Technical Report, University of Zurich, Department of Informatics. Zurich (CH).

Bernstein, P., A. Halevy, R. Pottinger, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. 2002. '*Data Management for Peer-to-Peer Computing: A Vision*'. In Proc. 5th International Workshop on the Web and Databases (WebDB), 29(4): 55-63. Madison (WS, US).

Bernstein, P., and E. Rahm. 2000. '*Data warehouse scenarios for model management*'. In Proc. 19th International Conference on Conceptual Modeling (ER), 1-15. Salt Lake City (UT, US).

Bouquet, P., M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, L. Serafini, G. Stamou, Y. Sure, and S. Tessaris. 2004. '*Specification of a common framework for characterizing alignment*'. Deliverable: KWEB/2004/D2.2.1/v1.2.

Bratko, I., J. Žabkar, and M. Možina. 2009. '*Argument-Based Machine Learning*'. In Argumentation in Artificial Intelligence: 463-482. Springer.

Bratman, M. 1987. '*Intention, Plans and Practical Reason*'. Cambridge University Press.

Bratman, M., D. Israel, and M. Pollack. 1988. '*Plans and resource-bounded practical reasoning*'. In Computational Intelligence, 4(3): 349-355. Blackwell Publishing Ltd.

Buckingham, S., E. Motta, and J. Domingue. 2000. '*ScholOnto: an ontology-based digital library server for research documents and discourse*'. In International Journal on Digital Libraries, 3(3): 237-248. Springer.

Buckingham, S., V. Uren, G. Li, B. Sereno, and C. Mancini. 2007. '*Modeling naturalistic argumentation in research literatures: Representation and interaction design issues*'. In International Journal of Intelligent Systems, 22(1): 17-47. Wiley.

Caminada, M., and L. Amgoud. 2007. '*On the evaluation of argumentation formalisms*'. In Journal of Artificial Intelligence, 171(5-6): 286-310. Elsevier.

Castano, S., V. De Antonellis, and S. di Vimercati. 2000. '*Global Viewing of Heterogeneous Data Sources*'. In Transactions on Knowledge and Data Engineering, 13(2): 277-297. IEEE.

Cayrol, C., and M. Lagasquie-Schiex. 2005a. '*On the Acceptability of Arguments in Bipolar Argumentation Frameworks*'. In Symbolic and Quantitative Approaches to Reasoning with Uncertainty, LNAI 3571: 378-389. Springer.

Cayrol, C., and M. Lagasquie-Schiex. 2005b. '*Gradual Valuation for Bipolar Argumentation Frameworks*'. In Symbolic and Quantitative Approaches to Reasoning with Uncertainty, LNAI 3571: 366-377. Springer.

Cayrol, C., and M. Lagasquie-Schiex. 2010. '*Coalitions of arguments: A tool for handling bipolar argumentation frameworks*'. In International Journal of Intelligent Systems, 25(1): 83-109. Wiley.

Chawathe, S., H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. 1994. '*The TSIMMIS Project: Integration of heterogeneous information sources*'. In Proc. 16th Meeting of the Information Processing Society of Japan (IPSJ), 7-18. Tokyo (JP).

Chesñevar, C., J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. 2006. '*Towards an Argument Interchange Format*'. In Knowledge Engineering Review, 21(4): 293-316. Cambridge University Press.

Cody, W., J. Kreulen, V. Krishna, and S. Spangler. 2002. '*The integration of business intelligence and knowledge management*'. In IBM Systems Journal, 41(4): 697-713. IBM.

David, J., J. Euzenat, F. Scharffe, and C. Trojahn. 2011. '*The Alignment API 4.0*'. In Semantic Web Journal, 2(1): 3-10. IOS Press.

Dean, M. 2004. '*OWL Web Ontology Language Reference*'. http://www.w3.org/TR/owl-ref/.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, R. Harshman, R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. 2004. '*iMAP: Discovering Complex Semantic Matches between Database Schemas*'. In Proc. 23rd International Conference on Management of Data (SIGMOD), 41(6): 383-394. Paris (FR).

Do, H., and E. Rahm. 2002. '*COMA — A System for Flexible Combination of Schema Matching Approaches*'. In Proc. 28th International Conference on Very Large Data Bases (VLDB), 610-621. Hong Kong (CN).

Doan, A., J. Madhavan, P. Domingos, and A. Halevy. 2004. '*Ontology matching: A machine learning approach*'. Handbook on Ontologies. Springer.

Doran, P., T. Payne, V. Tamma, and I. Palmisano. 2010. '*Deciding Agent Orientation on Ontology Mappings*'. In Semantic Web - ISWC 2010, LNCS 6496: 161-176. Springer.

Draper, D., A. Halevy, and D. Weld. 2001. '*The nimble integration engine*'. In Proc. 20th International Conference on Management of Data (SIGMOD), 567-568. Santa Barbara (CA, US).

Dung, P. 1995. '*On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*'. In Journal of Artificial Intelligence, 77(2): 321-357. Elsevier.

Ehrig, M., and J. Euzenat. 2005. '*Relaxed Precision and Recall for Ontology Matching*'. In Proc. K-CAP Workshop on Integrating Ontologies, 25-32. Banff (CA).

Ehrig, M., and Y. Sure. 2004. '*Ontology mapping - an integrated approach*'. In Semantic Web: Research and Applications, LNCS 3053:76-91. Springer.

Ehrig, M., and Y. Sure. 2005. '*FOAM - Framework for Ontology Alignment and Mapping; Results of the Ontology Alignment Initiative*'. In Proc. K-CAP Workshop on Integrating Ontologies, 156: 72-76. Banff (CA).

Eijk, R., F. de Boer, W. Hoek, and J. Meyer. 2001. '*On dynamically generated ontology translators in agent communication*'. In International Journal of Intelligent Systems, 16(5): 587-607. Wiley.

Euzenat, J. 2003. '*Towards composing and benchmarking ontology alignments*'. In Proc. ISWC Workshop on Semantic Integration, 165-166. Sanibel Island (FL, US).

Euzenat, J. 2004. '*An API for Ontology Alignment*'. In Semantic Web - ISWC 2004, LNCS 3298: 698-712. Springer.

Euzenat, J. 2007. '*Semantic Precision and Recall for Ontology Alignment Evaluation*'. In Proc. 20th International Joint Conference on Artificial Intelligence. Hyderabad (IN).

Euzenat, J. 2008. '*Algebras of Ontology Alignment Relations*'. In Semantic Web - ISWC 2008, LNCS 5318: 387-402. Springer.

Euzenat, J., F. Scharffe, and A. Zimmerman. 2007. '*D2.2.10: Expressive Alignment Language and Implementation*'. Deliverable: KWEB/2004/D2.2.10/1.0.

Euzenat, J., and P. Shvaiko. 2007. '*Ontology Matching*'. Springer.

Euzenat, J., and H. Stuckenschmidt. 2003. '*The `family of languages' approach to semantic interoperability*'. In Knowledge Transformation for the Semantic Web, 95: 49-63. IOS Press.

Euzenat, J., and P. Valtchev. 2004. '*Similarity-based ontology alignment in OWL-Lite*'. In Proc. 16th European Conference on Artificial Intelligence (ECAI), 333-337. Valencia (ES).

Falappa, M., G. Kern-Isberner, and G. Simari. 2009. '*Belief revision and argumentation theory*'. In Argumentation in Artificial Intelligence: 341-360.Springer.

Fellbaum, C. 1998. '*WordNet: an electronic lexical database*'. MIT Press.

Fensel, D. 2001. '*Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*'. Springer.

Fensel, D., and FIPA 2002. '*FIPA ACL Communicative Act Library Specification*'. Springer.

Fensel, D., F. Harmelen, Y. Ding, M. Klein, H. Akkermans, J. Kampman, U. Davies, and V. Gothenburg. 2002. '*On-To-Knowledge: Semantic Web Enabled Knowledge Management*'. IEEE Computer.

Finin, T., Y. Labrou, and J. Mayfield. 1997. '*KQML as an Agent Communication Language*'. In Software agents, 291-316. MIT Press.

FIPA00037. 2002. '*Communicative Act Library Specification*'. Standard SC00037J. IEEE Foundation for Intelligent Physical Agents.

FIPA00061. 2002. '*FIPA ACL Message Structure Specification*'. Standard SC00061G. IEEE Foundation for Intelligent Physical Agents.

Gabriel, L., H. Martins, P. Maio, and N. Silva. 2008. '*Ontology Mapping Systematization, Negotiation and Evolution*'. Technical Report: EDGAR-PR-2008-01.GECAD – ISEP. Porto (PT).

Gale, D., and L. Shapley. 1962. '*College Admissions and the Stability of Marriage*'. American Mathematical Monthly, 69(1): 5-15. Mathematical Association of America.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1994. '*Design Patterns: Elements of Reusable Object-Oriented Software*'. Addison-Wesley Professional.

Genesereth, M., and R. Fikes. 1992. '*Knowledge Interchange Format Version 3 Reference Manual*'. Technical Report Logic-92-1. Stanford (CA, US).

Gilson, O., N. Silva, P. Grant, and M. Chen. 2008. '*From web data to visualization via ontology mapping*'', In Computer Graphics Forum, 27(3): 959-966. Blackwell Publishing Ltd.

Giunchiglia, F., P. Shvaiko, and M. Yatskevich. 2005. '*Semantic schema matching*'. In On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. LNCS 3761: 347-365. Springer.

Goguen, J. 1999. '*An introduction to algebraic semiotics, with application to user interface design*'. In Computation for Metaphor, Analogy, and Agents, LNCS 1562: 242-291. Springer.

Gordon, T., H. Prakken, and D. Walton. 2007. '*The Carneades model of argument and burden of proof*'. In Artificial Intelligence, 171: 875-896. Elsevier.

Gruber, T.. 1993. '*A translation approach to portable ontology specifications*'. In Journal of Knowledge Acquisition, 5(2): 199-220. Academic Press.

Guha, R., and B. McBride. 2004. '*RDF Vocabulary Description Language 1.0: RDF Schema*'. http://www.w3.org/TR/rdf-schema/

Halevy, A. 2001. '*Answering queries using views: a survey*'. In International Journal on Very Large Data Bases, 10(4): 270-294. Springer.

Halevy, A., N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. 2005. '*Enterprise information integration: successes, challenges and controversies*'. In Proc. 24th International Conference on Management of Data, 778-787. Baltimore (MD, US).

Hey, J. 2004. '*The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link*'. Intergovernmental Oceanographic Commission (UNESCO).

Hogger, C. 1990. '*Essentials of logic programming*'. Oxford University Press.

Horrocks, I., P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. '*SWRL: a semantic web rule language combining OWL and RuleML*'. W3C Member Submission.

Hu, W., N. Jian, Y. Qu, and Q. Wang. 2005. '*GMO: A Graph Matching for Ontologies*'. In Proc. K-CAP Workshop on Integrating Ontologies, 43-50. Banff (CA).

Ichise, R., H. Takeda, and S. Honiden. 2003. '*Integrating Multiple Internet Directories by Instance-based Learning*'. In Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), 22-30. Acapulco (MX).

Isaac, A., C. Trojahn, S. Wang, and P. Quaresma. 2008. '*Using quantitative aspects of alignment generation for argumentation on mappings*'. In Proc. ISWC'08 Workshop on Ontology Matching. Karlsruhe (DE).

Ji, Q., P. Haase, and G. Qi. 2008. '*Combination of Similarity measures in Ontology Matching using the OWA Operator*'. In Proc. 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems (IPMU'08). Málaga (ES).

Jian, N., W. Hu, G. Cheng, and Y. Qu. 2005. '*Falcon-AO: Aligning Ontologies with Falcon*'. In Proc. K-CAP Workshop on Integrating Ontologies, 87-93. Banff (CA).

Kalfoglou, Y., B. Hu, N. Shadbolt, and D. Reynolds. 2005. '*CROSI - Capturing Representing and Operationalising Semantic Integration*'. http://www.aktors.org/crosi/.

Karacapilidis, N., and D. Papadias. 2001. '*Computer Supported Argumentation and Collaborative Decision Making: The Hermes System*'. In Information Systems, 26(4): 259-277. Elsevier.

Kirschner, P., S. Shum, and C. Carr. 2003. '*Visualizing argumentation: Software tools for collaborative and educational sense-making*'. Springer.

Klein, M. 2001. '*Combining and relating ontologies: an analysis of problems and solutions*'. In Proc. International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Ontologies and Information Sharing, 53-62. Seattle (WA, US).

Kozareva, Z., and E. Hovy. 2010. '*Learning arguments and supertypes of semantic relations using recursive patterns*'. In Proc. 48th Annual Meeting of the Association for Computational Linguistics, 1482-1491. Uppsala (SE).

Laera, L., I. Blacoe, V. Tamma, T. Payne, J. Euzenat, and T. Bench-Capon. 2007. '*Argumentation over Ontology Correspondences in MAS*'. In 6th International Joint Conference on Autonomous Agents and Multiagent Systems, 228-235. Honolulu (HI, US).

Levenshtein, V. 1965. '*Binary codes capable of correcting deletions, insertions, and reversals*'. In Doklady akademii nauk SSSR, 163(4): 845-848.

Lomuscio, A., M. Wooldridge, and N. Jennings. 2003. '*A Classification Scheme for Negotiation in Electronic Commerce*'. In Group Decision and Negotiation, 12(1): 31-56. Springer.

Lopez, V., E. Motta, and V. Uren. 2006. '*PowerAqua: Fishing the Semantic Web*'. In Semantic Web: Research and Applications, LNCS 4011: 393-410. Springer.

Luz, N., N. Silva, P. Maio, and P. Novais. 2012. '*Ontology Alignment through Argumentation*'. In AAAI Spring Symposium on Wisdom of the Crowd. Stanford (CA, US). (to appear).

Macagno, F., G. Rowe, C. Reed, and D. Walton. 2006. '*Araucaria as a tool for diagramming arguments in teaching and studying philosophy*'. In Teaching Philosophy, 29(2): 111-124. SSRN.

Mack, R., Y. Ravin, and R. Byrd. 2001. '*Knowledge portals and the emerging digital knowledge workplace*'. In IBM Systems Journal, 40(4): 925-955. IBM.

Madhavan, J., P. Bernstein, and E. Rahm. 2001. '*Generic Schema Matching with Cupid*'. In 27th International Conference in Very Large Databases, 49-58. Rome (IT).

Maedche, A., B. Motik, N. Silva, and R. Volz. 2002. '*MAFRA - A Mapping Framework for Distributed Ontologies*'. In Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, LNCS 2473: 69-75. Springer.

Maio, P., N. Bettencourt, N. Silva, and J. Rocha. 2006. '*Building Consensus on Ontology Mapping*'. In Proc. 5th International Semantic Web Conference, 1-2. Athens (GA, US).

Maio, P., N. Bettencourt, N. Silva, and J. Rocha. 2007. '*Evaluating a confidence value for ontology alignment*'. In Proc. Ontology Matching Workshop on the Sixth International Semantic Web Conference, 281-285. Busan (KR).

Maio, P., and N. Silva. 2009a. '*GOAlS - Gecad Ontology Alignment System*'. http://www.dei.isep.ipp.pt/~pmaio/goals/.

Maio, P., and N. Silva. 2009b. '*GOALS - A test-bed for ontology matching*'. In Proc. 1st IC3K International Conference on Knowledge Engineering and Ontology Development (KEOD), 293-299. Funchal (PT).

Maio, P., and N. Silva. 2010. '*Ontology alignment argumentation with mutual dependency between arguments and mappings*'. In Proc. International Workshop on Data Engineering meets the Semantic Web (DESWeb) at 26th IEEE International Conference on Data Engineering (ICDE), 233-238. Long Beach (CA, US).

Maio, P., and N. Silva. 2011a. '*TLAF Meta-Model Layer as an Ontology*'. http://www.dei.isep.ipp.pt/~pmaio/TLAF/Ontology/TLAF_Ontology.owl.

Maio, P., and N. Silva. 2011b. '*A Three-Layer Argumentation Framework*'. In Proc. 1st International Workshop on the Theory and Applications of Formal Argumentation (TAFA) at IJCAI, 186-200. Barcelona (ES).

Maio, P., N. Silva, and J. Cardoso. 2011a. '*EAF-based Negotiation Process*'. In Proc. 4th International Workshop on Agent-based Complex Automated Negotiation (ACAN) at AAMAS, 89-92. Taipei (TW).

Maio, P., N. Silva, and J. Cardoso. 2011b. '*Generating Arguments for Ontology Matching*'. In Proc. 10th International Workshop on Web Semantics at DEXA, 239-243. Toulouse (FR).

Makhoul, J., F. Kubala, R. Schwartz, and R. Weischedel. 1999. '*Performance measures for information extraction*'. In Proc. DARPA Broadcast News Workshop, 249-252. Herndon (VA, US).

Marwick, A. 2001. '*Knowledge management technology*'. In IBM Systems Journal, 40(4): 814-830. IBM.

McCann, R., W. Shen, and A. Doan. 2008. '*Matching schemas in online communities: A web 2.0 approach*'. In Proc. 24th International Conference Data Engineering (ICDE), 110-119. Cancun (MX).

Meilicke, C., and H. Stuckenschmidt. 2008. '*Incoherence as a basis for measuring the quality of ontology mappings*'. In Proc. 3rd International Workshop on Ontology Matching, 1-12. Karlsruhe (DE).

Melnik, S., E. Rahm, and P. Bernstein. 2003. '*Rondo: A Programming Platform for Model Management*'. In Proc. 22nd International Conference on Management of Data (SIGMOD), 193-204. San Diego (CA, US).

Mena, E., V. Kashyap, A. Sheth, and A. Illarramendi. 1996. '*Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies*'. In Proc. 4th International Conference on Cooperative Information Systems (CoopIS), 14-25. Brussels (BE).

Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. '*Introduction to WordNet: An on-line lexical database*'. In Journal of Lexicography, 3(4): 235-244. Oxford University Press.

Moran, R. 2001. '*Authority and Estrangement: An Essay on Self-Knowledge*'. Princeton University Press.

Munkres, J. 1957. '*Algorithms for the Assignment and Transportation Problems*'. In Journal of the Society for Industrial and Applied Mathematics, 5(1): 32-38. Society for Industrial and Applied Mathematics.

Neches, R., R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. 1991. '*Enabling technology for knowledge sharing*'. In AI Magazine, 12(3): 36-56. AAAI.

Ngo, D., Z. Bellahsene, R. Coletta, and et. al. 2011. '*A flexible system for ontology matching*'. In Proc. International Conference on Advanced Information Systems Engineering, 73-80. London (UK).

OAEI. 2011. '*Ontology Alignment Evaluation Initiative. 2011 Campaign*'. http://oaei.ontologymatching.org/2011/.

Ouksel, A., and I. Ahmed. 1999. '*Ontologies are not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction*'. In Distributed and Parallel Databases, 7(1): 7-35. Springer.

Paglieri, F., and C. Castelfranchi. 2005. '*Arguments as belief structures: Towards a Toulmin layout of doxastic dynamics?*' In Proc. Uses of Argument (OSSA), 356-367. Hamilton (CA).

Parent, C., and S. Spaccapietra. 1998. '*Issues and approaches of database integration*'. In Communications of the ACM, 41(5): 166-178. ACM.

Pollock, J. 1967. '*Criteria and Our Knowledge of the Material World*'. In Philosophical Review, 76(1): 28-60. Duke University Press.

Pollock, J. 1971. '*Perceptual Knowledge*'. In Philosophical Review, 80(3): 287-319. Duke University Press.

Pollock, J. 1987. '*Defeasible reasoning*'. In Cognitive Science, 11: 481-518. Elsevier.

Pollock, J. 1994. '*Justification And Defeat*'. In Artificial Intelligence, 67: 377-407. Elsevier.

Prakken, H. 2010. '*An abstract framework for argumentation with structured arguments*'. In Argument and Computation, 1(2): 93-124. Taylor and Francis.

Qu, Y., W. Hu, and G. Cheng. 2006. '*Constructing virtual documents for ontology matching*'. In Proc. 15th International Conference on World Wide Web, 23-31. Edinburgh (UK).

Rahm, E., and P. Bernstein. 2001. '*A survey of approaches to automatic schema matching*'. In International Journal on Very Large Data Bases, 10(4): 334-350. Springer.

Rahwan, I., and B. Banihashemi. 2008. '*Arguments in OWL: A Progress Report*'. In Proc. International Conference on Computational Models of Argument (COMMA), 297-310. Amsterdam (NL).

Rahwan, I., B. Banihashemi, C. Reed, D. Walton, and S. Abdallah. 2011. '*Representing and classifying arguments on the semantic web*'. In Knowledge Engineering Review, 26(4): 487-511. Cambridge University Press.

Rahwan, I., and C. Reed. 2009. '*The Argument Interchange Format*'. In Argumentation in Artificial Intelligence, 383-402. Springer.

Rescher, N. 1977. '*Dialectics a controversy-oriented approach to the theory of knowledge*'. SUNY Press.

Ritze, D., J. Völker, and C. Meilicke. 2010. '*Linguistic Analysis for Complex Ontology Matching*'. In Proc. 5th International Workshop on Ontology Matching, 1-12. Shanghai (CN).

Robertson, G., M. Czerwinski, and J. Churchill. 2005. '*Visualization of mappings between schemas*'. In Proc. SIGCHI Conference on Human factors in computing systems, 431-439. Portland (OR, US).

Rousset, M., P. Adjiman, P. Chatalic, F. Goasdoué, and L. Simon. 2006. '*SomeWhere in the Semantic Web*'. In SOFSEM 2006: Theory and Practice of Computer Science, LNCS 3831: 84-99. Springer.

Russell, R. 1918. '*US Patent 1261167 (A)*'.

Russell, R. 1922. '*US Patent 1435663 (A)*'.

Sabou, M., V. Lopez, and E. Motta. 2006. '*Ontology Selection for the Real Semantic Web: How to Cover the Queen's Birthday Dinner?*' In Managing Knowledge in a World of Networks, LNCS 4248: 96-111. Springer.

Saruladha, K., G. Aghila, and B. Sathiya. 2011. '*A Comparative Analysis of Ontology and Schema Matching Systems*'. In International Journal of Computer Applications, 34(8): 14-21. FCS.

Scharffe, F. 2011. '*EDOAL: Expressive and Declarative Ontology Alignment Language*'. http://alignapi.gforge.inria.fr/edoal.html.

Scharffe, F., and D. Fensel. 2008. '*Correspondence patterns for ontology alignment*'. In Knowledge Engineering: Practice and Patterns, LNCS 5268: 83-92. Springer.

Sheth, S., and J. Larson. 1990. '*Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases*'. ACM Computing Surveys, 22(3): 183-236. ACM.

Shvaiko, P., and J. Euzenat. 2005. '*A survey of schema-based matching approaches*'. In Journal on Data Semantics IV, LNCS 3730: 146-171. Springer.

Shvaiko, P., and J. Euzenat. 2008. '*Ten Challenges for Ontology Matching*'. In On the Move to Meaningful Internet Systems: OTM 2008, LNCS 5332: 1164-1182. Springer.

Shvaiko, P., and J. Euzenat. 2012. '*Ontology Matching: State of the Art and Future Challenges*'. In Transactions on Knowledge and Data Engineering, (to appear). IEEE.

Shvaiko, P., F. Giunchiglia, P. Silva, and D. McGuinness. 2005. '*Web Explanations for Semantic Heterogeneity Discovery*'. In Semantic Web: Research and Applications, LNCS 3532: 303-317. Springer.

Silva, N. 1998. '*Sistemas Holónicos de Produção - Especificação e Desenvolvimento*'. Thesis/Dissertation, Faculty of Engineering, University of Porto. Porto (PT).

Silva, N. 2004. '*Multi-dimension Service-Oriented Ontology Mapping*'. Thesis/Dissertation, University of Trás-os-Montes and Alto Douro. Vila Real (PT).

Silva, N., and C. Ramos. 1999. '*Holonic Dynamic Scheduling Architecture and Services*'. In Proc. International Conference on Enterprise Information Systems. Setúbal (PT).

Silva, N., M. Viamonte, and P. Maio. 2009. '*Agent-Based Electronic Market With Ontology-Services*'. In Proc. IEEE International Conference on E-Business Engineering, 51-58. Macau (CN).

Silva, N., P. Maio, and J. Rocha. 2005. '*An approach to ontology mapping negotiation*'. In Proc. Workshop on Integrating Ontologies of the 3rd International Conference on Knowledge Capture, 54-60. Banff (CA).

Sirin, E., B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. 2007. '*Pellet: A practical owl-dl reasoner*'. In Web Semantics: science, services and agents on the World Wide Web, 5(2): 51-53. Elsevier.

Snowden, D. 1999. '*Liberating Knowledge*'. In CBI Business Guide, 9-19. Caspian Publishing.

Staab, S., S. Santini, F. Nack, L. Steels, and A. Maedche. 2002. '*Emergent semantics*'. In Intelligent Systems, 17(1): 78–86. IEEE.

Stoilos, G., G. Stamou, and S. Kollias. 2005. '*A string metric for ontology alignment*'. In Semantic Web - ISWC 2005, LNCS 3729: 624-637. Springer.

Straccia, U., and R. Troncy. 2005. '*oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies*'. In Proc. 6th International Conference on Web Information Systems Engineering (WISE), 133-147. New York (NY, US).

Studer, R., R. Benjamins, and D. Fensel. 1998. '*Knowledge Engineering: Principles and Methods*'. In Data and Knowledge Engineering, 25(1-2): 161-197. Elsevier.

Toulmin, S. 1958. '*The Uses of Argument*'. Cambridge University Press.

Trojahn, C., M. Moraes, P. Quaresma, and R. Vieira. 2008. '*A Cooperative Approach for Composite Ontology Mapping*'. In Journal on Data Semantics X, 237-263. Springer.

Verheij, B. 2002. '*On the existence and multiplicity of extensions in dialectical argumentation*'. Arxiv *cs/0207067*.

Verheij, B. 2003. '*DefLog: on the Logical Interpretation of Prima Facie Justified Assumptions*'. In Journal of Logic and Computation, 13(3): 319-346. Oxford University Press.

Viamonte, M., N. Silva, and P. Maio. 2011. '*Agent-based Simulation of Electronic Marketplaces with Ontology-Services*'. In Proc. 23rd European Modeling and Simulation Symposium (EMSS'2011). Rome (IT).

Visser, P., D. Jones, T. Bench-Capon, and M. Shave. 1997. '*An Analysis of Ontological Mismatches: Heterogeneity versus Interoperability*'. In Proc. AAAI Spring Symposium on Ontological Engineering. Stanford (CA, US).

von Alan, R., S. March, J. Park, and S. Ram. 2004. '*Design Science in Information Systems Research*'. In Management Information Systems, 28(1): 75-105. Quarterly.

Vreeswijk, G. 1997. '*Abstract argumentation systems*'. In Artificial Intelligence, 90(1-2): 225-279. Elsevier.

Wache, H., T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. 2001. '*Ontology-based integration of information - a survey of existing approaches*'. In Proc. Workshop on Ontologies and Information Sharing of the International Joint Conference on Artificial Intelligence, 108-117. Seattle (WA, US).

Walton, D. 2006. '*Fundamentals of critical argumentation'.* Cambridge University Press.

Walton, D., and E. Krabbe. 1995. '*Commitment in dialogue: basic concepts of interpersonal reasoning*'. SUNY Press.

Wiederhold, G. 1992. '*Mediators in the Architecture of Future Information Systems*'. In IEEE Computer, 25(3): 38-49. IEEE.

Wiesman, F., N. Roos, and P. Vogt. 2001. '*Automatic ontology mapping for agent communication*'. In Proc. 1st International Joint Conference on Autonomous agents and multiagent systems: part 2, 563-564. Bologna (IT).

Wooldridge, M. 2000. '*Reasoning about rational agents*'. MIT Press.

Wooldridge, M. 2009. '*An Introduction to MultiAgent Systems*'. Wiley.

Wooldridge, M., and N. Jennings. 1995. '*Intelligent Agents: Theory and Practice*'. In Knowledge Engineering Review, 10: 115-152. Cambridge University Press.

Zaihrayeu, I. 2006. '*Towards Peer-to-Peer Information Management Systems*'. Thesis/Dissertation, International Doctorate School in Information and Communication Technology, University of Trento. Trento (IT).

Zhdanova, A., and P. Shvaiko. 2006. '*Community-driven ontology matching*'. In Semantic Web: Research and Applications, LNCS 4011: 34-49. Springer.