# Automatic Schema Matching as a Complex Adaptive System: a new Approach based on Agent-based Modeling and Simulation

Hicham Assoudi, Hakim Lounis

Département d'Informatique
Université du Québec à Montréal, UQÀM
Succursale Centre-ville, H3C 3P8, Montréal, Canada
Email: assoudi.hicham@courrier.uqam.ca lounis.hakim@uqam.ca

***Abstract* — In this work, we have investigated the use of Complex Adaptive System theory, derived from systemic thinking, to seek innovative responses to the challenges that Automatic Schema Matching approaches always face (e.g., complexity, uncertainty). We propose a conceptual model for the Simulation of Automatic Schema Matching, and we describe how we modeled it using the approach of Agent-Based Modeling and Simulation. This effort gives rise to a tool (prototype) for schema matching. A set of experiments demonstrates the viability of our approach on two main aspects: (i) effectiveness (increasing the quality of the found alignments) and (ii) efficiency (reducing the effort required for this efficiency). The results obtained have first provided proof of the viability of our approach, but also demonstrated a significant paradigm shift in this domain, where automatic schema matching has never been addressed by adopting systemic thinking.**

***Keywords - Schema Matching; Systemic Approach; Complex Adaptive Systems; Agent-Based Modelling and Simulation.***

## I. INTRODUCTION

One of the key tasks in developing solutions for interoperability between heterogeneous information systems is schema matching. Indeed, it is omnipresent in several fields, involving the management of metadata (i.e., schemas, ontologies). This is the case of integration, exchange or migration of data, the semantic Web, e-commerce, etc. [1] [2] [3].

Several definitions exist for the schema matching process. Rahm and Bernstein [2] in their study of the different approaches for solving the problem of schema matching, define a schema as a set of elements connected by a given structure. The schema must be represented by a notation, to capture in a natural and logical way the notion of element and structure, such as, an object-oriented model, an entity-relation model, XML, or in the form an oriented graph. The task of schema matching is to find semantic mapping relationships between elements of data schemas. Such a process is illustrated in Figure 1. Generally, it aims at finding a pairing of elements (or groups of elements) from the source schema and elements of the target schema such that pairs are likely to be semantically related [2] [4].

Automatic Schema Matching (ASM) is a complex task in more than one way: (i) the heterogeneity and ambiguity intrinsic to the elements of schemas to be matched, (ii) the uncertain character of the matching results, (iii) the challenge of optimizing the pairing (combinatorial explosion), etc.

For a long time, this task remained a manual task reserved mainly for experts with a good understanding of the semantics of the different schemas, and a proficiency of transformation languages. On the other hand, as schemas became more complex, this task began to become tedious, time-consuming and error-prone.

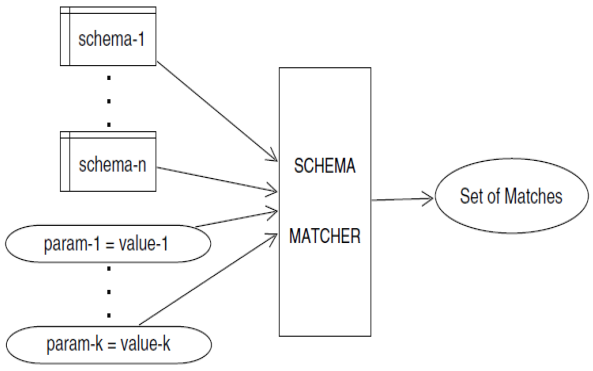


Figure 1. Matching process inputs and outputs

Schema matching existing approaches still rely largely on human interactions, either for the matching results validation, during the post-matching phase, or for the matching process optimization, during the pre-matching phase. Although this human involvement in the automatic matching process could be considered as acceptable in a lot of matching scenarios, nevertheless, it should be kept to a minimum, or even avoided, when dealing with high dynamic environments (i.e., semantic Web, Web services composition, agents communication, etc.) [5]. Thus, the existing approaches are not suited for all the matching contexts due to their intrinsic limitations. We can summarize those limitations as follows:

- Lack of autonomy to the extent that the user involvement is still needed for the results validation and analysis, but also for matching process configuration and optimization

(tuning) to improve the matching result quality and then reduce uncertainty.

- Lack of adaptation in sense that the optimization task of the matching tool must be repeated and adapted manually, for every new matching scenario.

Thus, we were motivated to investigate other paradigms to approach the issue of ASM, as a complex subject appropriate for an approach, which would allow to see this problematic as a whole, that is, as a system whose execution, configuration, and optimization depend not only on the different components of the system, but also on taking into account the relationships and interactions between these components. We try to answer the following general question: "How can we, with the help of a generic approach, better manage complexity and uncertainty inherent to the automatic matching process in general, and in the context of dynamic environments (minimal involvement of the human expert)?"

More specifically, we asked the following questions:

(i) How can we model the complexity of the matching process to help reduce uncertainty?

(ii) How can we provide the matching process of autonomy and adaptation properties with the aim to make the matching process able to adapt to each matching scenario (self-optimize)?

(iii) What would be the theoretical orientation that may be adequate to respond to the above questions?

In our work, we have investigated the use of the theory of Complex Adaptive System (CAS) emanating from systemic thinking, to seek, far from the beaten path, innovative responses to the challenges faced by classical approaches for automatic schema matching, (e.g., complexity, uncertainty). The central idea of our work is to consider the process of matching as a CAS and to model it using the approach of Agent-Based Modeling and Simulation (ABMS). The aim being the exploitation of the intrinsic properties of the agent-based models, such as emergence, stochasticity, and self-organization, to help provide answers to better manage complexity and uncertainty of Schema Matching.

Thus, we propose a conceptual model for a multi-agent simulation for schema matching called Schema Matching as Multi-Agents Simulation (SMAS). The implementation of this conceptual model has given birth to a prototype for schema matching (Reflex-SMAS).

Our prototype Reflex-SMAS was submitted to a set of experiments, to demonstrate the viability of our approach with respect to two main aspects: (i) effectiveness (increasing the quality of the found matchings), and (ii) efficiency (reducing the effort required for this efficiency). The results came to demonstrate the viability of our approach, both in terms of effectiveness or that of efficiency.

The empirical evaluation results, as we are going to show in Section IV of this paper, were very satisfactory for both effectiveness (correct matching results found) and efficiency (no optimization needed to get good result from our tool).

The current paper is organized as follows: Section II discusses schema matching through a state of the art that identifies the important factors affecting the schema matching process. Section III presents the chosen paradigm to address the problem. Section IV shows the results obtained by our approach, and how we can compare them to those obtained in other works. Finally, the last section concludes and summarizes this work.

## II. CURRENT APPROACHES OF SCHEMA MATCHING

The schema matching process is often used as a prerequisite for solving other issues, such as data integration or exchange. Indeed, as part of a process of integration or exchange of data, the matching process becomes the task that is responsible for finding an alignment that is semantically equivalent between a source schema and a destination schema. This alignment, in the case of data integration for example, will participate in finding "answers to requests" made to several disparate data sources by consolidating the schemas of these disparate sources into a common schema (i.e., a global scheme). In the case of data exchange, finding an alignment between a source schema and a target one serves for the exchange of data between heterogeneous enterprise systems or applications (such as Enterprise Resource Planning systems, databases, or legacy systems) by helping transform data from the source system format to the target system format.

Generally, Schema Matching is a manual task reserved for human experts with a good understanding of the semantics of different schemas. However, this task can be a tedious, time-consuming and error-prone task.

For many years, several researches have investigated the problematic of automating the task of Schema Matching (including matching ontologies). The main goal is the development of techniques (algorithms, tools, etc.) allowing the automatic or semi-automatic discovery of the correspondences between the elements.

Many algorithms and approaches were proposed to deal with the problem of schema matching and mapping [2][6]-[16]. Although the existing schema matching tools comprise a significant step towards fulfilling the vision of automated schema matching, it has become obvious that the user must accept a degree of imperfection in this process. A prime reason for this is the enormous ambiguity and heterogeneity of schema element names (descriptions). Thus, it could be unrealistic to expect a matching process to identify the correct matchings for any possible element in a schema [17] [18].

Despite the profusion of approaches, human involvement is always required, for the automatic matching process itself, or for the optimization of the performances. They also rely largely on immediate human interaction for the validation of the result of the process, or for configuration/optimization of the process during the pre-matching phase. This human interaction is generally conceivable for many contexts; on the other hand, in other contexts where environments are highly dynamic (e.g., semantic Web, composition of Web services, communication between agents), expert involvement must be reduced to a minimum.

The vision of a complete automation of the matching process of schemes is compromised by a human involvement.

A comprehensive literature review, of the existing matching tools and approaches, allowed us to identify the most important factors affecting, in our opinion, the schema matching process. Moreover, some causal relationships, between those different factors, participating to the schema matching difficulties and challenges, were identified.

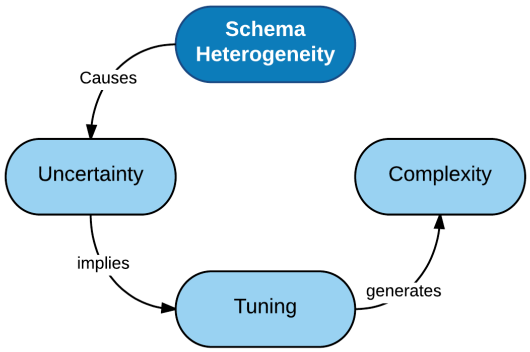


Figure 2. Schema Matching impacting factors causality diagram

As shown in Figure 2, the factors influencing the Schema Matching are:

- Heterogeneity: in general, the task of matching involves semantics (understanding the context) to have complete certainty about the quality of the result. The main challenge in all cases of automatic matching is to decide the right match. This is a very difficult task mainly because of the heterogeneity of the data.
- Uncertainty: the cause for this uncertainty lies mainly in the ambiguity and heterogeneity, both syntactic, and semantic, which often characterize the Schema Elements to match.
- Optimization: the uncertainty about the matching results implies the optimization of the process to improve the matching quality, and the testing of different combinations (e.g., different Similarity Measures, Aggregate Functions, and Matching Selection Strategies). Each step of the matching process involves choosing between multiple strategies, which leads to a combinatorial explosion (complexity).
- Complexity: the matching process optimization generates complexity because of the search space (combinatorial explosion). In addition, changing matching scenarios exacerbates this complexity to the extent that the result of the optimization often becomes obsolete with changing scenarios.

One of the commonalities between all existing approaches is the thinking behind these approaches, namely, reductionism (as opposed to holism). The reductionist thinking is a very common and efficient thinking approach. It is at the basis of the almost totality of previous schema matching approaches, and then, on their characteristics that are, in our view, the root causes preventing the automatic matching schemes to cope fully with the challenges and difficulties.

Reductionism, as opposed to systemic (holism), is a philosophical concept that refers both to the way of thinking solutions as well as to their modeling methodology (Figure 3). Reductionism advocates reducing system complexity or phenomenon to their basic elements, which would then be easier to understand and study [19]. This reductionist approach, despite its high efficacy in several areas, shows, however, its limits within certain contexts. In fact, for explaining certain phenomena or solving certain problems, the approach consisting of reducing or abstracting the reality to a linearization of simple relationships of causes and effects between a complex system underlying fundamental components, appears as a highly limiting and simplifying approach.

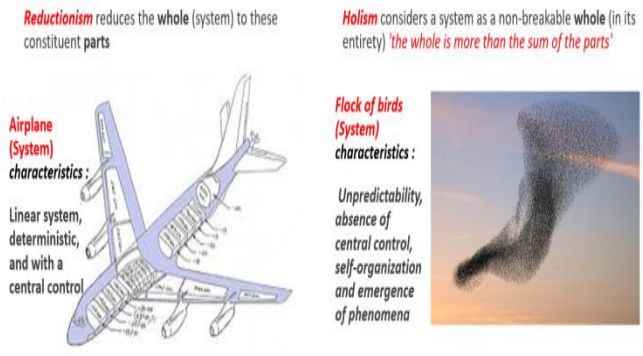


Figure 3. Reductionism vs. Holism

With regard to schema matching, it seems clear, as Figure  illustrates it, that all current approaches follow the reductionist thinking. They abstract the matching process to a linear function with a set of inputs and outputs. This function is decomposed into a series of modules, each of

which is responsible for the running of a stage of the process (e.g., selection and matching execution).

In fact, the matching process can be summarized to three important steps: (i) the stage of selection and execution of the matchers (calculation of similarities), (ii) the stage of the combination of the results (the best similarity scores) based on aggregation functions, and finally (iii), the step of the selection of the alignment (selection of the most promising matches) based on thresholds or maximums.

So, in order to solve the automatic Schema Matching problem, the existing solutions are adopting a linear and analytical approach. At each stage of the matching process, problems are analyzed and broken down into sub problems and then for each specific sub-problem, dedicated and specific solution are proposed.

Some fundamental and intrinsic characteristics, common to all current Schema Matching systems, may partially explain their inability to overcome the limitation of the complexity and other challenges, such as uncertainty. Those characteristics are declined as following: these systems are (i) complicated and not complex, (ii) linear (analytical, deterministic and predictable) and not non-linear, (iii) centralized rather than decentralized (parallelism and emerging solutions), (iv) and finally, configurable and not adaptable (self-configuration, self-optimization).

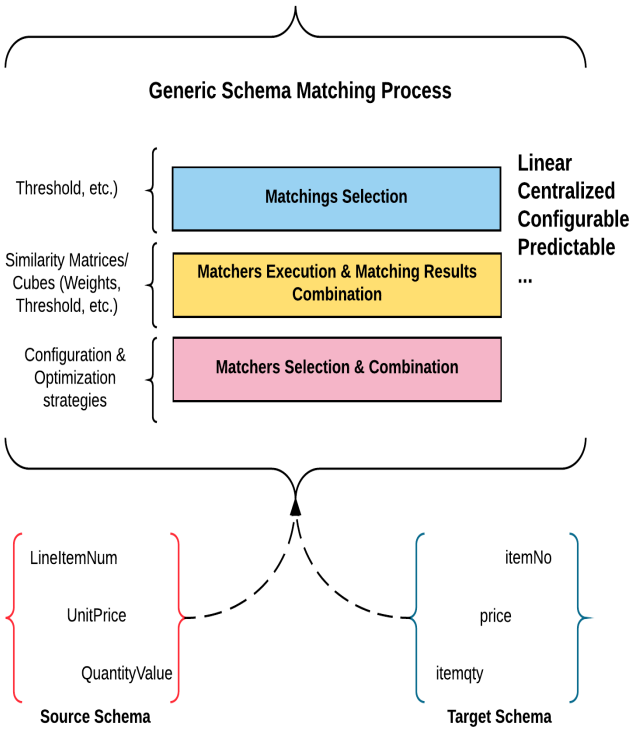


Figure 4. Generic Schema Matching process (linear process with an analytical-based resolution)

We can, thus, postulate that the Reductionist school of thought, leading to complicated and not complex systems, could be the root cause that prevents existing approaches from coping with the challenges of uncertainty and complexity.

The need to explore new approaches to make systemic and holistic responses to the problems of matching leads us to raise the question: how can we have a matching solution that could give us high-quality matching results, for different matching scenarios and this with a minimal optimization effort from the end-user?

Our premise is that a good part of the answer may come from the theory of CAS where modeling the complexity of adaptation and evolution of the systems is at the heart of this theory. Having a schema matching approach that can face and overcome the challenges facing the existing schema matching tools requires, in our view, a paradigm shift, placing the notions of adaptation, evolution, and self-organization at its center. We strongly believe that the theory of CAS, which is exploited to explain some biological, social, and economic phenomena, can be the basis of a programming paradigm for ASM tools.

Our hypothesis is that, to realize the vision of complete automation, it would be necessary to operate paradigm-shift and move from reductionist and complicated solutions to a holistic (systemic) solution based on the paradigm of Complex Adaptive Systems (CAS) with the Multi-agents Simulation as the cornerstone at the heart of our proposed approach.

Our goal is to find an innovative solution, for the challenges of automatic matching, by exploiting multi-agent simulations, taking place in an artificial world, and taking advantage of the computing power of current computers.

« *Simulation models provide virtually unlimited power; or rather, they provide unlimited virtual power. If you can think of something, you can simulate it. Experimenting in a simulated world, you can change anything, in any way, at any time - even change time itself.* » [20].

Figure 5 below represents the different CAS fundamental characteristics that could allow the matching system to move from a chaos state (i.e., initial state) to a state of equilibrium (i.e., final state of the system).


Figure 5. From chaos to equilibrium

The initial state of the system is a state of chaos where the agents are unstable because of their matching status (indeterminate matching). After the start of the simulation, the adaptation stage begins where the agents interact

searching for the best match. Over the cycles of the simulation, consensual matchings (self-organization) begin to form (i.e., local solution representing a local equilibrium to the pair of agents), and thus make emerge the final solution of the alignment (global solution).

The final state of the system is reached once a balance for the system in its entirety is found, thereby signaling the end of the simulation.

## III. SCHEMA MATCHING AS A SYSTEMIC APPROACH

As part of our research we investigated the use of the theory of CAS (systemic thinking), to try to find an innovative response to challenges (i.e., complexity, uncertainty) that the conventional approaches for schema matching are still facing.

We think that the CAS could bring us the adaptation capability to the realm of schema matching tools (self-configuration and self-optimization), which should relieve the user from the complexity and effort resulting from configuring and optimizing the automatic schema matching systems.

Before going any further, let us first try to explain our vision of the problematic of ASM under the prism of the theory of CAS. First and foremost, we wonder about the nature of ASM solutions, whether they can be considered as systems. The answer is unequivocally positive. Even if one strictly stands for the definition of the term system, which means "a coherent set of closely related parts", it is clear that the automatic matching of schemas is a system, because of ASM is a coherent set consisting of several components, related to each other, in this case schemas, matchers, etc. Now, if we come to the systemic meaning of the term, the automatic matching of schemas is a system whose different components, for example elements of schemas (which can be represented by agents seeking to find the best match), must interact within this system with the objective of producing a final result called alignment.

Next, consider why a system of ASM can be described as complex. To this end, let us recall the distinction between a "complex" system and a "complicated" system [21]. First, in complex systems, relationships between agents (i.e., system elements) are more important than the agents themselves, unlike complicated systems where elements and their relationships are of equal importance. Second, in complex systems, simple rules can produce surprising and complex responses, while in complicated systems, the results of simple algorithms are simple and predictable. And finally, in complex systems the agents have the latitude to respond according to the limits of their rules, as opposed to the complicated systems, where the response of the components is completely determined.

On the basis of this distinction, we consider that the ASM process must be thought and modeled as a complex system (our approach) and not as a complicated system (current approaches). So, our conceptual model for schema matching, based on the theory of complexity, sees the schema matching process as a complex adaptive system.

As illustrated in Figure 6, in this model, each schema element of the schemas to match (source or target schema) is modeled as an autonomous agent, belonging to a population (source or target schema population). Each agent behaviors and interaction, at the micro level, with the other agents in the opposite population and with its environment, brings out at the macro level, a self-organized system that represents the global solution to matching problem (i.e., relationships between schemas elements). In other words, the resolution of the matching problem goes through individual effort deployed by each agent, locally, throughout the simulation to find the best match in the opposite population.

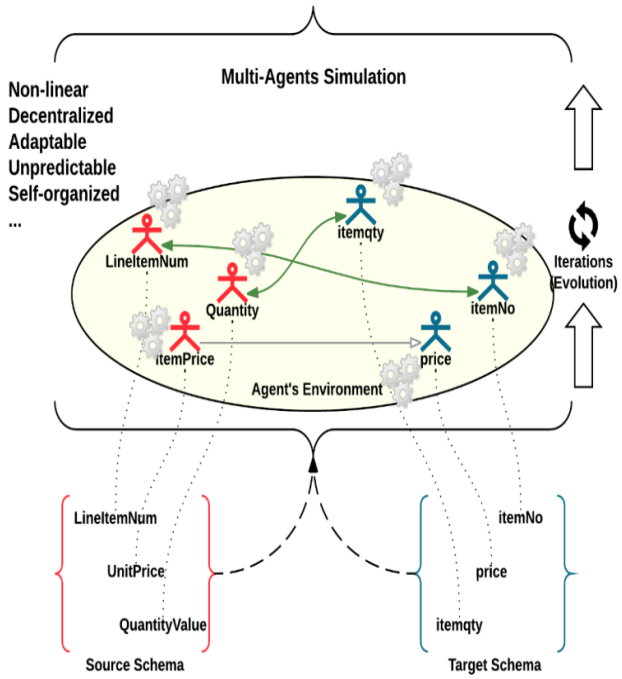


Figure 6. Schema Matching as Multi-Agents Simulation (non-linear process with emergence-based resolution)

We think that many intrinsic properties of our model, derived from the ABMS modeling approach, can contribute efficiently to the increase of the matching quality and thus the decrease of the matching uncertainty. These properties are:

- Emergence: the emergence of the macro solution (schema matching) comes from local behaviors, rules and interactions between agents (micro solutions).
- Self-organization: the cooperation of source and target schema elements (represented as agents) helps to reach a consensus about their best matching.
- Stochasticity (randomness): the randomness within the model, gives the ability to perform statistical analysis on

the outcome of multiple simulations (meta-simulation) for the same matching scenario.

Briefly, our idea is to model the Schema Matching process as interactions, within a self-organized environment, between agents called "Schema Attribute Agent". In the rest of the paper, we are going to refer to the "Schema Element Agent" simply as agent. Each schema element is modeled as an agent belonging to one of two populations: source or target schema group. Furthermore, the schema matching process is modeled as the interaction between the two populations of agents.

In our model, the internal architecture of the agents is Rule-based (reflexive agent). The agents have as a main goal to find the best matching agent within the other group of agents.

The foundation of the rules governing the agent's behaviors is stochasticity (randomness). In fact, a certain degree of randomness is present in each step executed by each agent during the simulation.

The main random elements influencing the simulation are as follows:

- Similarity Calculation based on similarity measures selected randomly from a similarity measures list.
- Similarity Scores aggregation based on aggregation functions selected randomly from an aggregation function list (MAX, AVERAGE, WEIGHTED).
- Similarity score validation based on generated random threshold value (within interval)

As opposed to deterministic solutions for schema matching (all the existing matching solutions), the nondeterministic and stochastic nature of our agent-based simulation increase the confidence in the quality of the matching results. Even though the agent's behaviors are based on randomness, our model can often produce the right matchings at the end of each simulation run.

In the context of our operational model, the agent during the perception phase, perceives its environment by interrogating it, by performing similarity calculations (which can be considered as an act of recognition) or by capturing certain events. The result of this phase will be a set of percepts, allowing the agent to identify the agents of the other group, available for matching. The capture of events, coming from the environment, is another action of perception: for instance, the event that is triggered when the agent is chosen by another one as a matching candidate. During the decision phase, the agent from the results of the perception phase, reasons, deliberates and decides on the action to be selected. The decisions, involving the choice of actions, are the following: (i) the decision concerning the convergence of similarities and the selection of a candidate matching, (ii) the decision concerning the reset of the beliefs concerning the candidate matching, and (iii) the decision on consensual matching. During the action phase, the agent executes the actions selected during the previous phase. The current iteration of the simulation ends with this phase.

Figure 7 illustrates the internal states of each agent. It allows representing the transitions between the internal states, during the perception-decision-action cycle of the agent.

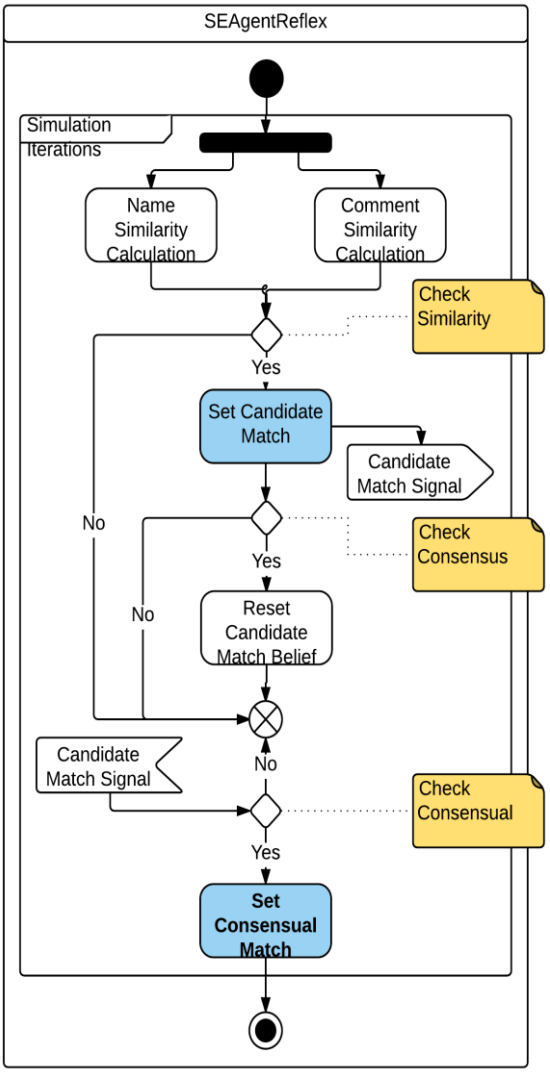


Figure 7. Agent behavior

The behavior of the agent is driven by the goal of finding a consensual match. The consensus-selection approach is a naive approach, consisting of waiting for a consensus that must coincide for both agents (which may imply a longer duration for the simulation).

The main key-features of our conceptual model are summarized as follows:

- Stochastic Linguistic Matching: similarity calculation based on similarity measures selected randomly from a similarity measures list. Similarity Scores aggregation based on aggregation functions selected randomly from an aggregation function list (MAX, AVERAGE, WEIGHTED). Similarity score validation based on generated random threshold value (within interval). Figure 8 illustrates this process.

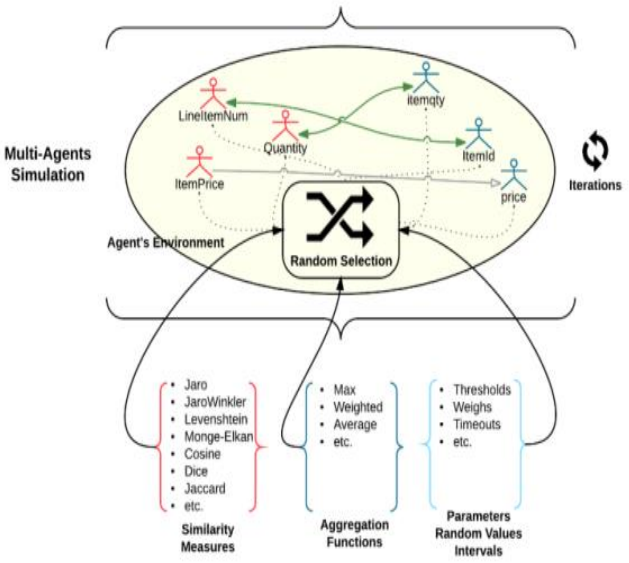


Figure 8. Stochastic Matching

- Consensual Matching Selection: to form a valid pairing/correspondence, the two agents (form opposite populations: source and target schemas) should refer to each other as candidate match (in the same time). Figure 9 is an illustration of such a process.

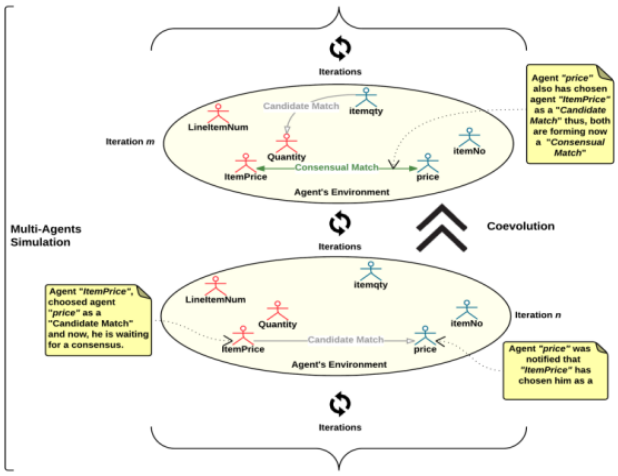


Figure 9. Consensual Matching

- Meta-Simulations and Statistical Analysis: performing statistical analysis on multiple simulation runs data is a good way to improve the confidence in the matching result obtained from our model. It is illustrated by Figure 10.

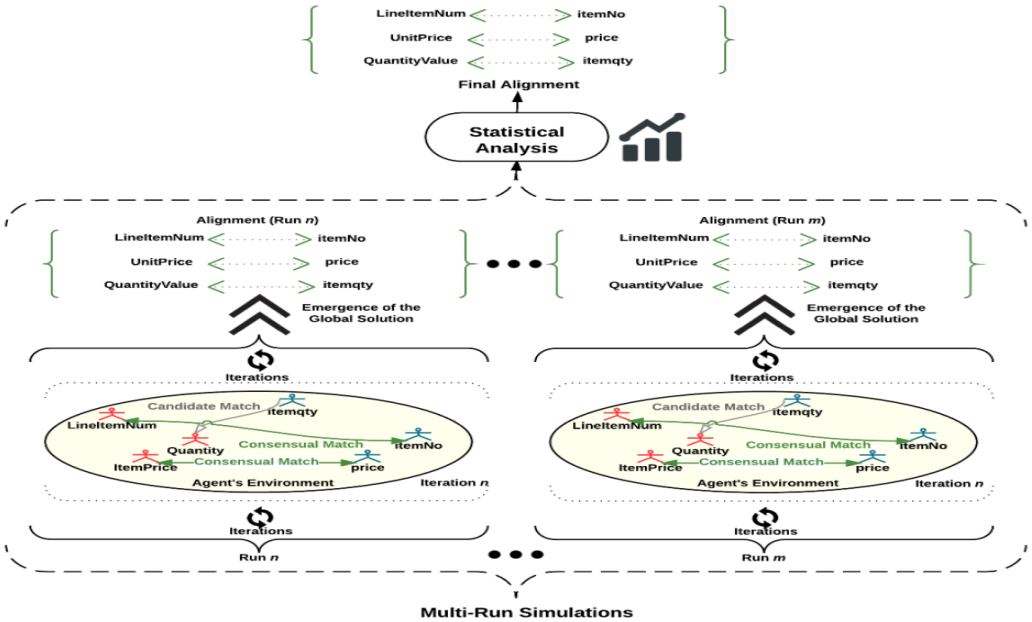


Figure 10. Meta-simulation and statistical analysis

We believe that the conceptualization and the modeling of schema matching as multi-agent simulation will allow the design of a system exhibiting suitable characteristics:

(i) An easy to understand system, composed of simple reflexive "agents" interacting according to simple rules.

(ii) An effective and efficient system, autonomously changing over time, adapting, and self-organizing; a system allowing the emergence of a solution for any given matching scenario.

As depicted in Figure 1, our Reflex-SMAS prototype core was implemented in Java using the open source ABMS framework Repast Simphony (2.1) [22], [23], and the open source framework for Text Similarity DKPro Similarity (2.1.0) [24]. The open source R language (R 3.1.0) [25] was used for statistical data analysis.

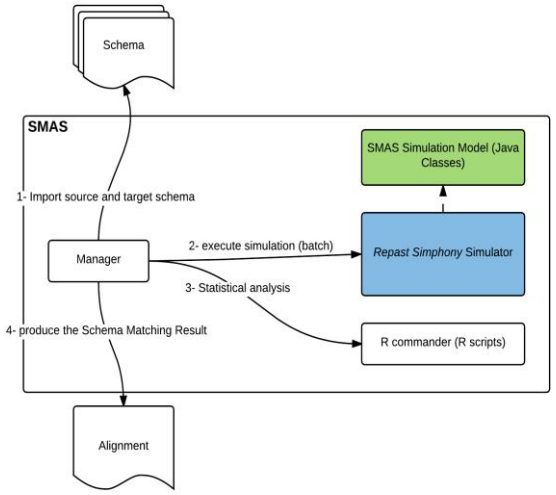


Figure 11. High-level Architecture for Reflex-SMAS

In the next section, we are going to describe the empirical evaluation of the prototype Reflex-SMAS.

## IV. EMPIRICAL EVALUATION

The validation of agent-based simulation models is a topic that is becoming increasingly important in the literature on the field of ABMS. Three types of validation could be identified [26]: (i) Empirical Validation, (ii) Predictive Validation, and (iii) Structural Validation.

As we will see in detail, the empirical validation is the type of validation that we have adopted for the evaluation of our Agent-based Simulation Model for Schema Matching (i.e., prototype Reflex-SMAS).

First, we will start with the description of the methodology used as our validation approach, and then we continue by providing a summarized view of our validation results.

### A. Evaluation Objectives and Strategy

We are seeking, through this empirical evaluation, to validate the following aspects of our prototype Reflex-SMAS:

- That our solution is, indeed, an effective and efficient automatic schema matching system, capable of autonomously changing behaviors and evolving over time, to adapt, and to self-organize and thus make the solution for any matching scenario to emerge. The effectiveness must be translated mainly by an increase in the quality of the found alignment and a reduction of the uncertainty. Efficiency, on the other hand, must be materialized by reducing the effort required for this efficiency, in particular reducing the user effort required to configure or optimize the matching system, without significantly impairing the response time, which should remain within the limits of the viable and the acceptable (especially for the use of matching in highly dynamic environments).
- That our solution is easy to understand, and therefore, could display a high degree of maintainability (e.g., adding new matchers).

The proof strategy consists on conducting experiments and then collecting and analyzing data from these experiments. Thus, the validation approach that we have adopted is considered as a hybrid validation approach combining two validation approaches coming from two different fields, namely Schema Matching and ABMS. On one hand, from the field of Schema Matching, we are leveraging a popular evaluation method consisting of the comparison of results with those expected by the user [27]. On the other hand, from the field of ABMS, we are using the Empirical Validation, which is mainly based on the comparison among the results obtained from the model and what we can observe in the real system.

Thus, the strategy adopted for the validation of our prototype (implementing our multi-agent simulation model for schema matching) consists of:

- Defining different synthetic matching scenarios (three matching scenarios namely "Person", "Order" and "Travel") with different sizes and different level of lexical heterogeneity, so we can evaluate the prototype matching performance in different situations (adaptation).
- Conducting experiments, compiling results and evaluating the matching performance by comparing, for those three matching scenarios, the matching results (matchings) obtained from our prototype Reflex-SMAS with the results expected by the user.

In the first matching scenario "Person", we need to match two schemas with small size (i.e., six elements) showing a medium lexical heterogeneity level. The second

matching scenario "Order" is composed of schemas with medium size with a high lexical heterogeneity level. The schemas in the last matching scenario "Travel" have a relatively big size with a low lexical heterogeneity level.

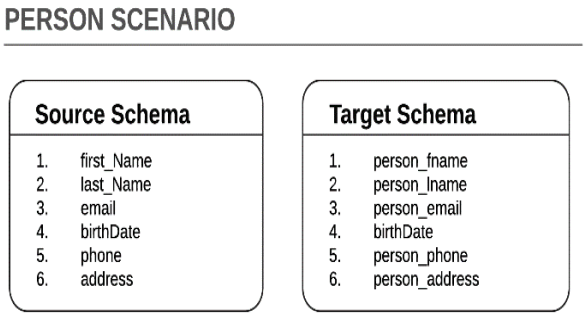


Figure 12. Matching Scenario "Person"

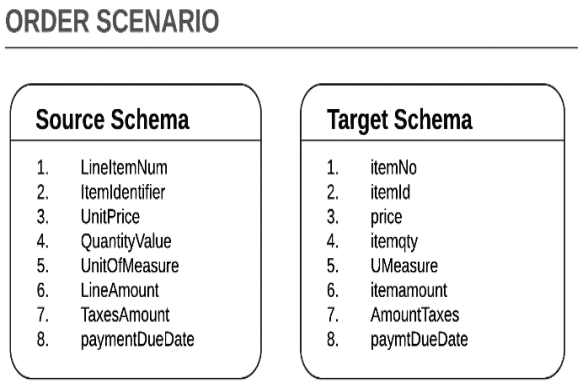


Figure 13. Matching Scenario "Order"

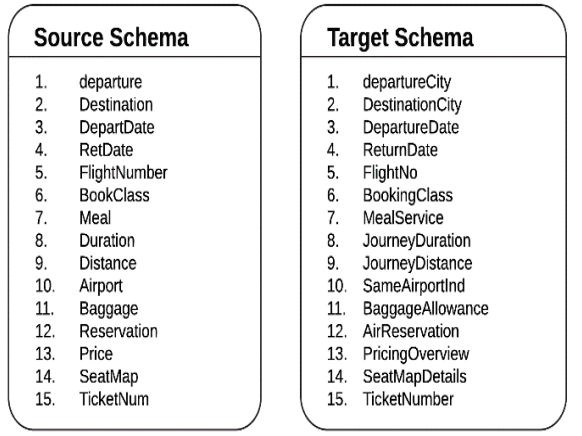


Figure 14. Matching Scenario "Travel"

In order to assess the relevance and level of difficulty that can represent those synthetic matching scenarios (i.e., "Person", "Order" and "Travel"), we decided to evaluate them, first, using the well-known matching tool COMA [28]–[30]. Since, the COMA tool was not able to resolve all the all expected matches for those scenarios, we can say that the proposed synthetic matching scenarios, should be enough challenging scenarios for our validation (from their level of heterogeneity perspective).

Regarding the experiments execution and results compilation, we have decided to run series of three meta-simulations for each scenario (each meta-simulation includes 10 simulations).

The final matching result is based on a statistical analysis of each meta-simulation outcome. In other word, the matching result relies on the calculation of the frequency of occurrence of a found match on the ten simulations composing the meta-simulation. Furthermore, executing for each scenario the meta-simulations three times is a choice that we made to help with the assessment of the experiment repeatability.

### B. *Experiment Results*

This section summarizes the results obtained from the experiments conducted to evaluate the Reflex-SMAS tool. After executing the set of three meta-simulations for each matching scenario, we have compiled the results for the performance for each meta-simulation for all scenarios. As indicated in Table I, our tool was able to correctly find all the expected correspondence by the user (a 100% success rate) after each meta-simulation, and for each scenario.

TABLE I. REFLEX-SMAS EXPERIMENT COMBINED RESULTS

| Scenario | M.S. | M. to F. | C.M.F. | % C.M.F. |
|---|---|---|---|---|
| Person | 1 | 6 | 6 | 100% |
| Person | 2 | 6 | 6 | 100% |
| Person | 3 | 6 | 6 | 100% |
| Order | 1 | 8 | 8 | 100% |
| Order | 2 | 8 | 8 | 100% |
| Order | 3 | 8 | 8 | 100% |
| Travel | 1 | 15 | 15 | 100% |
| Travel | 2 | 15 | 15 | 100% |
| Travel | 3 | 15 | 15 | 100% |

M.S: Meta Simulation
M. to F: Matchings to Find
C.M.F: Correct Matchings Found
% C.M.F: % Correct Matchings Found

Regarding the response time, it corresponds to the execution, in parallel, of 10 individual simulations. At the level of the individual simulations, we are interested in the

discrete time analysis (i.e., iterations), which represents in some ways the measure of the effort expended by the agents to form consensual matching. The figure below shows the response time with respect to each meta-simulation and each scenario. By examining the graph, we can deduce the correlation between the response time and the nature of the different scenarios. We can for example see that the size of the schemas (i.e., "Travel" scenario) or the high lexical heterogeneity (i.e., "Order" scenario) correlates negatively with the number of iterations necessary to find the solution for each meta-simulation. On the other hand, we can notice in Figure 15, that the execution time (in minutes) of the different meta-simulations for the different scenarios oscillates, approximately, between 2 and 3 minutes.

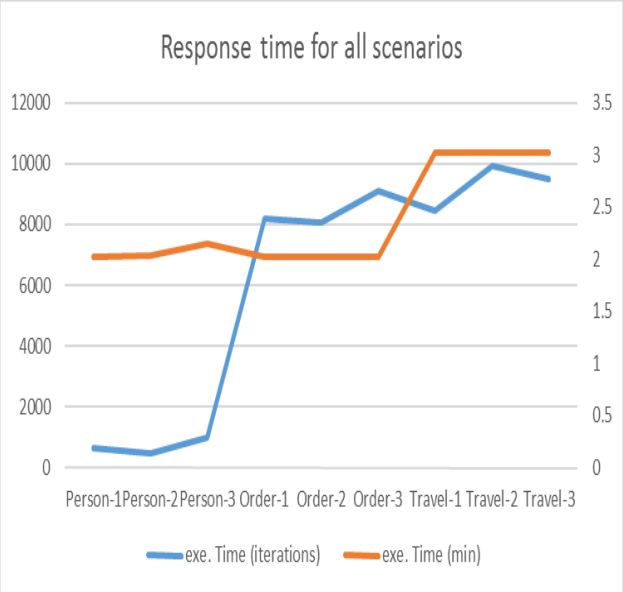


Figure 15. Number of iterations for all scenarios

Now, if we compare the results of our Reflex-SMAS prototype with COMA tool results, we can clearly notice that our tool outperformed the COMA tool in all the syntactic matching scenarios. Table II shows the compared result for Reflex-SMAS vs. COMA.

TABLE II. REFLEX-SMAS VS. COMA EXPERIMENT COMBINED RESULTS

| | | ***Reflex-SMAS*** | | ***COMA*** | |
|---|---|---|---|---|---|
| **Scenario** | **M. to F.** | **C.M.F.** | **% C.M.F.** | **C.M.F.** | **% C.M.F.** |
| Person | 6 | 6 | 100% | 5 | 83% |
| Order | 8 | 8 | 100% | 6 | 75% |
| Travel | 15 | 15 | 100% | 13 | 87% |

Figure 16 shows a comparison of the performance obtained for scenarios "Person", "Order" and "Travel" with our prototype compared to those obtained with the COMA tool.

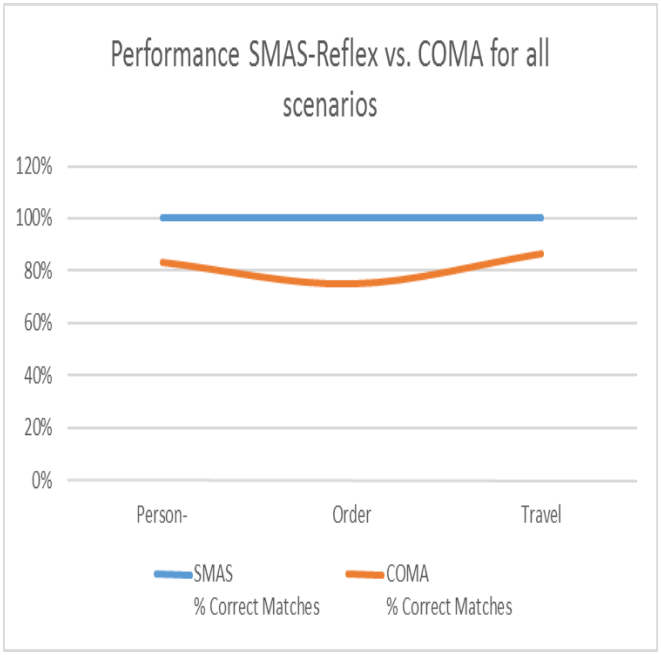


Figure 16. Comparative result between Reflex-SMAS and COMA

To challenge the “perfect” results obtained with our tool Reflex-SMAS for the synthetic matching scenarios, we were curious to know to what extent the performance obtained at the meta-simulations, may be impacted by a reduction in the number of individual simulations composing a meta-simulation. Therefore, we decided to conduct further experimentation, reducing, this time, the number of individual simulations of a meta-simulation from ten simulations to only three simulations.

The performance obtained in the experiment with the meta-simulations composed of three individual simulations instead of ten, has dropped for the scenarios "Order" and "Travel". It means that our matching tool Reflex-SMAS was not able to find all the expected matchings during some of the meta-simulations for those two scenarios (due to the high level of heterogeneity of the scenario “Order” and the big size of the scenario “Travel”). Unquestionably, we can conclude that the number of individual simulations, composing the meta-simulation is an important factor to ensure good matching performance (better quantification of the uncertainty regarding the outcome of the matching process) especially when it comes to scenarios involving large schemas and/or having a high level of heterogeneity.

## V. CONCLUSION AND FUTURE WORK

As part of our research, we have proposed an approach arising from systemic thinking, situating it more precisely, in the field of CAS. This approach has resulted in a multi-agent simulation approach based on a non-deterministic, stochastic (random), unpredictable and non-centralized model, where each simulation can give rise, even with the same inputs, to different outputs. The multiplication of the simulations allows the expansion of the space of possibilities and consequently the increase of the potentialities of a better quality of the pairing (i.e., reduction of the uncertainty on the matching obtained).

Our Reflex-SMAS prototype was subjected to a series of experiments whose objective was to demonstrate the viability of our approach in two main aspects: effectiveness and efficiency. This empirical evaluation step showed us clearly its capability of providing a high-quality result for different schema matching scenarios without any optimization or tuning from the end-user. The experiments results are very satisfactory. Thus, we can conclude that approaching the schema matching as a CAS and modeling it as ABMS is a viable and very promising approach that could greatly help to overcome the problems of uncertainty and complexity in the field of schema matching.

Our approach, which is part of the current of systemic thinking and which lies in the lineage of solutions coming from the field of CAS, brings a new perspective to the field of ASM, and represents, in this sense, a significant paradigm shift in this area.

As future work, we are planning to enhance the conceptual model of our prototype to tackle challenges, such as complex schema ($n$:$m$ cardinalities) by exploiting other Similarity Measures, such as Structural Similarities (schemas structures).

On the other hand, in order to open up new perspectives and to overcome the limits of purely reactive behavior, we are thinking on a "conceptual" evolution of the internal architecture of our agent, evolving it from a reactive agent to an agent of rational type. This evolution, as illustrated by Figure 17, consists in the implementation of a decision-making model under uncertainty, at the level of the decision-making phase of the agent, giving it the ability to reason and to choose between conflicting actions. The rational agent we are aiming for, should have a memory, a partial representation of its environment and other agents (its perception), and a capacity for reasoning, allowing it to make a rational choice (to choose the action with the greatest utility) that can guarantee it to maximize its satisfaction (measure of performance). We also speak of cognitive agent because it possesses explicit representations of it goals on which it is able to reason in order to produce action plans.

The result of this conceptual evolution could give rise to a new version of our prototype, and of course, it must be verified that the change from a reactive behavior to a rational one does not penalize the performance of the simulations.

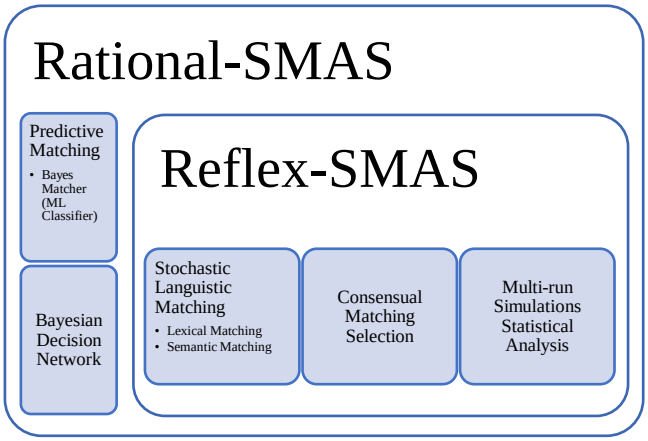


Figure 17. Towards a rational SMAS

## REFERENCES

[1] H. Assoudi and H. Lounis, "Implementing Systemic Thinking for Automatic Schema Matching: An Agent-Based Modeling Approach", in Proceedings of the 10th International Conference on Advanced Cognitive Technologies and Applications COGNITIVE 2018, pp. 43-50, 2018.

[2] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching ", the VLDB Journal, vol. 10, n° 4, pp. 334–350, 2001.

[3] H. N. Q. Viet, H. X. Luong, Z. Miklos, K. Aberer, and T. T. Quan, "A MAS Negotiation Support Tool for Schema Matching", in Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems, 2013.

[4] P. Bohannon, E. Elnahrawy, W. Fan, and M. Flaster, "Putting context into schema matching", in Proceedings of the 32nd international conference on Very large data bases, pp. 307–318, 2006.

[5] V. Cross, "Uncertainty in the automation of ontology matching", in Uncertainty Modeling and Analysis, 2003. ISUMA 2003. Fourth International Symposium on, pp. 135–140, 2003.

[6] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic schema matching with cupid ", in VLDB, vol. 1, pp. 49–58, 2001.

[7] B. Villanyi, P. Martinek, and A. Szamos, "Voting based fuzzy linguistic matching ", in 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), 2014, pp. 27-32, 2014.

[8] F. Duchateau and Z. Bellahsene, "Designing a benchmark for the assessment of schema matching tools", Open Journal of Databases (OJDB), vol. 1, n° 1, pp. 3–25, 2014.

[9] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao, "Reducing uncertainty of schema matching via crowdsourcing", in Proceedings of the VLDB Endowment, vol. 6, n° 9, pp. 757–768, 2013.

[10] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges", Knowledge and Data

Engineering, IEEE Transactions on, vol. 25, n° 1, pp. 158–176, 2013.

[11] E. Peukert, "Process-based Schema Matching: From Manual Design to Adaptive Process Construction", 2013.

[12] E. Peukert, J. Eberius, and E. Rahm, "A self-configuring schema matching system", in Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 306–317, 2012.

[13] W. Nian-Feng and D. Xing-Chun, "Uncertain Schema Matching Based on Interval Fuzzy Similarities", International Journal of Advancements in Computing Technology, vol. 4, n° 1, 2012.

[14] D. Ngo and Z. Bellahsene, "YAM++: a multi-strategy based approach for ontology matching task ", in Knowledge Engineering and Knowledge Management, Springer, pp. 421–425, 2012.

[15] J. Gong, R. Cheng, and D. W. Cheung, "Efficient management of uncertainty in XML schema matching", The VLDB Journal—The International Journal on Very Large Data Bases, vol. 21, n° 3, pp. 385–409, 2012.

[16] A. D. Sarma, X. L. Dong, and A. Y. Halevy, "Uncertainty in data integration and dataspace support platforms", in Schema Matching and Mapping, Springer, pp. 75–108, 2011.

[17] A. Gal, "Managing uncertainty in schema matching with top-k schema mappings", in Journal on Data Semantics VI, Springer, pp. 90–114, 2006.

[18] A. Gal, "Uncertain schema matching", Synthesis Lectures on Data Management, vol. 3, n° 1, pp. 1–97, 2011.

[19] R. Fortin, "Comprendre la complexité: introduction à La Méthode d'Edgar Morin". Presses Université Laval, 2005.

[20] L. W. Schruben, "Don't Try These in the Real World". 2015.

[21] W. Jones, "Complex Adaptive Systems." Beyond Intractability, Eds. Guy Burgess and Heidi Burgess, Conflict Information Consortium, University of Colorado, Boulder, Posted: October 2003.

[22] M. J. North, E. Tatara, N. T. Collier, and J. Ozik, "Visual agent-based model development with Repast Simphony", Technical report, Argonne National Laboratory, 2007.

[23] M. J. North, "R and Repast Simphony", 2010.

[24] D. Bär, T. Zesch, and I. Gurevych, "DKPro Similarity: An Open Source Framework for Text Similarity", in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 121–126, 2013.

[25] R. C. Team, "R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing; 2012", Open access available at: http://cran. r-project. org, 2011.

[26] M. Remondino and G. Correndo, "Mabs validation through repeated execution and data mining analisys", International Journal of Simulation: Systems, Science & Technology, vol. 7, n° 6, 2006.

[27] Z. Bellahsene, A. Bonifati, F. Duchateau, and Y. Velegrakis, "On evaluating schema matching and mapping", in Schema matching and mapping, Springer, pp. 253–291, 2011.

[28] H.-H. Do and E. Rahm, "COMA: a system for flexible combination of schema matching approaches", in Proceedings of the 28th international conference on Very Large Data Bases, pp. 610–621, 2002.

[29] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with COMA++", in Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 906–908, 2005.

[30] S. Massmann, S. Raunich, D. Aumüller, P. Arnold, and E. Rahm, "Evolution of the coma match system", Ontology Matching, pp. 49, 2011.