Transdisciplinary Lifecycle Analysis of Systems
R. Curran et al. (Eds.)

doi:10.3233/978-1-61499-544-9-603

Search Engine Optimization Process: A Concurrent Intelligent Computing Approach

Sylvain SAGOT [a,1], Alain-Jérôme FOUGÈRES [a, b] and Egon OSTROSI [a]
[a] *UTBM, University of Technology of Belfort-Montbéliard, 90010 Belfort, France, IRTES-M3M*
[b] *ESTA, School of business & engineering, 90010 Belfort, France;*

Abstract. The modification of customers' behaviors, the competition and the increase in the number of websites have forced companies to improve their visibility on the Internet. Search engines are widely used by customers and companies have to be present in the search engines results pages to be able to reach their clients. Different techniques have been developed to optimize the website's ranking on search engines, such as the SEO (Search Engine Optimization). The search engine optimization is the process of improving a website's position in the Internet search engine results. But the search engines protect their ranking models and results are long and difficult to obtain. Thus the non-transparency of ranking models, the important number of interactions, and the uncertainty in terms of results make the SEO process a complex problem. In order to permit its adaptability and sustainability in a dynamic and uncertain environment, the SEO process needs the elaboration of holistic and concurrent engineering approaches. In this paper, we used the multi-agent paradigm which is appropriate to solve concurrent and distributed problems. By decomposing the SEO process in sub-entities represented by communities of autonomous agents such as requirements, functions, constraints and solutions, we were able to analyze interactions and actions that take place into this process. A multi-agent based simulation using data from pharmacy websites was developed to test our approach.

Keywords. Search engine optimization, multi-agent system, concurrent engineering, engineering meta-model

Introduction

The use of Internet has been significantly impacted by the development of search engines in the mid-1990s [1] [2]. Nowadays, Internet users widely use search engines to find relevant information and the constant increase in website development intensifies the competitiveness. That is the reason why companies' websites have to be present on search engine results pages to hope to reach their clients.

Different techniques have been developed to optimize the website's ranking on search engines such as the Search Engine Optimization (SEO) [3]. The search engine optimization is the process of improving a website's position in the Internet search

[1] Corresponding author, E-mail: sylvain.sagot@utbm.fr

engine results. By using several techniques, it is possible for SEO practitioners to improve a website ranking in order to entice qualified traffic [4] [5].

The purpose of search engines is to bring to the users the most relevant information according to their search terms; this is why search engines' algorithms are constantly changing. They have to be adapted to technology evolutions and users' behaviors. If search engines algorithms are not made public, search engines offer some advices to webmasters in order to improve their website's ranking [6]. The problem for SEO practitioners is that results are uncertain, long to obtain, and that the SEO process is not clearly defined. Thus, the non-transparency of ranking models, the important number of interactions, and the uncertainty in terms of results make the SEO process a complex problem.

Concurrent engineering approaches permit to improve processes of multiple disciplines, especially the process of product development. The design of complex processes such as the SEO process could also be improved by using concurrent engineering approaches. As well as the process of product development, the SEO process begins by a client's requirement that the SEO practitioner must be able to understand in order to bring to him relevant solutions. These solutions have to be in harmony with the client's needs to reduce the process delay and improve results. SEO can also be considered as a concurrent and dynamic process. SEO process involves the cooperation of many distributed entities: *requirements*, *functions*, *solutions* and *constraints*. Distribution and cooperation allow SEO process' modelling by multi-agent system.

Thus, the goal of this paper is to propose a concurrent intelligent computing approach for intelligent SEO process. A model of multi-agent system for intelligent SEO process and its implementation are proposed. In the proposed model, agents are organized into communities. Four communities are proposed: community of requirement agents, community of function agents, community of solution agents and the community of constraint agents. The ranking emerge from concurrent intra-community and inter-community interactions.

The paper is structured as follows. Section 2 proposes a meta-model for SEO process. Section 3 proposes the abstract formulation of multi-agent system: the model of multi-agent system. The implementation of the proposed ASEO (Agents for Search Engine Optimization) multi-agent system is presented in the Section 4. Discussions and conclusions present and analyze our findings.

1. Proposed meta-model for SEO

To analyze the SEO process, we choose to use an engineering meta-model, which is derived from the field of product's design [7]. Indeed, the creation of a process that has the ability to adapt itself in an upgradable environment is typically a design problem [8]. This meta-model is structured in four domains and four models.

1.1. Models

- *Conceptual model*: it represents a concept or rules, in our case it can be the expression of a SEO technical concept; e.g. "an old web page will have better chance to improve its ranking than a youngest one".

- *Mathematical model*: it expresses itself from a mathematical point of view, in our case it can be a mathematical formula; e.g. "a web page with a keyword density between 3% and 8% could win two positions in one month on this keyword".
- *Computational model*: it is the representation of the mathematical model from a computational point of view; in our case it can be a simulation tool permitting to predict the ranking of a web page.
- *Experimental model*: its results are observations or tendencies; in our case it can be an experiment to compare the ranking of several web pages.

1.2. Domains

- *Customer domain*: it expresses a client's need; e.g. "I want my website to reach the first position on that keyword".
- *Functional domain*: processes have to be designed using functions; e.g. "to improve the website ranking, to increase the volume of qualified traffic, etc."
- *Physical domain*: it is solutions that respond to the functional domain; e.g. "insert relevant keywords, remove Flash language, etc."
- *Process domain*: it is constraints elements linked to the physical domain; e.g. "to be adapted to algorithms updates, to be adapted to technologies updates"

1.3. Discussion on the meta-model

The SEO process involves the cooperation between the client (the website owner) and the SEO practitioner. This multi-level information exchange is organized in four steps represented in the meta-model (Table 1). Firstly, the client formulates his requirements (1). Then the SEO practitioner transforms client's requirements and constraints (2) into functions (3). Finally when all functions are identified, the SEO practitioner finds technical solutions (4). After receiving feedback from the client, if solutions do not provide expected results, the SEO practitioner will have to reconsider all steps. The SEO practitioner's objective is to limit the number of exchange with the client while finding effective solutions to improve the website ranking with a minimum of delay. Thus, the SEO can be considered as a concurrent and dynamic process.

Table 1. Matrix of engineering modeling analysis.

	(1) **Customer Domain**	(3) **Functional Domain**	(4) **Physical Domain**	(2) **Process Domain**
Conceptual model	Conceptual model of customer domain	Conceptual model of functional domain	Conceptual model of physical domain	Conceptual model of process domain
Mathematical model	Mathematical model of customer domain	Mathematical model of functional domain	Mathematical model of physical domain	Mathematical model of process domain
Computational model	Computational model of customer domain	Computational model of functional domain	Computational model of physical domain	Computational model of process domain
Experimental model	Experimental model of customer domain	Experimental model of functional domain	Experimental model of physical domain	Experimental model of process domain

2. Proposed multi-agent system for intelligent SEO process

2.1. From SEO process properties to agent paradigm

The SEO process involves the cooperation of many distributed entities: *requirements, functions, solutions and constraints.* The distribution and cooperation between entities allow the SEO process to be modelled using a multi-agent system (M.A.S). A multi-agent system is a computerized system composed of multiple interacting intelligent agents within an environment [9] [10] [11] [12] [13]. Researchers have already proven the effectiveness of the agent's approach in solving concurrent and distributed problems [14] [15]. Indeed, agents' properties (autonomy, distribution, adaptability, flexibility, and cooperation) [16] [17] permit to manage efficiently heterogeneous, autonomous and distributed solutions. It also facilitates exchanges of information and sharing of resources between solutions. To design an intelligent SEO process, the configuration has to be focused on an efficient collaboration between its members and entities. The agent paradigm allows to design efficient cooperative and distributive information systems [18] [19] [20] [21] [22] making it a suitable tool for designing an intelligent SEO process.

Different agents are involved in the SEO process; we highlighted three properties explained below:

- **Property 1 (perception):** agents have consciousness of themselves, they have their own level of knowledge, they can share and communicate their knowledge [23], and they are able to perceive information from their environment and coming from other agents.
- **Property 2 (interaction):** agents have a degree of affinity with other agents, they can approve or discriminate a received message, and they are able to interact within their environment [17].
- **Property 3 (adaptation):** agents are able to adapt themselves to the environment's evolution; they are able to modify their interactions with other agents and to adjust their behavior [16].

2.2. Agent modeling for intelligent SEO process

Searchers [24] [25] [26] have proposed a multi-agent model that we adapted to design a distributed and intelligent SEO platform and simulate the intelligent SEO process. This multi-agent platform is called ASEO (*Agents for Search Engine Optimization*). We consider that the multi-agent system developed for the ASEO platform is described by the following 4-tuple (1):

$$M = < A, I, P, O > \tag{1}$$

where A is the set of ASEO *agents*, I is the set of *interactions between ASEO agents*, P is the set of *roles* to be played by ASEO agents, and O the *organizations* of ASEO agents into communities. According to our engineering meta-model, each agent belongs to one of these four communities: <*Requirement*>, <*Function*>, <*Constraint*> or <*Solution*>.

An agent $\alpha_i \in A$ of ASEO platform is described by the following 4-tuple (2):

$$\alpha_i =< \Phi_{\Pi(\alpha_i)}, \Phi_{\Delta(\alpha_i)}, \Phi_{\Gamma(\alpha_i)}, K_{\alpha_i} > \tag{2}$$

where $\Phi_{\Pi(\alpha_i)}$ is the *observations* function of ASEO agent α_i; $\Phi_{\Delta(\alpha_i)}$ is the *decisions* function of ASEO agent α_i; $\Phi_{\Gamma(\alpha_i)}$ is the *actions* function of ASEO agent α_i; K_{α_i} is the finite set of *knowledge* of ASEO agent α_i - the agent's knowledge is contained in its memory, it contains the decision rules.

ASEO agents and their knowledge (A is the set of ASEO agents). Each community (requirement, function, solution and constraint) defined in ASEO platform contains agents. In each community, the agents have the same formal model. An agent of the ASEO platform has five kinds of knowledge: (a) *environment knowledge* (b) *self-knowledge* (c) *knowledge of their own skills* (d) *knowledge to control the agent's activity* (e) *knowledge to interact with other ASEO agents*.

Interaction and communicational interaction into ASEO platform (I is the set of interactions). During the SEO process, there are two types of interactions between agents: (a) *Intra-communities interactions*. Intra-communities interactions can take place between agents from the same community and several results emerge from these interactions (e.g. a functional network emerges from the interactions between *function* agents; the ranking emerges from the interactions between *solution* agents.) (b) *Inter-communities interaction.* Inter-communities interactions can take place between agents from different communities. These inter-communities interactions occur between *requirement* agents and *function* agents, between *function* agents and *solution* agents, and between *constraint* agents and *solution* agents. Finally, from these interactions emerges the optimal ranking.

An interaction $\iota_i \in I$ between two ASEO agents $\alpha_s \, and \, \alpha_d$ is defined by the following 3-tuple (3):

$$\iota_i =< \alpha_s, \alpha_d, \gamma_c > \tag{3}$$

where $\alpha_s \in A$ is the ASEO agent source of the interaction, $\alpha_d \in A$ is the ASEO agent destination of the interaction, and $\gamma_c \in \Gamma$ is an act of cooperation. ASEO agents can perform eight cooperative acts (4):

$$\Gamma = \begin{Bmatrix} inform, reply, diffuse, propose, \\ ask, accept, refuse, acknowledge \end{Bmatrix} \tag{4}$$

Cooperation between ASEO agents (*e.g.* sequences of interactions) is controlled by a protocol in which a response is required for some interactions ([*ask, reply*], [*propose, accept*||*refuse*], [*inform, acknowledge*||*reply*], [*diffuse, acknowledge*]).

The Figure 1 (a) shows a typical exchange between two ASEO agents. These agents can be webpages that exchange information about their domain’s age, as shown in the Figure 1 (b).

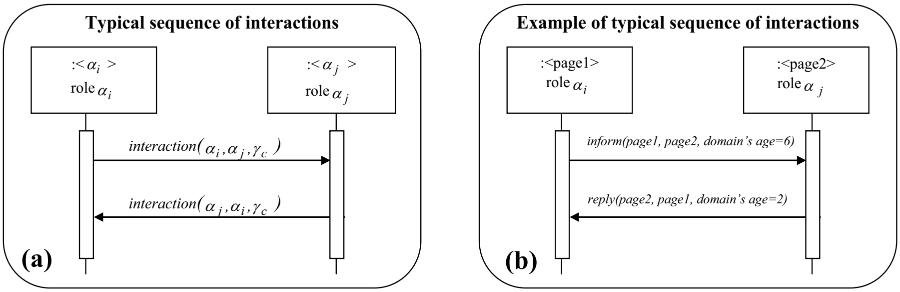

Figure 1. Diagrammatic representation of an exchange between two ASEO agents (double interaction).

Roles and organization of ASEO agents (P is the set of roles to be played by ASEO agents and O the organizations of ASEO agents into communities). Organized in four communities (*requirement, function, constraint, and solution),* respectively: $R, F, C, S \subseteq \mathrm{A}$, ASEO agents are all involved in the same activity of problem-solving: the optimal ranking. During the SEO process, networks' structures of affinities of *inter-communities* and *intra-communities* will emerge. Therefore, the coordination of agents and the self-organization of the four communities are carried out by exchange of messages.

2.3. Architecture of ASEO platform

The architecture of the ASEO platform is presented below (Figure 2):

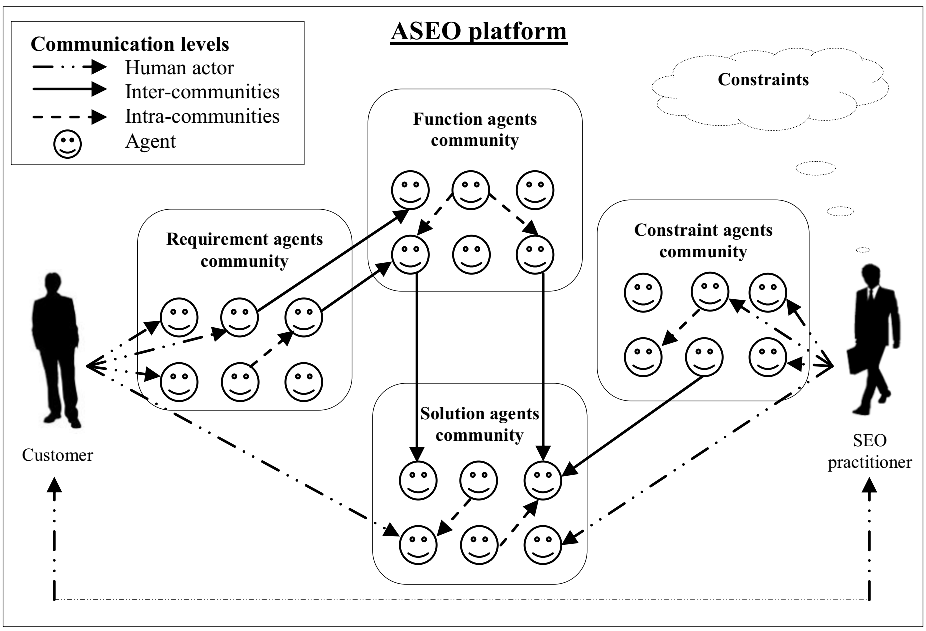

Figure 2. Architecture of ASEO platform.

According to our engineering meta-model, the ASEO platform is organized in four distributed communities of agents: *requirements, functions, solutions and constraints*. Two human actors are also involved into the platform: *customer* and *SEO practitioner*. Each of these actors and entities are presented below:

- *Customer:* requirements are expressed by the customer; to do it, he collaborates with the SEO practitioner. Indeed, the SEO process is a participative (concurrent) process.
- *SEO practitioner:* constraints as "algorithm update" or "technology evolution" are identified by the SEO practitioner. He is an expert; he decides when a constraint has to be taken into account in the SEO process. Constraints which do not modify the SEO process are not identified by the SEO practitioner.

Table 2. ASEO agents' characteristics.

ASEO agents	Description and characteristics
Requirement agent $r_i \in \{r_1, ..., r_r\}$	Interactions with: customer, requirement and function agents $\{(R\text{-}r_i) \cup F\}$ Knowledge: $\{r_i, [r_i \times (R - r_i)], [r_i \times F]\}$
Function agent $f_i \in \{f_1, \ldots, f_f\}$	Interactions with: requirement, function and solution agents $\{R \cup (F\text{-}f_i) \cup S\}$ Knowledge: $\{f_i, [f_i \times R], [f_i \times (F - f_i)], [f_i \times S]\}$
Solution agent $s_i \in \{s_1, ..., s_s\}$	Interactions with: function, constraint and solution agents $\{F \cup C \cup (S\text{-}s_i)\}$ Knowledge: $\{s_i, [s_i \times F], [s_i \times C], [s_i \times (S - S_i)]\}$
Constraint agent $c_i \in \{c_1, ..., c_c\}$	Interactions with: SEO practitioner, constraint and solution agents $\{(C - c_i) \cup S\}$ Knowledge: $\{c_i, [c_i \times (C - c_i)], [c_i \times S]\}$

The previous table (Table 2) shows the description and characteristics of each ASEO agent:

- *Requirement agents' community*: ASEO agents from this community receive requirements expressed by the customer. They interact with other agents from their community to build a network of requirement agents, and they interact with function agent community to send their requirement values (cf. $\{r_i, [r_i \times (R - r_i)], [r_i \times F]\}$ in Table 2).
- *Constraint agents' community:* ASEO agents from this community receive constraints identified by the SEO practitioner. They interact with other agents from their community to build a network of constraint agents, and they interact with solution agent community to send their constraint values (cf. $\{c_i, [c_i \times (C - c_i)], [c_i \times S]\}$ in Table 2).
- *Function agents' community*: ASEO agents from this community receive messages sent by the requirement agent community and the constraint agent community. They interact with other agents from their community to build a network of function agents, and they interact with solution agent community to send their function values (cf. $\{f_i, [f_i \times R], [f_i \times (F - f_i)], [f_i \times S]\}$ in Table 2).
- *Solution agents' community*: ASEO agents from this community receive messages sent by the function agent community and the constraint agent community. They interact with other agents from their community to build a network of solution agents, emergency of the SEO solutions (cf. $\{s_i, [s_i \times F], [s_i \times C], [s_i \times (S - S_i)]\}$ in Table 2).

Intra-communities interactions can take place between ASEO agents from the same community, and several results emerge from these interactions. Analyzing the intra-interactions between solutions agents allows the understanding of agents' local behaviors. The question is: "How to emerge an optimal ranking from knowledge and solution agent characteristics"? In the following, we focused on the simulation of the solution agents' community.

3. Analysis of the solution agent behavior

3.1. Solution agents formalization

Modeling of ASEO solution agents satisfies equations (1), (2), (3) and (4). Therefore, this sub-multi-agent system is described by the following 4-tuples (5):

$$M_s = < A_s, I_s, P_s, O_s > \qquad (5)$$

where A_s is the set of *page agents*, I_s is the set of *interactions between page agents*, P_s is the set of *roles* to be played by page agents, and O_s the *organizations* of page agents into sub-communities. The sub-communities of pages agent are defined depending on the thematic, for instance the "health" community. This one can be divided into other communities (e.g. pharmacy, surgery, etc.)

A *page agent* has the knowledge of the other pages agent in the same community. It also knows its self-characteristics. In the literature [27] [28], we have distinguished 15 characteristics which can influence the behaviors of the page agents towards the best ranking. The set of these characteristics is the following (6):

$$C = \begin{Bmatrix} c_1 = \text{domain's age};\ c_2 = \text{keyword's density};\ c_3 = \text{bounce rate};\ c_4 = \text{time spent on page}; \\ c_5 = \text{number of Google+1};\ c_6 = \text{number of Facebook likes};\ c_7 = \text{page load time}; \\ c_8 = \text{title tag length};\ c_9 = \text{meta description tag length};\ c_{10} = \text{keyword repetition in title tag}; \\ c_{11} = \text{keyword repetition in h1 tag};\ c_{12} = \text{number of backlinks};\ c_{13} = \text{PageRank}; \\ c_{14} = \text{keyword as first in title tag};\ c_{15} = \text{keyword as first in h1 tag} \end{Bmatrix} \qquad (6)$$

We suppose that each criterion has an influence on the ranking process. Three kinds of influences are distinguished: *positive, neutral or negative*. Table 3 shows the definition of positive, neutral and negative values for the criteria $\{c_8 = \text{title tag length}\}$. For instance, the domain of definition of the title tag length is decomposed into three intervals: *positive value* for the length between $50 < x < 70$, *neutral value* for the length between $30 < x < 49$ and *negative value* for the length between $0 < x < 29$ and $71 < x < 200$.

Table 3. Values' example for the criterion c_8.

Criterion	Positive value (+)	Neutral value (0)	Negative value (-)
{c₈=title tag length}	a < x < b (a=50, b=70)	a < x < b (a=30, b=49)	a < x < b (a=0, b=29) and (a=71, b=200)

Each page agent α_i can compute its positive, neutral and negative values for each criterion $c_j \in C$ and its corresponding value x in its domain of definition. Three maximum operators w_1, w_2 and w_3, respectively associated with the positive, neutral and negative measures, are used. They are subjected to the constraint (7):

$$\sum_{i=1}^{n} w_i = 1,\ 0 \le w_i \le 1,\ i = 1,2,3 \tag{7}$$

Figure 3 shows the graphical representation of these three cases. Table 4 shows the equations for each sub-domain.

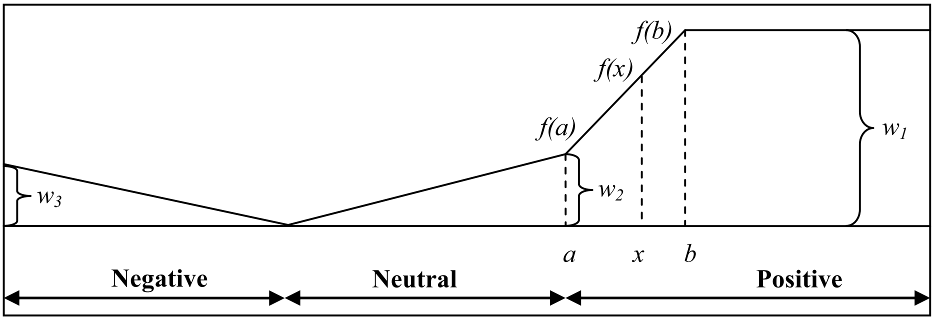

Figure 3. Graphical representation of our three cases.

Table 4. Proposed formula for our three cases.

Case	Positive	Neutral	Negative
Formula	$f(a) = w_2$ $f(b) = w_1$ $f(x) = w_2 + [(w_1-w_2) \times (x-a)/(b-a)]$	$f(a) = 0$ $f(b) = w_2$ $f(x) = w_2 \times (x-a)/(b-a)$	$f(a) = w_3$ $f(b) = 0$ $f(x) = w_3 \times (b-x)/(b-a)$

The global ranking measure is defined as (8):

$$GlobalRankingMeasure(\alpha_i) = \sum_{\forall c_j \in C} Positive(c_j^i) + \sum_{\forall c_j \in C} Neutral(c_j^i) - \sum_{\forall c_j \in C} Negative(c_j^i) \tag{8}$$

where $GlobalRankingMeasure(\alpha_i)$ is the computed global ranking measure for the page agent $\alpha_i \in A_s$; $Positive(c_j^i)$, $Neutral(c_j^i)$ and $Negative(c_j^i)$ are respectively the positive value, neutral value and negative value for the criterion j for the page agent α_i.

3.2. Behavior analysis: case study

In this section, we present a case study to analyze the SEO activity with the help of ASEO solutions agents. It is focused on the improvement of the website's ranking.

Problem definition. We completed a study in a SEO firm; clients contact this firm to improve their website's ranking on the Google search engine. A pharmacy (called later "pharmacy F") contacted this SEO firm to improve its ranking on the keyword "pharmacy + city's name", comparing to other pharmacies in the same region. The pharmacy F was in the sixth position on the Google search engine (Table 5) whereas its competitors A, B, C, D, and E were respectively in the 1st, 2nd, 3rd, 4th and 5th position. By searching "pharmacy + city's name" on Google.fr, some pages of results appeared.

We decided to ignore the pages which were not considered as pharmacy websites (e.g. web directories, yellow pages, university websites, etc.). We also decided to consider only homepages, as these pages were present on all websites. Then, we identified their characteristics using online SEO tools (SEOquake, Ahrefs, GTMetrix, SimilarWeb and Whois), as shown in the Table 5:

Table 5. List of characteristics for the six pharmacy homepages.

Page	Google's ranking	c_1 (months)	c_2 (%)	c_3 (%)	c_4 (s)	c_5	c_6	c_7 (s)	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}
A	#1	119	3.13	47	96	0	1	2.40	29	225	1	0	5	1	1	0
B	#2	81	3.89	50	132	0	0	4.50	96	143	1	1	3	1	0	1
C	#3	73	3.39	48	121	0	0	5.63	51	98	1	1	0	0	0	1
D	#4	37	1.58	51	164	0	1	6.84	33	0	1	1	1	0	1	1
E	#5	14	2.21	47	125	0	0	5.72	65	0	2	0	6	0	1	0
F	#6	10	2.91	52	153	0	0	3.74	69	0	2	0	2	0	0	0

Agentification and simulation with ASEO solutions. The set of page agents is divided in two sub-sets: page agent pharmacy and page agent non-pharmacy. Page agent pharmacy represents the homepages of the pharmacies A, B, C, D, E and F. Page agent non-pharmacy represents the pages with which the page agent pharmacy is linked to (backlinks). After having identified the characteristics and interactions between solution agents, we implemented a simulation using NetLogo, which is a multi-agent programmable modeling environment [29]. Our implementation is designed as a scalable system of page ranking. This simulation permitted us to reproduce a page ranking environment without being influenced by the time factor. Indeed, in our simulation, the ranking evolution is immediate, contrary to real search engines that need weeks to reassess a page position. Thus, a change in the characteristics value, or in the influence of a characteristic compared to others, can instantaneously reconsider the ranking process. When the simulation is over, the score of the page ranking is given according to the values of *GlobalRankingMeasure*. As we can see in the Figure 4, the ranking of our six pharmacies' homepages is similar to the Google established ranking.

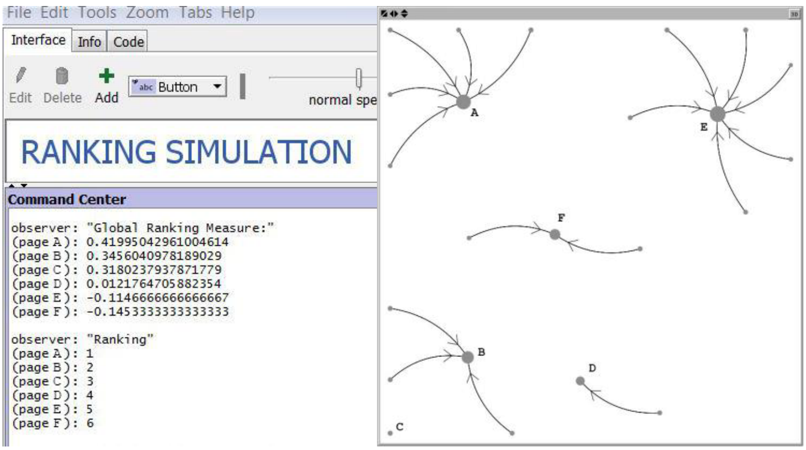

Figure 4. NetLogo simulation.

The scalability of this simulation permitted us to add page characteristics and to change their influence on the ranking process. In addition, the range of positive, neutral and negative values can be changed dynamically. Finally, this simulation permitted to identify characteristics or group of characteristics, which most affect the ranking process (as domain's age in our case).

4. Discussion and Conclusion

Designing a SEO process that is capable of functioning as intended despite changes, disturbances and adverse events on the Internet, is a new emergent topic in information processing technologies. By clarifying the relationships between different models, a meta-model is proposed to consider the SEO as a process to be designed, integrating different approaches aiming to improve and optimize it, rather than a "combination" of techniques.

In order to better understand and simulate the SEO activity, a multi-agent system called ASEO is proposed. ASEO platform helps to show some properties of SEO-design, such as: co-evolution of the criteria, emergence of a new criterion and the incompleteness of the criteria. ASEO platform revealed that changing SEO space can be fully mastered by page solution. SEO space, as a network of page solution agents, helps to overcome the cognitive limit of the SEO practitioner.

The simulation reveals that the considered criteria are pertinent. The decomposition in definition's domain for each criterion in positive, neutral and negative permitted to optimize the ranking process. The results of the simulation have shown that the comparability with the Google ranking is relevant. However, the integration of fuzzy modelling can improve the results and can help to better master the uncertainty.

In the future, the simulation will have to explore some dimensions of the interaction and the emergence, considering the other communities: requirement, function and constraints. A page agent's belief about what other page agent may believe and do, could be as useful as knowing their actual behavior to analyze some aspects of their interactions.

Acknowledgments

The authors gratefully acknowledge the financial and technical support provided by the SEO firm Orixa Media, Montbéliard, France.

References

[1] D. Hawking, Web search engines. Part 1, *Computer*, 39(6) (2006), 86-88.
[2] D. Hawking, Web Search Engines: Part 2, *Computer*, 39(8) (2006) 88-90.
[3] N. Yalçın et al., What is search engine optimization: SEO?, *Procedia-Social and Behavioral Sciences*, 9 (2010), 487-493.
[4] K. Curran, Tips for Achieving High Positioning in the Results Pages of the Major Search Engines, *Information Technology Journal*, 3(2) (2004), 202-205.
[5] M. P. Evans, Analysing Google rankings through search engine optimization data, *Internet research*, 17(1) (2007), 21-37.
[6] Best practices to help Google find, crawl, and index your site, (2015) [Online] https://support.google.com/webmasters/answer/35769?hl=en

[7] E. Ostrosi, D. Choulier, M. Kurth, A Meta-Model for Engineering Analysis in Product Design, In: J. Stjepandić et al. (eds.), *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment,* Springer-Verlag, London, 2013, pp. 545-556.
[8] S. Sagot, A-J. Fougères, E. Ostrosi, P. Lacom, Search Engine Optimization: From Analysis based on an Engineering Meta-Model towards Integrative Approaches, in *Proceedings of International Conference on Information Society (i-Society),* London, 2014, pp. 276-283.
[9] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, 1999.
[10] R.D. Beer, A dynamical systems perspective on agent-environment interaction, *Artificial Intelligence*, 72(1), (1995) 173-215.
[11] M. Wooldridge, N. R. Jennings, Intelligent agents: Theory and practice, *The Knowledge Engineering Review*, 10(02), (1995) 115-152.
[12] M. Wooldridge, Agent-based software engineering, *IEE Proceedings-software*, 144(1), (1997), 26-37.
[13] N. R. Jennings, On agent-based software engineering, *Artificial intelligence*, 117(2), (2000), 277-296.
[14] E.H. Durfee, J. S. Rosenschein, Distributed problem solving and multi-agent systems: Comparisons and examples, *Ann Arbor*, 1001(48109) (1994), 29.
[15] K. Slimani, D. Pallez, L. Médini, P. Ghodous, Multi-agent system architecture for concurrent design and manufacturing, *11th ISPE International Conference on Concurrent Engineering*, Beijing, (2004), 209-215.
[16] N.R. Jennings, On agent-based software engineering, *Artificial Intelligence*, 117 (2000), 277-296.
[17] M. Wooldridge, Agent-based Software Engineering, *IEEE Proceedings on Software Engineering* 144(1) (1997), 26-37.
[18] S.E. Lander, Issues in Multiagent Design Systems, *IEEE Expert,* 12(2) (1997), 18–26.
[19] V. Marik, D. McFarlane, Industrial adoption of agent-based technologies, *IEEE Intelligent Systems*, 20(1) (2005), 27-35.
[20] L. Monostori, J. Vancza, SRT. Kumara, Agent-Based Systems for Manufacturing, *Annals of the CIRP,* 55(2) (2006), 697-720.
[21] P. Leitão Agent-based distributed manufacturing control: A state-of-the-art survey, *Engineering Applications of Artificial Intelligence,* 22(7) (2009), 979-991.
[22] L. Cernuzzi, M. Cossentino, F. Zambonelli Process models for agent-based development, *Engineering Applications of Artificial Intelligence*, 18(2) (2005), 205-222.
[23] O. Chira, C. Chira, T. Roche, D. Tormey, A. Brennan, An agent-based approach to knowledge management in distributed design, *Journal of Intelligent Manufacturing,* 17(6) (2006), 737-750.
[24] A.-J. Fougères, Modelling and simulation of complex systems: an approach based on multi-level agents, *International Journal of Computer Science Issues,* 8(6) (2011), 8-17.
[25] E. Ostrosi, A.-J. Fougères, M. Ferney, D. Klein, A fuzzy configuration multi-agent approach for product family modelling in conceptual design, *Journal of Intelligent Manufacturing,* 23(6) (2012), 2565-2586.
[26] E. Ostrosi, A.-J. Fougères, M. Ferney, Fuzzy Agents for Product Configuration in collaborative and distributed design Process, *Applied Soft Computing,* 12(8) (2012), 2091-2105.
[27] A.J. Su, Y.C. Hu, A. Kuzmanovic, C.K. Koh, How to Improve Your Google Ranking: Myths and Reality, *Web Intelligence*, (2010), 50-57.
[28] SEO Rank Correlations And Ranking Factors 2014, (2014), [Online], http://www.searchmetrics.com/knowledge-base/ranking-factors/
[29] NetLogo, a multi-agent programmable modeling environment, (2015) [Online] https://ccl.northwestern.edu/netlogo/index.shtml