

Restricted and Oriented Transformation In Autonomous Self-Interactive Information Systems

Hanh Pham

*Department of Computer Science
State University of New York at New Paltz
pham@cs.newpaltz.edu*

Abstract

This paper proposes a framework for autonomous self-interactive information systems, where each single information element is represented by a semi-active agent, and the autonomous information transformation is carried out through the interactions between agents representing information units. A concept of interaction distance and an interaction control mechanism are proposed to make the information transformation oriented to the system policy and user goals. Computing powers consumed by agents for their interaction are reduced by activating agents just before their interactions.

1. Introduction

In traditional information systems most of information items, such as documents, video, or audio, are formed manually and remain unchanged until further user actions. In modern information systems as information items are much more dynamic, being created, added, changed, and removed or moved frequently, managing these systems can be very difficult. Searching[8], collecting[6], transforming[2], and integrating information[1,9] to meet user tasks can also be very costly and time-

consuming [7,10]. Therefore, our goal is to build autonomous information systems where information items can be transformed autonomously.

In this paper we propose a framework on autonomous self-interactive information systems (ASIIS), where information transforms by itself through the self interactions between information items. Each information item can interact with each other by themselves without user or third-party actions. Each single information item is represented by a semi-active agent and the self-interaction between information items are carried out through the interactions between the agents which represent them. However, if the information items-agents are freely interacting their transformation can grow exponentially and the system would not be able to handle the agents and their interactions[5,6]. Therefore, we develop an interaction control mechanism to restrict the interactions allowing only interactions which may produce the desired information transformations which lead to the user goals. This control mechanism is based on a concept called interaction distance to measure how information items are suitable for each particular interaction type. The interaction between given items would occur only if their distance meet the given conditions.

The rest of this paper is organized as follows. Section 2 provides a formal description of the autonomous self-interactive information system ASIIS. In section 3, the concept of interaction distance is explained. The basic laws of self-interaction in ASIIS are described in section 4. The interaction control mechanism is depicted in section 5. Finally, several conclusions are given in section 6.

2. Formal Description of ASIIS

In ASIIS, the smallest meaningful information items are called information elements or IE. For example, a paragraph on a specific topic can be an IE. We have a set of information elements $E = \{e_1, e_2, \dots, e_N\}$ in the system. Each information element can consist of several data elements or DE which are undivided data and can be in different formats: text, image, audio, video, fields in databases. Data elements can be taken out or be generated from different element bases EB_1, EB_2, \dots, EB_X . Each data element base EB_i has its subdomain Q_i . Thus, correspondingly, we would have X subdomains Q_1, Q_2, \dots, Q_X for data elements. The domain of all data elements in ASIIS will be:

$$Q = Q_1 \cup Q_2 \cup \dots \cup Q_X .$$

Where, for examples, Q_1 can be a set of phrases, Q_2 can be a set of images, ..., Q_X can be a set of songs.

The next level of information organization in ASIIS consists of information units or IU. In the system we have a set of units $U = \{u_1, u_2, \dots, u_N\}$. Each unit consists of information elements (IE) and somewhat is equivalent to a file in the file systems. The difference is that a unit in ASIIS can be dynamically defined based on the unit structure.

The information organization of a unit u_i is represented by the unit structure $S(u_i)$ using a

hyper-marking language such as XML, it specifies the type of the unit, location, time related attributes and most importantly the specification for the components and their relationships including the requirements for selecting the components. For example:

```
S(ui) = {
  <head>
    <title> TL <requirement> rT
    <type> TP <requirement> rP
    <time> TM <requirement> rM
    <time> LC <requirement> rL
  </head>
  <body>
    <item1> p1 <requirement> r1
    ...
    <item Q> pQi <requirement> rQi
  </body>
}
```

The unit structure $S(u_i)$ can be used to define the unit dynamically. For each component position there can be a competition among candidate information elements where their satisfaction of given condition for that position may be compared to define the winner. This dynamic feature can also be used to define the replacement of a component in case that component of the unit is damaged or missing.

The contents of a unit is represented by the unit index $I(u_i)$ which is a set of keywords, with weights, to which the given unit relates to:

$$I(u_i) = \{(k_1, \beta_1), (k_2, \beta_2), \dots, (k_z, \beta_z)\},$$

where, $k_v \in \dot{K}$, $\beta_v \in D_s$, $v=1..z$, D_s is a domain for weight coefficients. \dot{K} is the set of keywords $\dot{K} = \{k_1, k_2, \dots, k_m\}$ which can be added with the new ones as the system progresses. The relationships between the keywords are given in the keyword matrix:

$$MK = \{\phi_{1,1}, \phi_{1,2}, \dots, \phi_{1,m}, \phi_{2,1}, \dots, \phi_{2,m}, \dots, \phi_{m,1}, \dots, \phi_{m,m}\}$$

Each information unit can be represented by a semi-active generic agent (SGA), similar to [4],

which unlike the ones in [4] the SGA is usually passive and will be activated just before the unit interacts with the others. The condition and mechanism for activating the SGA are described in more details in section 5. Each SGA controls the unit structure $S(u_i)$, the unit index $I(u_i)$, as well as the unit participation in collections.

The highest level of information organization of ASIIS consists of information collections (IC). In the system we have a set of collections $C = \{c_1, c_2, \dots, c_M\}$. Each collection is a list of units

and is somewhat equivalent to a directory in the file systems. The difference is that a collection in ASIIS can be dynamically defined based on the collection structure, similarly to the way a unit can be dynamically defined based on its structure.

Thus, an autonomous self-interactive information system is characterized by a triplet $\{C, U, E\}$ where C is the set of information collections, U is the set of information units, and E is the set of information elements.

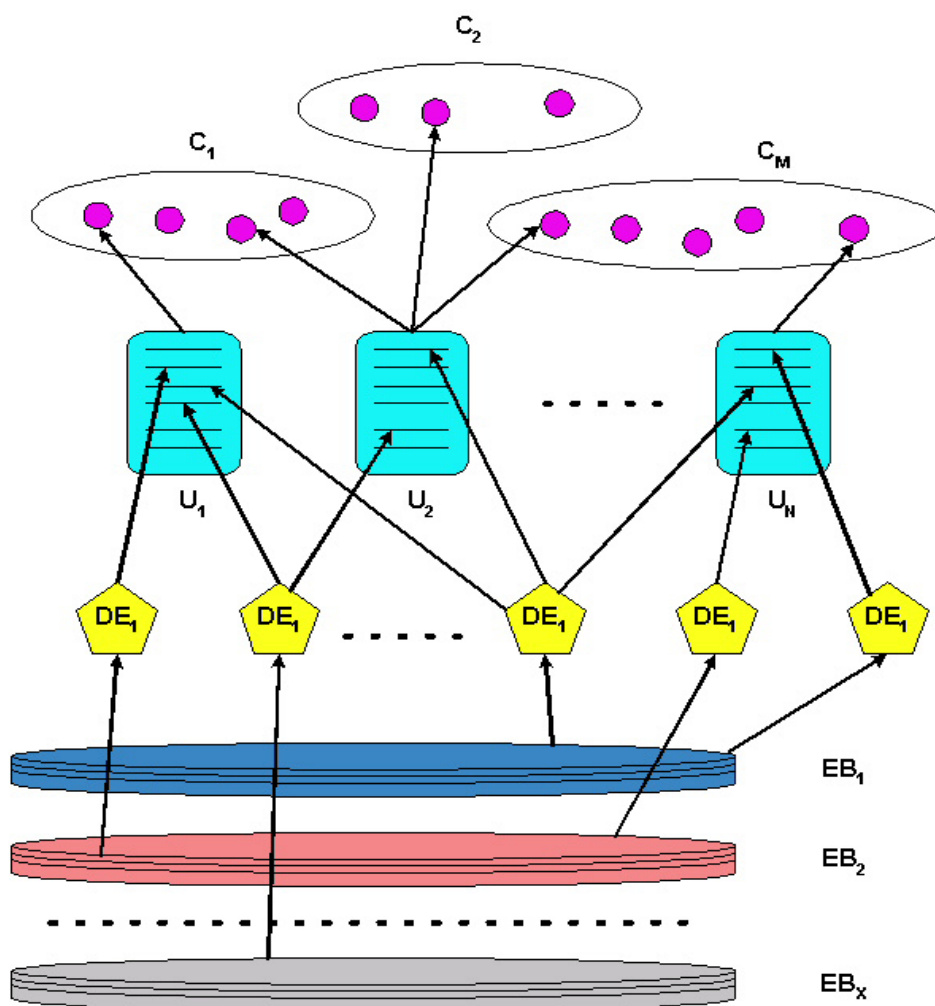


Figure 1

The self-interaction of information in ASIIS is carried out through the interactions of the semi-active generic agents or SGA. When a SGA_i , representing unit u_i , interacts with another SGA_j , representing unit u_j , the structure of u_i and u_j may be changed or the units may be eliminated, or some new unit may be created, depending on the type of the interaction.

When the structure of a unit $S(u_i)$ is changed the unit contents $I(u_i)$ will be automatically redefined. Thus, after their interaction the indexes of these two units u_i and u_j may also be changed. Since the participation of a unit in a collection depends on the unit index, these changes will lead to the changes in the contents of the collections and therefore will lead to the changes of the collections. Hence, the self-interaction of the SGAs in fact triggers the transformation of the information in the whole system.

3. Interaction Distance

The self interaction of information in ASIIS is carried out through the interactions of the semi-active generic agents or SGA. If there are N units in the system then we would have a set of N agents $A = \{SGA_1, SGA_2, \dots, SGA_N\}$.

Assume that the autonomous self-interactive information system has a set O of interaction operations or interaction types:

$$O = \{\theta_1, \theta_2, \dots, \theta_H\}$$

where, an interaction operation θ_k , $k = 1..H$, between agents in a set A_{IN} will produce a new set of agents A_{OUT} , where some agents may be eliminated, created, and changed into a new state. Examples of interaction types are union, intersection, etc, are studied in [3].

If we let the SGAs interact freely with each other in all dimensions, i.e. using all interaction types from O at the same time without any restriction the system would quickly be

overloaded and crashed. Therefore, the SGAs should interact with restrictions.

We propose a concept called interaction distance. The interaction distance $\delta_{i,j}$ between two agents: SGA_i , representing unit u_i , and SGA_j , representing unit u_j , measures how information in unit u_i and in unit u_j are suitable for a particular interaction operation θ_k . The interaction between given units should occur only if their distance meets a given condition. The interaction distance is defined by a distance function $\omega_i(A_{IN})$, $A_{IN} = \{SGA_x, SGA_y\}$, which is various depending on the type of the interaction.

In general, we have:

$$\theta_k: A_{IN} \rightarrow A_{OUT} \mid \omega_k(A_{IN}) \cong \varepsilon_k$$

where, ε_k is the condition for the interaction θ_k to occur and ω_k is the distance function which measures the degree or possibility of agents in A_{IN} to interact with each other by this θ_k type of interaction.

For example, θ_k can be a union interaction, and we may have:

$$\begin{aligned} A_{IN} &= \{SGA_x, SGA_y\} \\ A_{OUT} &= \{SGA_x, SGA_y, SGA_w\} \end{aligned}$$

The structure of the united unit u_w of the units u_x and u_y is represented by a newly created agent SGA_w :

$$S(u_w) = S(u_x) \cup S(u_y)$$

We can use the unit indexes to define the interaction distance function as follows. Assume that:

$$\begin{aligned} I(u_x) &= \{(k_1, \beta_{x_1}), (k_2, \beta_{x_2}), \dots, (k_z, \beta_{x_z})\} \\ I(u_y) &= \{(k_1, \beta_{y_1}), (k_2, \beta_{y_2}), \dots, (k_z, \beta_{y_z})\} \end{aligned}$$

Then, the interaction distance function can be defined as:

$$\omega_i(SGA_x, SGA_y) = \sqrt{(\beta_{x_1} - \beta_{y_1})^2 + \dots + (\beta_{x_z} - \beta_{y_z})^2}$$

The interaction occurs when its condition holds, that is:

$$\omega_k(A_{IN}) \leq \varepsilon_k$$

where, ε_k is a given number. In the given example, this indicates that the given units, represented by agents in A_{IN} , contain certain level of the same keywords and therefore probably address certain common topics and can be combined into one unit u_w .

4. Self-Interaction Laws

The self interactions between the units or agents in ASIIS should be self-restricted and are based on the following laws:

Eligibility Law:

If $\omega_k(SGA_x, SGA_y) \cong \varepsilon_k$, where \cong is a comparison operator and ε_k is a given threshold derived from user goal and system policies, then SGA_x or u_x is "**interactable to**" SGA_y or u_y by type θ_k and we denote it as $\{SGA_x \xrightarrow{\theta_1} SGA_y\}$ or $\{u_x \xrightarrow{\theta_1} u_y\}$. Otherwise, SGA_x or u_x is "non-interactable" to SGA_y or u_y and denoted as $\{SGA_x | \xrightarrow{\theta_1} SGA_y\}$ or $\{u_x | \xrightarrow{\theta_1} u_y\}$.

Interaction Law:

SGA_x and SGA_y can interact with each other and we denote as:

$$\{SGA_x \xleftrightarrow{\theta_k} SGA_y\}$$

if and only if: SGA_x or u_x is interactable to SGA_y or u_y by type θ_k and vice versa SGA_y or u_y is interactable to SGA_x or u_x , that is:

$$\{SGA_x \xrightarrow{\theta_1} SGA_y\} \text{ or } \omega_k(SGA_x, SGA_y) \cong \varepsilon_k \text{ and } \{SGA_y \xrightarrow{\theta_1} SGA_x\} \text{ or } \omega_k(SGA_y, SGA_x) \cong \varepsilon_k$$

Notice that the interaction functions $\omega_k(SGA_x, SGA_y)$ can be different from one interaction type to another. However, if $\omega_k(SGA_x, SGA_y)$ is a symmetric function, that is, $\omega_k(SGA_x, SGA_y) = \omega_k(SGA_y, SGA_x)$, then

obviously we would have $\{SGA_x \xleftrightarrow{\theta_k} SGA_y\}$ if just SGA_x is interactable to SGA_y by type θ_k or just SGA_y is interactable to SGA_x .

Exclusion Law:

At a time, an agent SGA_i , $i=1..N$, representing an information unit u_i , can participate in *only one interaction type* from $O = \{\theta_1, \theta_2, \dots, \theta_H\}$.

At a time, an agent SGA_i , $i=1..N$, representing an information unit u_i can interact with *at most one agent* from the N agents in $A = \{SGA_1, SGA_2, \dots, SGA_N\}$.

Thus, there will be several competitions in defining an interaction for an agent:

- with which agent from the N agents in $A = \{SGA_1, SGA_2, \dots, SGA_N\}$ the given agent will interact with? For example, as shown in Figure 2, unit u_1 is "interactable" to units: $u_2, u_3, u_5, u_7, u_{10}, u_{11}, u_{12}, u_{14}$, and u_{16} . However, the candidates for unit u_1 would only be u_2, u_5 , and u_{10} , which are "interactable" to unit u_1 . Then, which unit from u_2, u_5 , and u_{10} , unit u_1 should interact with?
- through which operation from $O = \{\theta_1, \theta_2, \dots, \theta_H\}$ the interaction will be carried out?

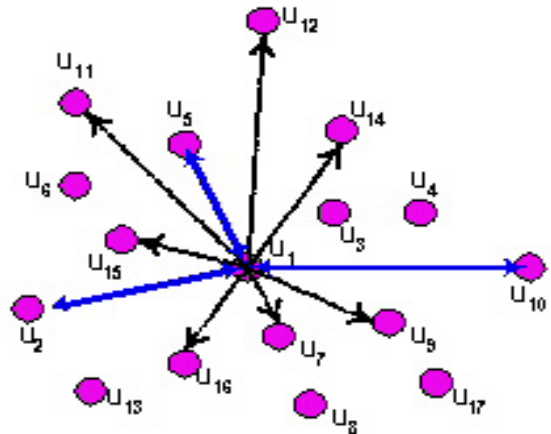


Figure 2

5. Restricted Transformation

When the information items or agents are freely interacting with each other, the information units can be divided and combined and the number of units can grow exponentially. The system would not be able to handle the agents and their interactions. Therefore, we develop an interaction control mechanism called RIC to restrict the interactions allowing only interactions which may produce the desired information transformations to meet the user goals. This control mechanism is based on the concept of interaction distance described above. The distance is used to measure how information units are suitable for a particular interaction type. The interaction between given units should occur only if their distance is within a given threshold.

Assume that there are N units in the system, that means we would have a set of N agents $A = \{SGA_1, SGA_2, \dots, SGA_N\}$. The set of interaction operations is $O = \{\theta_1, \theta_2, \dots, \theta_H\}$.

Step-1: Setting the restrictions based on user goal and system policies. The weights of interaction types are defined to adjust the distances between the interaction types:

$\{\theta_1:\alpha_1, \theta_2:\alpha_2, \dots, \theta_H:\alpha_H\}$, where $\sum_{i=1}^H \alpha_i = 1$.

These weights represent the preferences of the user and the system on interaction types. For example, when the user goal is to find the related documents which address the same topics then let's say θ_1 , the "intersection" should be the most preferred interaction type. In that case we may have $\{\theta_1:\alpha_1=0.9, \theta_2:\alpha_2=0.1, \dots, \theta_H:\alpha_H=0\}$.

The user goal should also have effects on defining the interaction distance functions $\omega_1, \omega_2, \dots, \omega_H$ and the values of the thresholds $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_H$ for the interaction law.

Step-2: Defining the distance. For each agent SGA_i , representing unit u_i , the adjusted interaction distances to other agents can be defined for $\forall SGA_j \in A$ and $SGA_j \neq SGA_i$, $j=1..N$:

- for type θ_1 : $\alpha_1 \times \omega_1(SGA_i, SGA_j) = d_{i,j,1}$
- ...
- for type θ_H : $\alpha_H \times \omega_H(SGA_i, SGA_j) = d_{i,j,H}$

Step-3: Define eligible interaction options. Use the interaction laws and the thresholds $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_H$ to define the interactable units. The network of interactable units can be represented by a graph G where each node is a unit and each arc shows a possible interaction between the two units it connects. Since there can be several possible interaction types between the two units there can be several arcs between the two nodes in this graph.

Step-4: Define for each unit-agent SGA_i an interaction partner agent SGA_m with the closest interaction distance by a type θ_k , that is to find a SGA_m , where:

$$d_{i,m,K} = \underset{j=1,N}{\overset{k=1,H}{MIN}} \{ \alpha_k \times \omega_k(SGA_i, SGA_j) \}$$

Step-5: After the interaction partner agent SGA_m for agent SGA_i is defined these two agents will be activated and perform the interaction. During the interaction these two agents should be temporarily removed from the graph G.

Thus, depending on the user goal and the system policies the weights of interaction types $\{\theta_1:\alpha_1, \theta_2:\alpha_2, \dots, \theta_H:\alpha_H\}$ as well as the thresholds $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_H$ can be adjusted so that the transformation information through the interactions between units would be *oriented* toward the user goal. For example, if the user goal is to create documents or units which cover other units contents then the system would increase the weight α_s for union interaction type θ_s which will make that type of interaction

dominating. Then, the system has two options: (i) increase the threshold ε_s , which will increase of the number of units eligible for uniting with each other by that type, but those units may be more divergent than the second case; (ii) decrease the threshold ε_s , which will narrow the number of units eligible for uniting with each other by that type, however, those units may be more similar than the ones in the first case.

Hence, if the user goal is to combine only very the similar documents then the option (ii) should be selected, if the user goal is to combine as many as possible documents which may address some same topics then the option (i) should be chosen.

6. Conclusions

We have provided a framework on restricting information interaction for autonomous self-interactive information systems. In order to reduce the scope of information interaction in ASIIS we have proposed a concept of interaction distance and described a mechanism for interaction control which changes the scope of information interaction by regulating the number of interactions of each type and the kinds of the participating information units. The described restrictions, which are defined based on user goal and system policies, guide the autonomous transformation of information in ASIIS and make it oriented to user goal and system policies.

References

[1].Bergamaschi S., Beneventano D., "Integration of Information from Multiple Sources of Textual Data" in Matthias Klusch (Ed.), Intelligent Information Agents, pp.53-77, Springer 1999.

[2].Bouchon-Meunier B et. al (edited), "Intelligent Systems for Information

Processing: from Representations to Applications", Elsevier, 2003.

[3].Hanh Pham, "Self-Processing of Heterogenous and Dynamic Information Through Agent Interactions in AuInSys", in the Proceedings of the 2005 International Conference on Information and Knowledge Engineering and Data Fusion (IKE+FUS'05), pp.302-308, Las Vegas, USA, June 20 - 23, 2005.

[4].Hanh Pham, Yiming Ye, Chirakkal Easwaran, "Agent Visibility and Invisibility in Real-Time Agent-based Business Systems", in the Proceedings of the 2002 International Conference on Internet Computing Volume III, pp.1067-1073, 2002.

[5].Jensen D, Dong Y, Lerner B, McCall E, Osterweil L, Sutton Jr S., and Wise A. *Coordinating agent activities in knowledge discovery processes*. In Proc. of WACC, 1999.

[6].Kargupta H., Hamzaoglu I., and Stafford B.. Scalable, distributed data mining-an agent architecture. In Daryl Pregibon et. al. (editor), 3rd Conference on Knowledge Discovery & Data Minin. AAAI Press, August 1997.

[7].Klosgen and Zytkow, "Handbook on Data Mining and Knowledge Discovery", Oxford, 2002.

[8]. Lancaster F.W., *Indexing and Abstracting in Theory and Practice*, The Library Association, London, 1991.

[9].Suzuki G., Iizuka, and Kasuga S., "Integration of Keyword-Based Source Search and Structure-Based Information Retrieval", pp.149-158, in Heterogeneous Information Exchange and Organization Hubs, (Bestougeff et al edited), Kluwer Academic Publishers, 2002.

[10]. Van Rijsbergen C. J., *Information Retrieval*, Second Edition Butterworths, London, 1979.