



**A tutorial report for SENG 609.22  
Agent Based Software Engineering  
Course Instructor: Dr. Behrouz H. Far**

## **Interface Agent**

**Presented by: Shuangshuang Zhang**

### **Abstract:**

Interface agents are semi-intelligent systems which assist users with daily computer-based tasks. Recently, various researchers have proposed a learning approach towards building such agents and some working prototypes have been demonstrated. Such agents learn by `watching over the shoulder' of the user and detecting patterns and regularities in the user's behavior. Despite the successes booked, a major problem with the learning approach is that the agent has to learn from scratch and thus costs some useful time. We will introduce two interface agents to solve such problem.

## Collaborative Interface Agents

We propose a collaborative solution to the problem above. While a particular agent may not have any prior knowledge, there may exist a number of agents belonging to other users who do. Instead of each agent re-learning what other agents have already learned through experience, agents can simply ask for help in such cases. This gives each agent access to a potentially vast body of experience that already exists. Over time each agent builds up a trust relationship with each of its peers analogous to the way we consult different experts for help in particular domains and learn to trust or disregard the opinions of particular individuals.

Collaboration and communication between various agents can take many different forms. This paper is only concerned with those forms that aid an agent in making better predictions in the context of new situations. There are two general classes of such collaboration.

***Desperation based communication*** is invoked when a particular agent has insufficient experience to make a confident prediction. For example, let us suppose that a particular agent  $A_1$  has just been activated with no prior knowledge, and its user receives a set of new mail messages. As  $A_1$  doesn't have any past experience to make predictions, it turns in desperation to other agents and asks them how their user would handle similar situations.

***Exploratory communication***, on the other hand, is initiated by agents in bids to find the best set of peer agents to ask for help in certain classes of situations. We envisage future computing environments to have multitudes of agents. As an agent has limited resources and can only have dealings with a small number of its peers at a given time, the issue of which ones to trust, and in what circumstances, becomes quite important. Exploratory communication is undertaken by agents to discover new (as yet untried) agents who are better predictors of their users' behaviors than the current set of peers they have previously tested.

For agents to communicate and collaborate they must speak a common language as well as follow a common protocol. We assume the existence of a default ontology for situations in a given domain (such as electronic news, e-mail, meeting scheduling, etc). . This allows agent creators the freedom to decide which types of ontologies their agents will understand. the protocol for collaboration is designed to be flexible, efficient and non-binding. We briefly present the protocol below.

***Registration:*** Agents wishing to help others register themselves with a "Bulletin Board Agent" whose existence and location is known to all agents. While registering, agents provide information regarding how they can be contacted, what standard domains they can provide assistance in, what ontologies they understand and some optional information regarding their user. Every agent registering with a bulletin board agent is given a unique identifier by the bulletin board.

***Locating peers:*** Agents wishing to locate suitable peers may query bulletin board agents. An agent querying a bulletin board agent need not itself register with that bulletin board. Queries to a bulletin board agent can take many different forms depending on the type of information required. This allows agents to locate suitable peers in the most convenient way. For example, an agent's user may explicitly instruct it to ask a specific user's agent for help in dealing with certain types of situations.

**Collaboration:** Collaborative communication between agents occurs in the form of request and reply messages. An agent is not required to reply to any message it receives. This leaves each agent the freedom to decide when and whom to help. Any request always contains the agent's identifier, the agent's contact information (for replies), the ontology used in the request, and a *request identifier (reqid)* generated by the agent issuing the request. The *reqid* is necessary since an agent may send out multiple requests simultaneously whose replies may arrive out of order.

Analogously every reply always contains the replying agent's identifier and the *reqid* used in the request.

The types of requests and their associated replies are presented below.

**Situation level collaboration:** When a situation occurs for which an agent does not have a good prediction, it sends off a *Request-for-Prediction* message to its peers. A prediction request contains all the features of the situation which the agent issuing the request wishes to divulge. This allows the requesting agent the freedom to withhold sensitive or private information. An agent receiving a prediction request may choose to ignore it for have no good prediction or too busy to respond. If however, an agent decides to respond to a prediction request, it sends back a response.

**Agent level collaboration:** An agent may send its peers a *Request-for-Evaluation* request. An evaluation request is sent when an agent wants to know what some of its peers think about a certain agent in terms of being able to model their users in particular classes of situations. An evaluation request contains the identifier of the agent to be evaluated (designated as the *target agent*) and the particular class of situations for which the evaluation is needed.

## Autonomous Interface Agents

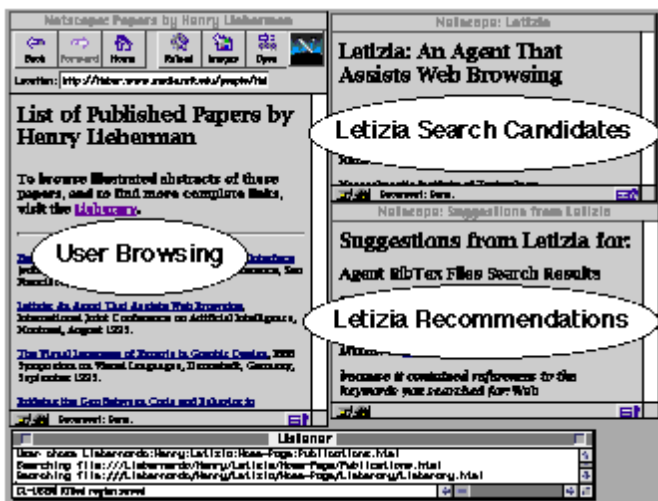
An autonomous interface agent can make real-time suggestions for web pages that a user might be interested in browsing. Autonomous interface agents lead to a somewhat different design style, brought on by the possibility that the agent may need to interact with the interface while the user is also interacting with the interface. The user may or may not be aware of the agent's activities at any given moment. The agent may discover a condition that might interest the user and independently decide to notify him or her. The agent may remain active based on previous input long after the user has issued other commands or has even turned the computer off.

In order for an agent to be considered both an interface agent and autonomous, it follows that there must be some part of the interface that the agent must operate in an autonomous fashion. The user must be able to directly observe autonomous actions of the agent and the agent must be able to observe actions taken autonomously by the user in the interface. Concretely, the user will see interface elements that appear to "move by themselves" in response to input that the agent appeared to "see for itself" rather than having been explicitly instructed

Letizia is an autonomous interface agent that treats search through the Web space as a continuous, cooperative venture between the user and a computer search agent. Letizia records the URLs chosen by the user and reads the pages to compile a profile of the user's interests. A simple keyword-frequency information retrieval measure is used to analyze

pages. Letizia is always active, searching the Web space that is "nearby" the user's current position, in parallel with the user's browsing activity. Letizia then uses Netscape's own interface to present its results, using an independent window in which the agent browses pages thought likely to interest the user. The following sections will explain Letizia in more detail, and some of the rationale for its design.

Letizia's usual interface consists of three Netscape windows. By default, the left-hand window is reserved for user browsing activity. The user may browse in this window in a normal manner, and can completely ignore any agent activity. The two windows on the right side are under control of the agent. The top window displays search candidates, those pages which Letizia is considering to recommend to the user. The bottom right window displays those pages that Letizia actually decides to recommend to the user, passing Letizia's tests for user interest. The user may choose to continue browsing with either his or her own selected pages, or Letizia's suggestions, at any moment in time.



A traditional search engine is conversational -- the user asks a question, then waits for the answer. After the answer is received, the user continues the browsing process, either accepting the recommendations of the agent, or doing unrelated browsing.

In autonomous agents like Letizia the view of Web browsing as query-based information retrieval is replaced by a view of Web browsing as a real-time activity. The goal is not to retrieve the "best answer" in any absolute sense, but to make the best use of the most limited and valuable resource -- the attention of the user.

### Design principle for autonomous interface agents

Autonomous interface agents lead to a somewhat different design style. Below, we discuss some interesting aspects of interface design.

**Suggest rather than act:** the agent makes suggestions rather than has responsibility for solving the whole problem. In these situations, the agent does not have to make the absolute best choice in order to be useful, but only offer a suggestion that is "better than nothing", or a "good enough guess".

Web browsing is a good application for an autonomous interface agent. When the user is facing a choice between a set of unknown Web links, a recommendation made by the

agent can simply increase the chances slightly that the user will make the best choice of what to look at next. Suggesting relies on the user to examine choices by the agent and makes browsing a cooperative activity between the user and the agent.

***Take advantage of information the user gives the agent "for free":*** The actions taken by the user in the user interface constitute information that the system can use to infer the goals and interests of the user "for free" -- that is, without requiring a separate interaction. Even the simplest of interactions that lasts just a couple of seconds can be disruptive enough to the user's workflow that he or she won't bother to do it.

Search engines have an absolutely minimal interaction -- bring up a single-line query form, the user types a word, then clicks "enter" -- but it is disruptive enough to the browsing process that it is impractical to use the search engine after every few browsing steps. Letizia's automatic inference of terms that would otherwise have to be typed into a search engine makes bringing up recommendations based on each browsing step possible.

***Take advantage of the user's think time:*** A disadvantage of the traditional conversational interface is that the agent remains idle when the user is thinking about what input to provide next. Running the agent autonomously while the user is thinking takes advantage of compute time that otherwise be wasted. This is especially important in search or exploration tasks when the idle time can be used to provide independent exploration of the search space. Available time can also be used to deepen the results of previous searches or interests, and other lower-priority tasks that might provide useful but would also potentially risk disrupting the primary interaction.

***The users attention may be time-shared:*** A consequence of running an interface agent autonomously is that the agent cannot assume it has the full attention of the user in the way a traditional, modal interaction does. The user may find themselves providing input to the agent without being explicitly aware of where the input is going. The user may find output happening on the screen at a time that they may not expect it. This can be both good and bad. It can make more information available "at a glance", provide a better sense of context, and reduce the need for context-switching.

***Autonomous interface agents may have a different tradeoff between deliberation and action:*** For a traditional search engine, the judgment as to whether a particular document should be considered relevant is critical, because once the search engine returns a list of suggestions for a given query, it never gets another chance. By contrast, when Letizia needs to decide to recommend a document, the decision is less critical because it will usually have future opportunities to make a better decision as to how relevant that document is relative to other documents that it might propose as alternatives.

## Conclusion

In Collaborative interface agent, experienced agents can help a new agent come up to speed quickly as well as help agents in unfamiliar situation. Thus each agent also learns which its peer is a reliable 'expert' vis-à-vis its user for different types of situation.

The ability of a software agent to operate the same interface operated by the human user, and the ability of a software agent to act independently of, and concurrently with, the human user will become increasingly important characteristics of human-computer interaction. Agents will observe what human users do when they interact with interfaces,

and provide assistance by manipulating the interface themselves, while the user is thinking or performing other operations. Increasingly, applications will be designed to be operated both by users and their agents, simultaneously.

We've illustrated some of the principles behind autonomous interface agents with a description of Letizia, an agent for assisting a user browsing the Web. Letizia is an interface agent in that it observes input that the user directs at the browser, not at the agent. It reports its results by automatically browsing Web pages in a Netscape window. It is autonomous in that it browses concurrently with the user, searches and analyzes Web pages while the user is browsing, and displays recommendations continually, without explicit user request or other intervention. We think that autonomous interface agents like Letizia will be the next step in the evolution of interfaces beyond direct manipulation.

## Reference:

1. <http://www.media.mit.edu/~lieber/Lieberary/Letizia/Letizia-Intro.html>
2. <http://www.media.mit.edu/~lieber>
3. Robert Armstrong, Dayne Freitag, Thorsten Joachims and Tom Mitchell, WebWatcher: A Learning Apprentice for the World Wide Web, in AAAI Spring Symposium on Information Gathering, Stanford, CA, March 1995.  
<http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/project-home.html>
4. Oren Etzioni and Daniel Weld, A Softbot-Based Interface to the Internet, Communications of the ACM, July 1994.  
<http://www.cs.washington.edu/homes/weld/pubs.html>
5. Dent, L.; Boticario, J.; McDermott, J.; Mitchell, T.; and Zabowski, D. 1992. A personal learning apprentice. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 96-103. San Jose, California: AAAI Press.
6. Gruber 1993  
Gruber, T. 1993. A translation approach to portable ontology specification. *Knowledge Acquisition* 5(2):199-220.
7. Lai, Malone, & Yu 1988  
Lai, K.; Malone, T.; and Yu, K. 1988. Object lens: A spreadsheet for cooperative work. *ACM Transactions on Office-Information Systems* 5(4):297-326.