3
Related Work

In this section, we present some background related with our study. It consists of approaches that have been proposed in order to promote software reuse in agent-oriented software engineering. Since as mentioned above, the agent-oriented software development process, that emphasizes the design and construction of software systems by using reusable agent components, is a cost effective approach to the software development.

3.1.
Literature Review

There have been some initiatives on exploiting software reuse in agent technology [8][7][35]. [40][41][42] have discussed the need of agent toolkits and libraries with agent parts, complete agents, and pre-connected agent societies, to define and generate individual agents and agent-based systems. Researchers have seen the fact of using agents as an advanced form of software components capable to exhibit interesting characteristics like autonomy [9][39][40]. They suggest agents could be adopted as reusable artifacts of software and the reasonable expectation is that they should exhibit at least, the same properties of software components. Reusability can be seen from dissimilar viewpoints because there are dissimilar types of artifacts that users would like to reuse [8].

The foremost obstacle to reuse agent-oriented software artifacts successfully is that users cannot locate and retrieve desired artifacts from a large collection quickly, easily and sometimes no properly. It is difficult retrieving artifacts which a software engineer wants to modify based on the given specification if the artifact does not fix with his requirements. Locating reusable software artifacts is frequently supported by artifact repository systems. Software repositories provide an interface for users to search the artifacts stored.

[17] presents a conceptual model for specification of agents developed in JADE and Jadex [25], that uses semantic web services for realization of its

actions, aiming to facilitate the discovery and monitoring the same web. Nevertheless, this model does not include nor does it explicate how an automated agent will be executed or how the specification of protocols is for the interaction and composition of agents. [51] defines a model for the development of single agents and a way for them to work together. It provides a method to implement agent capabilities, action plans and interactions. Besides, since it is based on a component middleware, it provides location services, communication platform and knowledge (ontology) development. The model could be improved using a variety of languages for describing agent capabilities and system specifications and experimenting with component platforms other than the used one. [22] proposes a reuse-directed software design process that carried out over a framework created by them, which provides an environment where a multi-agent system is created as an organization of agent-based components which are generated from a repository by design activity of repository agents. A protocol is proposed [22] for retrieving suitable candidates of an agent based component or an organization of agent-based components, which is a subsystem of an application system to design. A component programmer is supported to retrieve a suitable component or a module consisting of components designed by the repository agents. But there is no explanation how the model for agent-based components is, and it is just mentioned that there is no possible retrieval of the components through their relationships among them.

Like many other information repository systems, browsing- and querying-oriented schemes have long been used as the basic techniques for users to locate reusable artifacts. Therefore, it is desirable to structure the repositories for making the search, retrieval and browsing mechanisms as fast and appropriate as possible in order to make the access to the repository attractive. Faceted classification, known as faceted navigation or faceted browsing, was proposed by [63], that relies on facets which are extracted to describe features about artifacts. Features serve as artifact descriptors, such as the artifact's functionalities and implementation details. Numerous facets classify components, but there are usually a lot fewer facets than there are potential attributes. On occasion faceted search is as well raised to explorative search and guided search, because users are given choice to select features available for search. This helps the users to accomplish his search goals rapidly and efficiently. Faceted classification and retrieval has proven to be very effective in retrieving reuse artifact from repositories, but the approach is labor intensive.

There has been a previous effort to build reuse systems roughly classified in (i) those which draw information only from the structure of some documents that provide information about the artifacts. How no semantic knowledge is given, the reuse tool attempts to characterize the document rather than understand it; and in contrast (ii) those that are knowledge-based aims at understanding the queries and functionality of the components before providing an answer. Some web-based component repositories use taxonomies to structure components to facilitate retrieval [84]. These taxonomies provide an effective method to locate generic components (domain independent) that are well-known to users and correspond fine with their intuition. Nevertheless, when a user is uncertain of the taxonomy, the keyword-based search provided in these systems is not sufficient due to there are for example various terms to describe the same artifact. To go beyond this obstacle, the IR system needs another structure for the data with which it works. Approaches that use structural information to assist with the search have been proposed.

Over the last years, advances in artificial intelligence have provided considerable progress toward solving the problems of information access through the following three domains: techniques for multi-agents systems, information retrieval, and recommendation systems. In the first domain, there is a large community developing architectures and techniques for interaction and communication among agents. An important focus for the IR domain has been the development of methods for coping with the heterogeneous and dynamic information space represented by the WWW and the impact of new retrieval devices and interfaces. Nevertheless, IR systems do not traditionally take into consideration the individual searcher, preferring instead to focus on the development of global retrieval techniques rather than those adapted for the needs of the individual. As a result, recommendation systems research has focused on the interaction between IR and user modeling in order to provide a more personalized and proactive retrieval experience and help users to choose among retrieval alternatives and refine their queries. Thus, a virtual organization of agents with specific tasks such as profile acquisition, web searching and recommendation making could be made-up.

There are proposals of information retrieval, which are based on semantic web concepts and multi-agent systems. In existing systems, query and retrieval are done on centralized basis and do not consider semantics of the query. Since the repositories are usually distributed, diverse and autonomous, there is a need for a framework capable of providing an efficient integrated and semantically enriched query retrieval mechanism over multiple repositories. [5] employs a domain ontology to search for web pages that contain relevant information to each concept in the domain of interest. The search is then constrained to a specific domain to avoid as much as possible the analysis of irrelevant information. A multi-agent system was developed to make the whole process asynchronous, concurrent, intelligent and distributed. [43] proposes a multi-agent based architecture for querying multiple disjoint and heterogeneous distributed digital repositories. The agents use ontology for handling queries semantically and therefore are capable of providing results which are effective compared to basic text search. Thus, the proposed architecture is an effective and efficient solution for querying multiple data repositories which are distributed in nature using the concepts of semantic web. [12] proposes a methodology for constructing multi-agent systems that uses an existing generic reusable architecture as the basic agent-oriented artifacts. The main goal of the methodology is to extract the knowledge necessary for initializing a particular agent from the generic architecture. However, the knowledge is limited to plans, actions and interactions with other agents and the rest of characteristics of agents with architecture BDI and other architecture are not taken into account. In addition, the proposed methodology uses some of the notations from the existing object oriented methodologies and is used in an iterative process.

[33] combines concepts of an online store with agent oriented programming to describe a mechanism and tool by enabling users to share, search and deploy scripts and agents developed in JIAC agent framework [31]. The scripts represent plans or plan elements, which can be deployed to agents that are capable of interpreting these scripts. Web- and API-based interactions allow the integration in the common workflow of multi-agent system developers, setting a high value on socializing the agent developer, not the agents.

[56][57] present an exploratory study of the development of a family of agents, in which there are techniques to allow building agents using reusable

assets that address the domain variability. This agent's family was built using a software product line (SPL) architecture with fine-grained variability, which consists on beliefs, goals, plans and plan parameters. The development of the case study is described by the following points: variability types that they dealt with, how current multi-agent system methodologies are able to document variability and provide mechanisms to enable software reuse, and adopted implementation techniques. Therefore, customized agents can be derived by configuring them according to a user specification in order to enter in an existing MAS. These authors also discuss issues related to the lack of techniques, in design and implementation levels, to develop MASs according to traditional software engineering principles, such as modularity, reusability and maintainability. Nevertheless, in these papers, they did not implement the use of SPLs in order to dynamically change an agent configuration. All variability has a compilation binding time, therefore once the agent is derived, it cannot change its behavior to adopt a configuration with other variants. Therefore, using SPL architectures improves the development of agents that can change their configuration at runtime to adapt themselves according to the current context.

Moreover, none of these MAS methodologies address the development of SPLs, and consequently, do not provide notations to express agent variations. In the context of MAS, the investigated approaches do not provide explicit support to specify and model the SPL architecture and its respective component. Nevertheless, most of the SPL approaches provide useful notations to model the agent features but, none of them completely covers their specification. Agent technology provides particular characteristics that need to be considered in order to take advantage of this paradigm.

We study besides how semantic technologies can support the identification and retrieval of relevant agent component from open-access, different structured, very large repositories. Semantic web supports semantic inference in searches. Recent the web embraces more social semantics to improvement search. [47] introduces an ontology-based technique for requirement analysis of multi-agent applications through the reuse domain models. An ontology-driven technique for requirement analysis of multi-agent applications by reusing domain models representing common and variable requirements of a family of multi-agent applications in a domain. The requirements of a particular application are represented in models obtained through the instantiation of the ontology that expresses the knowledge of the methodology for multi-agent application engineering that integrates techniques for design and implementation of a specific application in a domain family. Part of a case study models a multi-agent application for recommendation of touristic packages through web usage mining and collaborative filtering.

While many repositories exist or are under development, some problems continue to persist. Remaining familiar with all the components in such a repository presents a significant challenge to software developers. Recommendation systems have emerged as solutions, but up to now, the majority has been tied to the web. Several of them represent mature technology delivered as part of commercial systems, such as Amazon.com[2] and eBay[3]. Recommendation systems help users efficiently overcome the problem of content overload by filtering irrelevant items or services, when users search for desired information.

In traditional RSs the user profile is keyword-based which, to some extent, involves only those items that match specific keywords in the user's preferences [77]. This paper surveys the state-of-the-art of the techniques which are used to semantically enhance user modeling within the recommendation phases. It introduces the web usage mining as an approach to capture and model the user's behavioral patterns as the user interacts with the system. An approach to overcome losing useful information, inaccuracy in the recommendation items and user dissatisfaction, is to extracting and utilization semantic-based information from the domain and incorporating them within the stages of the personalization process. The semantic web mechanisms incorporate additional knowledge about the user's preferences into recommendation process, specifically in the user profile, by acquiring information from the pages a user visits or items he searched and rated during his practice with the system. Therefore, this provides more accuracy and flexibility to the personalization processes [77]. In addition, since semantic web technologies enable greater accuracy for the meaning of the underlying information, it helps to develop semantically enrich descriptions of user interests for further improvement of web personalization techniques [34].

[2] http://www.amazon.com/

[3] http://www.ebay.com/

3.2.
Final Remarks

We realize which difficulties exist to reusing software agents, such as the lack of mechanism or standards to retrieve agent-oriented artifacts, and the lack of architectures and programming languages that allow multi-agent systems to implement all the features of agents. The existing component repositories are not applicable to provide the reuse of heterogeneous agent-oriented artifacts, since they do not support a model of agents that includes for example their behaviors and interactions neither the retrieval according agent characteristics and, they do not considerate commonality and variability of agents in a certain domain in order to increase the reusability.