KNOWLEDGE ENGINEERING AND KNOWLEDGE DISSEMINATION IN A MIXED-INITIATIVE ONTOLOGICAL FRAMEWORK

by

Jurika Shakya

B.Sc. (Comp. Sc.), University of Saskatchewan, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Interactive Arts and Technology

SIMON FRASER UNIVERSITY
Fall 2005

APPROVAL

Name: Jurika Shakya

Degree: Master of Science

Title of Thesis: Knowledge Engineering and Knowledge Dissemination in a Mixed-Initiative Ontological Framework

Examining Committee: Dr. Rob Woodbury
Chair

____

Dr. Vive Kumar, Senior Supervisor

____

Dr. Marek Hatala, Supervisor

____

Dr. Allyson Hadwin, External Examiner,
Department of Education,
University of Victoria

Date Approved: Nov.25/2005

Abstract

This research postulates that a theory-centric mixed-initiative approach to systems design is critical for the success of technology-enhanced learning environments. It explores a formal ontological mechanism to represent the underlying educational theory. It presents a design for a mixed-initiative system, named MI-EDNA, which recognizes and utilizes explanation-aware opportunities for the dissemination of self-regulatory knowledge. MI-EDNA formally captures the theory of Self-Regulated Learning (SRL) in an ontological framework. It uses Description Logic and Production Rules as reasoning mechanisms to enable learners to reflect and regulate on their learning process. Using a model-tracing methodology, this research successfully maps learner interactions onto tactics, strategies, and phases/states that have been identified within the realms of SRL. Based on this mapping, MI-EDNA engages learners in a mixed-initiative interaction, formalizes recognition of system initiation opportunities, and provides a scaffolded learning environment to sustain sharing of learning experiences across domains and across learners.

Keywords:

knowledge representation, knowledge engineering, ontology, Web Ontology Language (OWL), Description Logic (DL), production rules, knowledge dissemination, mixed-initiative interactions, model of Self-Regulated Learning (SRL), educational theories, system-initiated interactions, system-oriented initiatives, shareable learner experiences, user modeling, tutoring systems

This thesis is dedicated to my parents for all their love, support and encouragement.

Acknowledgments

This thesis represents part of my journey in this life and reflects the continual support, encouragement and guidance that I received from everyone around me.

My sincere and foremost gratitude goes to my senior supervisor - Dr. Vive Kumar. Without his continual encouragement, optimistic perspective in new research ideas, and patience, this thesis would not have been possible. I really appreciate the countless discussions that provided me with intellectual insights and motivation.

My thesis committee was the ideal committee. Thanks to Dr. Marek Hatala for providing technical feedback from different perspective and guidance in my research. I appreciate Dr. Allyson Hadwin for accepting to be my external examiner and further demystify certain aspects of educational psychology such as the scaffolding/fading in SRL. Your valuable feedback has provided many improvements to this thesis and my research.

I owe a great deal to Mayo Jordanov and Liam Doherty for their tremendous help this summer. Without their help, MI-EDNA would not have existed on-time. MI3, Mixed-Initiative Interaction group was a great inspiration for my research. Cheers to everyone in the MI3 group, Mayo Jordanov, Liam Doherty, David Brokenshire, Pat Lougheed, Sam Menon, and Shilpi Rao. Aside from the discussions and paper writing craze, all your sarcastic analytical approaches to research provided me with humour and insight.

In this interdisciplinary research, I appreciate all the people involved in Learning Kit that provided me with much understanding of the world of educational psychology. I would especially like to thank Dr. Phil Winne, Dr. Dianne Jamieson-Noel, and Carmen Gress. Also, would like to extend my thanks to everyone involved in Laboratory of Ontological Research, which provided me opportunities to explore and discuss further in my knowledge of ontology. I really appreciated the discussions with Dr. Dragan Gasevic on ontology.

I would like to thank Andrew Hendriks for enlightening me LaTex, and all the hidden information on being a graduate student; Stephanie Dayes for reviewing the thesis; and Info-Media Lab friends for the fun atmosphere.

I admire and appreciate the encouragement from some of my previous work supervisors. I am grateful to Ron Koenderick, Andrew Pratt and Ken Mayhew for their support.

Especially, my deepest thanks to Jeff Allen for his patience, understanding and laughter. Akoma, Sameer, Shristi, Binendra and Tannis for more laughter. Mom and dad for being the best parents and for bringing more laughter.

Bahini, Hasina, Kalyani and Zaria for keeping me grounded and bringing fun attitude. Chris, Stephanie, Mike, Cindy, Geoff, Baljeet, Kreeti, Ashok, Davis, Frank, and Regina for being wonderful friends and tempting me with occasional distractions. Raz, Sabita, Rajen, Nina, Ang doka, Shree, Gyanu, Sadhana and Sameer for adventures that always refreshed me.

Lastly but not least, Edna Mode from the cartoon, The Incredibles, that gave me inspiration to view my research with an attitude.

Contents

List of Tables

List of Figures

List of Programs

Chapter 1

Purpose

Knowledge acquisition, sorting, storing, and retrieval are natural phenomena for human beings, in a given situation. Accuracy of knowledge, relevance of knowledge diffusion, and significance of inferred knowledge differs from person to person. However, humans are capable of learning, understanding, recognizing and reacting to this difference in knowledge across persons. Humans gather information through various sensors and build their knowledge about everything around them. Experts take a step further and take time to understand and learn the intricacies of the domain specific knowledge. When humans communicate, plan, and work, they are selective about which piece of knowledge to employ and which piece of information they need to process in order to make decisions, to take initiatives, to communicate, or to react to a certain situation. Typically, experts are selective and logical in their method of utilizing their knowledge. These are some of the key capabilities that computing technologies of recent times have attempted to replicate; in many cases, they have failed miserably. This is particularly true in the domain of educational technology.

Since 1970s, researchers have been attempting to employ computational artifacts to help humans learn better. The field of Educational Technology has been investigating the applicability of the means of computation to education, and has come out with a number of approaches to computer-oriented teaching and learning. However, these approaches do not quite capture the level of sophistication that a human expert would casually and logically exercise. For example, a human expert would adapt their teaching environment by employing a suite of teaching strategies to suit the needs of the learners, the needs of the content, and the needs of the infrastructure. *This research strive to mimic a significant portion of human's teaching expertise and to capture a significant portion of human's learning expertise,* *in a shareable form.*

Ideally, quality of learning should transcend boundaries, national, institutional, and even individual. However, the ability to maintain the infrastructure requirements of learning environments, to measure the impact of cognitive resources on learning, and to record and disseminate social impact on learning have played a major role in constraining policy-makers to accept an imbalance in the landscape of learning as a fact of life. Countries have evolved *beacons of learning*, in both K-12 and Higher Education institutions, that are considered as centres of excellence in learning. Countries also recognize the disparity of learning opportunities for people who do not have access to these beacons. In an effort to balance such a geographically distributed nature of the quality of education, nations are investing in technological innovations that not only support the evolution of learning methodologies in the beacons, but also disseminate these methodologies across the nation. Once validated for their utility, these technological innovations can be deployed in any learning environment, across varied learning institutions, irrespective of the economical, political, and social status of a country.

Academic communities and research groups have engaged in a number of research frontiers to estimate the impact of infrastructure, cognitive tools, and social experiences on learning and how to sustain the quality of learning [58], [93], [82], [27], [59], [52]. The Learning Kit[1] is one such research frontier not only allows educators to observe how institutions enhance opportunities to improve the quality of the learning environments but also to provide opportunities for learners to reflect on their understanding of the cognitive and meta-cognitive processes involved in learning. *We contend that the utility of this approach will result in a wide-ranging and positive impact on educators' efforts to sustain learning.*

Educational Technology has been explored extensively in the last few years under the aegis of electronic learning, intelligent tutoring, and performance support systems. It involves investigations from a number of fields in Social Sciences and Computing Sciences. The demand for educational technology originates from a variety of sources such as traditional academics, continuing education, and corporate training. This has resulted in considerable fluctuations in the development of educational technology methodologies. This variance in the focus of educational technology has lead to the development of a wide range of research angles. Over the last couple of decades, these technological solutions have

[1]http://www.learningkit.sfu.ca/

been employed in learning environments successfully (and not so successfully). Computer-Aided Instruction (CAI) brought the content of learning to an interactive electronic format. Computer-Mediated Communication (CMC) techniques enhanced human-computer interaction between the educational system and the learners. Computer-Supported Collaborative Learning (CSCL) and Computer-Supported Cooperative Work (CSCW) explored the social nature of learning. Intelligent Tutoring Systems enabled educational systems to deliver informed and pedagogically sound instructions. While these research areas proliferated the use of technologies in learning environments, they could not directly address the needs to sustain learning across domains, across learners, across institutions, and across geographical boundaries. An obvious side effect of such fluctuating research efforts is the non-standard and disjoint mappings between technology and learning experiences. *This research is an attempt to quantify the needs to sustain learning across these entities and to better map the technological solutions to learning experiences.*

In summary, the central purpose of this research is to advocate a proven technique to formally capture the underpinnings of educational theories; to provide the backbone knowledge structure for the system to be able to initiate interaction based on theoretical foundations; and for the educational learning tool to be able to initiate and sustain communication with the human learner.

1.1 Objectives

Progression of application of artificial intelligence technologies and the rise of interest in educational technology creates a perfect intersection for my research. Presently, this intersection provides a loose coupling between educational theories and the associated technologies. As a result, most system-oriented educational interaction, including feedback, lack a tight integration with the underlying educational theory. One of the main objectives being tackled in this thesis is the application of a theory-centric approach to improve the learning ambience in educational tools.

This research will aim to provide a methodology to capture domain specific knowledge, to manage the knowledge in the ontological framework, to provide a basis for system initiation in mixed-initiative knowledge dissemination, and to evaluate the representation and dissemination of knowledge.

1.1.1 Knowledge Engineering

How do systems capture knowledge? How is the knowledge managed once it is captured? Is there a need for knowledge filtration? How is the knowledge organized in the system? What knowledge is required to make the system explanation-aware? Knowledge Engineering, a complex process involving knowledge acquisition, design, and management, is a key contributing area to educational technology that addresses these questions.

The motivation for this research is centered around capturing domain-specific knowledge in the system for the purpose of guiding learners to regulate their own learning. This necessitates that the very knowledge on how students regulate their learning process needs to be formally encoded. Exploring knowledge engineering is an essential aspect of this research to address questions such as: Why do we need knowledge representation? How can we formally represent educational theories? What knowledge representation scheme will sustain the quality of learning?

The **first objective** in this thesis is to provide an ontological framework as a knowledge representation methodology in the domain of education.

1.1.2 Knowledge Dissemination

Knowledge Dissemination concerns how knowledge is diffused to the learners. Most technologies attempt to capture information about the domain and about the learners. Knowledge dissemination questions the method and reasoning behind the propagation of the captured information among learners. What is the knowledge that system is disseminating? Why is the system disseminating the knowledge? How can a system validate such dissemination?

The source of learning in education has changed drastically in the past decade. Learning and knowledge gain is not only limited to reading books, classroom interactions, and socializing with families and friends, but also includes online interaction and sharing online material.

How do the users interact online? How do the users process the information and interact with the material online? How do the systems use this information on how the users are interacting? The purpose of capturing the knowledge of learner interaction, and the domain knowledge is to be able to retrieve and disseminate the knowledge along with explanations based on the ta With the knowledge represented in the ontological framework, the knowledge of learner interactions can be inferred to get a deeper understanding of their learning process.

The **second objective** is to recognize and utilize theory-centric and explanation-aware knowledge dissemination opportunities.

1.1.3 Mixed-Initiative Interactions

Aside from capturing and disseminating knowledge in a particular domain, an interesting aspect of this research is the degree of participation of the user and the system. How much of the system-initiative actions can be triggered? How much of the learner-initiated operations can be used in system-oriented guidance? How much collaboration can occur between the learner and the system? How much control can be negotiated between the learner and the system to achieve any particular goal? These questions lead to the exploration and application of mixed-initiative interactive systems.

Learning is supplemented through interaction and communication. Thus, to enhance and to captivate the learners in the learning process it is quite appropriate to incorporate mixed-initiative interactive systems.

How can the system guide learners to regulate learning? How can the learners regulate their learning through system interaction? How can the learner/system take control to help learners achieve a higher degree of regulation in learning? These questions promote the appeal in designing a mixed-initiative educational system.

The **third objective** is to design a mixed-initiative system and analyze the conditions under which it can be operational.

1.1.4 Domain

Educational technology has evolved at a steady pace over the past couple of decades. The domain of education is rather massive, involving many interrelated models associated with educational activities. This research focuses only on one of these models–a meta-cognitive model of regulated learning.r

Curriculum for any course or program in any institute is not simply a compilation of material. Rather, curriculum design is a far more complex process involving planning, design, implementation, validation, and management through different academic entities, taking into account various requirements, resources, cost, and quality constraints. Representing curricular knowledge at such coarser levels poses significant interoperability, feasibility, and applicability issues. Similarly, representing curricular knowledge at finer levels (e.g., representing offloading strategies in meta-cognitive models) also poses similar concerns.

The **fourth objective** is to capture a domain model in a formal representation.

1.2 Scope

The overarching scope of this research involves a) knowledge engineering of tasks in domains such as reading, composition, and problem-solving; b) building a model of the self-regulatory capabilities of learners; c) evaluating the influence of mixed-initiative interactions and interfaces; d) developing a cognitive model of the self-regulatory skills of the learner; e) exploring the effects of co-regulated learning within self-regulated learning; f) verifying and validating the underlying self-regulation model; g) providing a common ontological framework for geographically distributed learners and instructors in a blended online learning environment; and h) explanation-aware modeling and scaffolding.

Specifically, this thesis explores ontological representations of online content, content-oriented interactions, learner characteristics, time, teaching tactics, teaching strategies, and self-regulatory phases. Then, it advocates how to instantiate the assertional knowledge in the ontology, automatically or semi-automatically. Further, it employs the utility of reasoners based on Description Logic and Productions Rules to recognize regulatory behaviour of learners and initiation opportunities for mixed-initiative interactions. Our approach is evaluated with respect to the degree to which learner interactions can be mapped onto the models of self-regulation.

1.3 Thesis Organization

The detailed structure and organization of the thesis is as follows:

Chapter 2 presents a literature review. It reviews and summarizes related work and research areas from the fields of knowledge representation, mixed-initiative interaction, and educational theories.

Chapter 3 describes the architecture details of a prototype system. The overall architecture is discussed and the implementation details of each of the components is presented.

Chapter 4 discusses the evaluation data, methodology used for evaluation, and an interpretive analysis of the results.

Chapter 5 concludes the thesis with an impact analysis of this research, extrapolation of the results to a larger-scale computational curriculum model, and future research directions.

Chapter 2

Literature Review

This section reviews *three* fundamental areas of research that are essential for the conception, design, development, and application of theory-centric systems in domain of education. The areas being reviewed are: Knowledge Representation, Mixed-Initiative Interactive Systems, and the Educational domain.

Many systems in Intelligent Tutoring (ITS) are built with minimal foundational connectivity with educational theories or social theories of human interaction processes. These systems employ a variety of knowledge representation schemes such as symbolic rules, fuzzy logic [49], Bayesian networks [43], neural networks, case-based reasoning [31], and even some hybrid approaches [73], without explicit theoretical connectivity between the knowledge that is represented and the interactions of the learners. Many researchers have advocated knowledge representation schemes for ITS systems that hinted at the need for a theoretical basis to model learner interactions [94], [35], [68]. Of late, ITS has employed machine learning, dialogue based communication, and planning systems with explicitly represented theories of mixed-initiative interactions [1], [2], [12], [7], [13], [30] that add a sense of naturalness [35] to the represented interaction knowledge. This section presents reviews on each of these areas and highlights attributes from these research areas that influence my research.

2.1 Knowledge Representation

The essence of knowledge representation [55] [71] is to represent the knowledge intended for processing by computers. The knowledge representation involves formally capturing, storing, and manipulating the information. Davis et. al [16] [15] takes a critical approach by defining knowledge representation to consist of five fundamental roles: surrogate, ontological commitment, fragmentary theory of intelligent reasoning, medium for efficient computation, and medium for human expression. Based on these five perspectives, the representation language is expected a) to be sophisticated enough to capture aspects and relations among these perspectives; b) to be able to embed theories of intelligent reasoning; c) to be able to integrate concepts from different domains; d) to be able to formally represent concepts and relations to provide computability and satisfiability; and e) to be able to reflect on the real world. We will briefly present an overview of Description Logic, Ontology and semantic web, and Production Rules, and how they contribute to the construction of knowledge in this research.

2.1.1 Description Logics

Minsky's Frames [62] and Quillan's semantic network [74] present a functional approach to knowledge representation. They lack structural expressivity and it is difficult to represent knowledge because of their vagueness and inconsistencies with the knowledge constructs. A transition that led researchers away from semantic network led to a more well-founded terminological logic based language called the Kl-One [10]. Kl-One became the founding language for many of the knowledge representation languages to follow, including Description Logic (DL). Conceptual graphs [87] evolved from semantic networks and logic based existential graphs. They enabled a way of representing conceptual structures very closely related to semantic networks. Similarly, in the field of database management, object-oriented programming, semantic data modeling, and other class-based formalisms were developed for specific types of knowledge representation. All these knowledge representation formalisms and languages are closely related to each other, and contributed to the development of Description Logic.

Description Logic (DL) is considered one of the important logic-based knowledge representation languages designed for expressing knowledge about concepts and relationships. The basic building blocks of DL formalism are *concepts*, *roles*, and *individuals*. Complex concepts are defined using constructors such as *intersection, union, negation, existential restriction, value restriction, number restriction, inverse role, transitive role,* and so on. Terminological axioms, such as *subclass, equivalent, sub-property, equivalent property, same as, disjoint, different individual as, inverse of, transitive property* are used to name complex concepts and to state subsumption relations between the concepts.

DL consists of two main components: knowledge base and reasoning engine [53]. The knowledge base is divided into "TBox" and "ABox". The **TBox** contains intensional knowledge, the definition of a new concept in terms of other previously defined concepts. Declaration of logical equivalence, which amounts to providing both sufficient and necessary conditions for classifying a concept is the characteristic feature of DL knowledge bases. This intensional knowledge (TBox) is usually thought not to change with time. However, the **ABox**, which contains the extensional/assertional knowledge, is usually thought to be contingent, or dependent on a single set of circumstances, and therefore subject to occasional or even constant change [6]. The ABox knowledge is specific to the individuals of the domain of discourse. DL[1] is considered an important formalism unifying and giving a logical basis to the well known traditions of frame-based systems, semantic networks and KL-ONE-like languages, object-oriented representations, and semantic data model systems.

Significant differences of Description Logic with respect to its ancestors are characterized by a) restriction on the set of constructs in such a way that subsumption would be computed efficiently, possibly in polynomial time - e.g., CLASSIC [8]; and b) complete algorithms for expressive languages - FaCT [39]. One significant difference is also the reasoning tools that are available for use with DL, unlike semantic network and object-oriented data models.

With the expressive power of description logic, there has been ongoing research in the development of DL reasoners. Of the many reasoners that exists, some of the well known reasoners that support DL are Pellet[2], Racer[3],and FaCT++[4]. All these reasoners for DL are implemented on tableau-based decision procedure for general *TBox* and *ABox*. TBox reasoning such as *satisfiability of concept, subsumption hierarchy, and classification* and ABox reasoning such as *consistency, instance checking, and retrieval* are essential in the design of ontologies, the integration among ontologies, and the development life cycle of ontologies.

A number of DL systems have been developed. Some well-known DL systems are CLASSIC [10], Loom [57], FaCT [39], and RACER [32]. Description logic systems facilitate

---

[1] http://www.dl.kr.org/

[2] http://www.mindswap.org/2003/pellet/index.shtml

[3] http://www.racer-systems.com/products/download/index.phtml

[4] http://owl.man.ac.uk/factplusplus/

the development of knowledge bases by detecting inconsistencies in the knowledge representations. Horrock et.al [40] presents a comparison of some of these systems with respect to the DL features. Some education-oriented systems that utilize description logic are reported in [89] and [72]. The role of description logics in the semantic web and their use in web ontology language standards have been significant [3], [41], and [21].

2.1.2 Ontology and Semantic Web

Ontology is a formal specification of knowledge in a domain. It formalizes conceptualizations [28], [87]. It captures not only the commonalities among different conceptualizations in the domain but also formally establishes differences among those conceptualizations. In this sense, we contend that one should focus on the process of capturing conceptualizations in the ontology rather than just the commonalities. In a simplified sense, ontology provides an extendable and shareable framework to capture a common vocabulary in a domain. It includes machine-interpretable definitions of basic concepts in the domain and the relations that exist among them [67]. Presently, ontology is one of the popular knowledge representation techniques in AI.

Formally, ontology consists of *entities, relationships, properties, instances, functions, constraints, rules,* and *other inference procedures.* The power of ontologies rests with the ability to represent knowledge explicitly (as concepts, properties, and constraints); to encode semantics (as meta-data, rules, and other inference procedures); and to allow for a shared understanding of the represented formal knowledge within and in-between humans and the machines.

The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries[5]. Assuming that ontologies promote the use and the extension of a common formal conceptualization in each domain, one may assume that simply employing ontologies in web-based systems would realize the goals of Semantic Web. Unfortunately, the world of Semantic Web is much more complicated than to be solved by such a simplistic notion. As we mentioned earlier, the centrality of ontology is in the process of capturing conceptualizations in the ontology. In a community of users interacting in a semantic web application that revolves around a common ontology, it is inevitable that inconsistencies arise in the ontology among multiple users

[5] http://www.w3.org/2001/sw/

over a period of time. Maintaining such inconsistencies in the ontology is quite intractable and remains the foremost challenge in Semantic Web.

A recent surge in semantic web research has resulted in the evolution of a W3C standard - Web Ontology Language (OWL)[6]. OWL enables the definition of domain ontologies, sharing of domain vocabularies, and the representation at different levels of granularity. From a formal perspective, axioms and constructors in OWL capture the DL reasoning in terms of class consistency and consumption, in addition to other ontological reasoning. OWL includes in its specification three levels of increasing expressivity and complexity. OWL-DL is based on description logic with a reasonable level of expressivity and computable satisfiability. OWL-Lite is a simpler subset of OWL-DL. OWL-Full, which extends OWL-DL with additional constructs and RDF extension, is most expressive but has acute computational complexity.

There are different types of ontologies.

- *Domain Ontologies* capture the knowledge related to a particular type of domain. E.g. Wine ontology [7]
- *Upper Ontologies* are related to several domains and are not referred to a particular one. E.g. SUMO [8]
- *Application Ontologies* contain all the necessary knowledge to model a particular application in or across domains. E.g. Airfare [9]
- *Structural Ontologies* capture the structure that bound the representational entities in any given domain without stating what should be represented. E.g. SUO-IFF [10]

With the ever-increasing development and use of ontologies in various domains, aspects of ontology mapping, ontology reuse, and ontology integration are critical to the stability of ontological sustenance. Ontological engineering emerged as a field to cater to this specific need. Knowledge acquisition, representation and management of the information are the essential elements of ontological engineering. Along with the research of ontology

[6]http://www.w3c.org/TR/owl-features/
[7]http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml/
[8]http://reliant.teknowledge.com/DAML/SUMO.owl/
[9]http://www.daml.org/ontologies/365
[10]http://suo.ieee.org/IFF/

development guidelines [67], [38], there has been movement and discussion on ontology engineering and standardizing some of the best practices by the research community. The World Wide Web Consortium (W3C)[11] has started a focus group to tackle issues around practical deployment, engineering guidelines, ontology/vocabulary development practices, educational material for ontologies, and effective demonstrations designed for semantic web deployment. Regarding the development and maintenance of ontologies in OWL, some of the best practices are published in [78] and [29].

Instructional design methodologies have been used to build ontology-aware educational systems [63], [9], [46], [20]. These methodologies explicitly connect students' task ontologies with their goals, their cognitive states, their interaction with the system, and the pedagogical knowledge. Other researchers [4], [17], [66] have developed instructionally well-designed task ontologies as part of an overall framework for web-based information systems, where the learner activities, the domain model, and the educational strategies/goals are independently represented and semantically connected.

2.1.3 Production Rules

One of the prevalent methods of representing knowledge is in the form of rules. Production Rules represent heuristics for certain actions to be triggered based on conditions. These condition-action pairs define the condition that has to occur for the action to take place. Thus, production rules can be viewed as IF-THEN rules, where there may be more than one *if* condition paving the way for different actions. Thus, rules act like a WHENEVER-THEN statement. The inference engine always keeps track of rules that have their conditions satisfied, and thus rules could immediately be executed as and when they become applicable. In the case when a set of production rules becomes eligible for execution under a specific condition, the set is called the *conflict set.* Only one of the rules from this conflict set of eligible rules will get executed in the current iteration. When such a rule is selected and the corresponding action is executed, it is termed as *conflict resolution.* An interpreter matches the antecedents of the rules with observations in the domain, and "fires" consequent actions until a problem is solved. One of the most widely used and efficient algorithms for Production Rules is called Rete [24]. Some key characteristics related to conflict resolution strategy that one would have to pay attention to in designing rulebases are *refraction,*

[11]http://www.w3.org/2004/01/12-swbpd-charter

recency, specificity, and explicit priority.

Knowledge is almost always incomplete and uncertain. An expert system uses uncertain or heuristic knowledge to tackle these problems. Usually, expert systems separate domain specific knowledge from more general purpose reasoning and representation techniques. This important feature of expert systems lessens the complexities of solving the problems. Inference engines are, in most cases, designed as general purpose processors of the underlying knowledge. Expert system shells provide the inference engine, a user interface, an explanation system, and sometimes a knowledge base editor. Explanations can be generated by tracing the line of reasoning used by the inference engine. Using shells to write expert systems generally reduces the cost and time of development. Expert System shells come equipped with an inference mechanism supporting modus ponens (forward chaining–facts to goal), modus tollens (backward chaining–goal reduced to facts), or both. JESS [25] is one such widely used expert shell that supports both forward and backward chaining reasoning mechanisms. JESS also has the ability to manipulate and directly reason about using Java objects. Hence, expert systems shells such as JESS, CLIPS [12], and OPS5 [23] create a highly conducive environment for developing expert systems.

The rule-based approach has been used to solve a wide variety of hard-to-solve problems. MYCIN [85] was one of the first expert systems in the area of diagnosis [36] and troubleshooting. Other areas of knowledge such as planning and scheduling, financial decision making, knowledge publishing, design, and manufacturing have found use for expert systems to a certain extent.

A rules-based approach, because of its WHENEVER-THEN inference policy, provides a perfect expert system foundation for an educational environment. The explainable feature of expert systems is ideal for aiding learners in regulating their learning styles.

2.2 Mixed-Initiative Interactions

Mixed-Initiative interactions attempt to model an interaction strategy where conversants (user or systems) contribute appropriate information, when it is best suited, towards mutually negotiated goals [37]. At any one time, one conversant might have the initiative, controlling the interactions, while the others contribute to the interactions as required [1].

[12]http://www.ghg.net/clips/CLIPS.html

Mixed-Initiative interactions are driven by conversants' relative knowledge, preferences, and task toward common, partially shared, and individual goals.

Mixed-Initiative (MI) interaction [1], [75] comprises a new set of methodologies that propound the need for independent initiative-taking to assume control of the conversation within the context of discussion. Mixed-initiative systems exhibit various degrees of involvement [1], [43], [14] in regards to the initiatives taken by the user or the system. In any discourse, the initiative may be shared either between a learner and a system agent, or between two independent system agents. Both parties in question establish and maintain a common goal and context, and proceed with an interaction mechanism involving initiative-taking that optimizes their progress towards the goal.

The Mixed-Initiative technique has been found more effective in the case of computational linguistics and planning. Problems in these fields were solved more efficiently under mixed-initiative/declarative methods than random initiative/directive methods [30], [86]. Mixed-initiative systems provide a platform for identifying situations when system-control is more efficient than user-control, and vice versa. The initiative-changing mechanisms using negotiation improve the quality of user models. Currently, most educational systems do not have an explicit representation for initiatives and they do not tightly map learner interactions to educational theories. Mixed-initiative interactions enable explorations in cognitive domains modeling affect, negotiation, motivation, and so on. Importantly, the mixed-initiative approach enables systems to be explanation-aware.

The architecture for mixed-initiative systems is similar to that of knowledge representation systems. Some of the common architectures for mixed-initiative systems reported in the literature are a) finite-state machines, b) planners, and c) frames. The finite-state machines enable mixed-initiative interactions to navigate across predefined states of problem solving [70], [75]. Frames provide a conduit for mixed-initiative systems to accept input from the participants in a non-linear fashion. Planning systems, such as TRAINS [22], concentrate on the dialogue for mixed-initiative interactions regulated by the preconditions and the action associated with the planner. PASSAT [48] focuses on integrated user guidance with a planning algorithm. Ontology is emerging as a much more suitable architecture for the design, development, and deployment of mixed-initiative systems.

One of the key elements for successful mixed-initiation is the ability of the system to recognize opportunities for initiatives based on well-founded theoretical principles [84] [83]. Being mixed-initiative, these ontology-oriented educational systems enable the learner and the learning platform to contribute mutually beneficial reasons to reach a common goal. Having captured the knowledge of the domain and the knowledge associated with the learner interactions, educational systems are better suited to understand a learner's mental state with respect to the learner's learning domain in the context of a predefined educational theory.

In addition to mixed-initiative planning and mixed-initiative machine learning systems [88], there are only a handful of research efforts in the field of Intelligent Tutoring System that addressed mixed-initiative approaches. Most ITS researchers have employed dialogues instead of mixed-initiative interactions [54] as their channel of communication, as depicted in systems such as AutoTutor [26], ALTAS-ANDES [80], and PACO [79].

2.3 Educational Models

Theoretical educational research ranges from educational psychology, human behaviour, and social science, to technological critiques. Educational theories and models hold valuable clues to the development of the conceptual underpinnings of educational technology. The medium of learning has changed radically since the introduction of online interactions and technology-oriented teaching, learning, and research. However, there have not been many exemplars that **tightly** couple the theories and models of education into the world of educational technology. This literature review will focus mainly on the conceptual models of curricula and also will explore a seminal theory and models associated with self-regulation, an area in Educational Psychology.

2.3.1 Conceptual Curriculum Models

Existing education curriculum models are mostly conceptual and descriptive. Examples include the ICF-2000 [65], the WCIT [13] toolkit, ACM's Computing Curricula 2001 [45], Information Systems-Centric Curriculum [56], and the Organization and End-user Information System Curriculum Model [44]. These educational curriculum models come from different backgrounds and are developed for different purposes; hence, these models are in different formats with varied scope and very little overlap among them. The conceptual curriculum models are basically a simplified representation of the underlying data without

[13] http://www.nwcet.org/products/Toolkit/index.asp

the curricular processes that operate on the data. Some models are simply guidelines for the curriculum planning and design. However, most universities and other institutions of higher education do not adhere to a standard curriculum. Education researchers have attempted to standardize curricula resulting in models such as UNESCO/IFIP Information Curriculum Framework 2000 [64], which also presents a framework for curriculum development in higher learning.

The literature in Education identifies many conceptual curriculum models, curriculum frameworks, curriculum guidelines, and even lists of curriculum requirements. The methods of delivery of education based on these curricula and expected results differ, but their general aim to deliver education with respect to a common set of goals and objectives remains unique - aiming at sustainability of the quality of education. Many educational institutions plan their curricula with the help of models such as the student model, courseware model, learning policy model, financial model, and pedagogical model. Hence, the divergence in the existing conceptual curriculum models, the variance in the format and scope, and the influence of culture, scope, institutional size, budget, policy, pose a rather large-scale challenge in representing curricular models in a computational form.

2.3.2 Self-Regulated Learning

Learning is viewed as an activity that students do for themselves in a proactive way, rather than as a covert event that happens to them in reaction to teaching [95]. Such proactive students are called self-regulated learners and the theory that models and predicts such cognitive and meta-cognitive traits is called the theory of Self-Regulated Learning. There has been much research in educational psychology [95], [91], [69], [60] that conceptually articulate how people regulate their learning, particularly how they create structural knowledge and processes that underlie their abilities.

Recently, there has been a surge of interest among educators and computer scientists to inject and maximize the experience of self-regulation in e-learning, especially in intelligent learning environments. The theory of Self-Regulated Learning (SRL) concerns how learners develop learning skills and how they develop expertise in using learning skills effectively [92]. SRL comprises a set of strategies and tactics employed by learners to regulate their own learning processes. It arises from two key observations. First, learners' goals for learning take precedence over goals set by teachers, authors of curricula, and developers of learning objects. Second, learners are in charge of how they learn. They choose which study tactics and learning/problem-solving strategies to use as they strive to achieve their goals.

In the realm of SRL, a collection of specific features that characterize a process (or an artifact) is called a **schema**. Many schemas are formatted as a set of rules for carrying out tasks. For instance, experienced programmers have schemas that not only help them recognize strategic formations of program pieces; their schemas also include sophisticated tactics for handling the interrelations among program pieces. Moreover, an automated schema is what is typically known as a skill. A **tactic** is a particular part of a schema that is represented as a rule in IF-THEN form, sometimes called a condition-action rule. IF a set of conditions is the case, THEN a particular action is carried out. IF not, a learner's ongoing behaviour or qualities of interacting with the task proceed unchanged. A strategy is a design or a plan for approaching a high-level goal, such as mastering a new software system. A **strategy** coordinates a set of tactics. Each tactic is a potential tool to use in carrying out a strategy, but not all tactics that make up the strategy are necessarily enacted.

The self-regulated learning model, as described by Winne [91] consists of various tactics and strategies that students use to reach their goals. McCombs and Marzano [60] identify some of the means to recognize strategies and tactics employed in computer programming. Research shows that learners often set unsuitable goals, have a limited repertoire of learning skills, often do not use learning skills they have, and frequently need extensive help to manage learning and collaborative tasks [92].

Self-regulation involves the selective use of specific processes that must be personally adapted to each learning task. In the paper [95], Zimmerman presents the structure of self-regulatory processes in terms of three cyclical phases. The ***forethought phase*** refers to learners' mental processes and efforts before the actual learning; the ***performance phase*** refers to various learning processes that occur during the action of learning; and the ***self-reflection phase*** refers to the final processes that occur after each learning effort. Each of these phases and the processes involved are listed below:

Forethought Phase This first phase of the self-regulated learning process involves *task analysis* and *self-motivation*. During this task analysis the learner goes through the process of *goal setting* and *strategic planning*. Learners increase their academic success if they review the task analysis, perform strategic planning on the task, and set goals. The self-motivation is dependent on the learner's self-efficacy beliefs and intrinsic interest. Self-motivation is dependent on individuals and it is not a learning process.

Performance Phase This second phase in the cycle of SRL also involves two main components: *self-control* and *self-observation*. Self-control involves the execution of the strategies that were planned in the forethought phase. There are various strategies such as imagery, self-instruction, attention focusing, task strategies, and so on. Self-observation is the process of self-recording, self-experimenting, and self-monitoring. This phase is the process of learning and controlling the learning process through self-recording.

Self-Reflection Phase *Self-judgement* and *self-reaction* form this third phase. During this phase, learners self-evaluate to judge their accomplishment in comparison to a standard or to other performances. It is also in this phase where learners analyze their strategy in comparison to the other strategies they could have taken. Self-reaction is directly related to self-satisfaction. Depending on the scale, self-satisfaction can lead to decreased/increased motivation. Learners can become defensive with the efforts being put into the task.

Similarly, Winne and Hadwin's [92] model of self-regulated learning displays the relationship between the goal, the current state of the task, and how monitoring the task reflects on the goals. According to this second model, there are four different states in self-regulated learning - *Knowledge*, *goals*, *tactics and strategies*, and *product*. This model explores the close relation of how each one of the states affects the rest during the learning process. Each of these states are briefly summarized below:

Knowledge This is the cognitive state of the learner which consists of *knowledge and beliefs, domain knowledge, strategy knowledge, and multiple motivational beliefs*. Depending on the learner's knowledge, the learner will proceed to the next cognitive state of goal setting.

Goals This is the initial state of the learning process. The learner sets the goals based on their knowledge. The learner's knowledge affects the goal setting process and conversely goals setting affects the resultant learner knowledge.

Tactics and Strategies This is the state where the learner decides on the tactics and strategies to use to achieve the goal. It is during this cognitive state of mind that the learner uses the tactics to achieve their goal.

Product This is the last cognitive state of the learner where they accomplish the goals. This state results in *self-monitoring* and *self-evaluation*. The state of self-monitoring implies that learners are currently reflecting on their knowledge, their goal setting, and the tactics and strategies they used. It is the monitoring process that helps learners self-regulate their learning process. The processes of self-performance and self-evaluation result in external feedback that in turn affect learners' knowledge and goal setting.

Reviewing these two self-regulated learning models opened up a number of avenues to capture the relationships between learners' interactions and their cognitive states of mind in a reliable, theory-centric, and operational manner.

2.4 Summary

This literature review identifies and explores the key conceptual and technical areas that provide the foundation for the rest of the research. Some of the main aspects of this research is the representation of knowledge in the educational domain, the dissemination of knowledge to the learner, and tracing the theory of self-regulation based on the principles of collaboration between the system and the learner. The three core concepts that are reviewed here concentrate on *knowledge representation*, *mixed-initiative interactions*, and the *models of self-regulation*.

The review in the field of knowledge representation identifies numerous techniques for various types of representation. The online learning material, the learner interactions with the material, the learner subject knowledge, and the educational learning theories are some of the knowledge that needs representing and reasoned with. The research aims to disseminate the acquired knowledge in a more natural method of communication with the goal of helping learners regulate their process of learning. Description logic provides a solid basis for this research since it is highly structural, formal, and expressive. DL reasoners that are readily available, such as RACER[14], can be utilized for the purposes of inferencing. Using an ontological approach allows for the knowledge to be represented in the DL structure, enables the expansion of the domain application, and importantly makes the knowledge more shareable.

[14]recently it has been commercialized

As identified in the review, the most common ontological representation language, OWL, also supports inferencing through Description Logic. The combination of DL structured ontology representation provides a required degree of expressivity and yet a formally structured backbone for this research. In consideration of the recent trend in online material, learning objects that use ontology based meta-tagging cater to a natural mechanism for interoperability.

Knowledge dissemination is the reason why one should represent knowledge. This research advocates the use of mixed-initiative interaction techniques in order to bring a sense of purpose and regulation to the human-machine communication dialogues. The foundation of mixed-initiative interaction research reveals the need for conversants being equally responsible in achieving the goal. This sets a perfect platform for our knowledge dissemination techniques. Designing the system with mixed-initiative interaction addresses the issue of the system recognizing the opportunities for system-initiation based on a) the inferred knowledge, b) the scaffolded rules from the theories of education, and c) the collaborative efforts of the learner striving to regulate his/her learning habits in light of the non-intrusive help from the system.

Study of pedagogical theories such as self-regulated learning provides a solid background for this research. Once represented, the theoretical models can track the meta-cognitive processes of the learner based on the learner interactions with the system. Furthermore, the theoretical models can provide a framework that derives explanation for specific patterns of learner interactions.

Chapter 3

MI-EDNA: Design and Implementation

3.1 Overview

This section discusses the core design of MI-EDNA[1], the system that has been built to investigate the design principles of Mixed-Initiative interactions (MII) [1], [88], [31]. Successful mixed-initiative systems employ mechanisms that explicitly recognize opportunities for initiatives between the system and the users. In MI-EDNA, the foundation for the initiative comes from the theory of self-regulated learning (SRL) [95] and the principles of MII. The design captures the technological needs to represent and reason with the SRL-oriented theoretical underpinnings of MII. This mapping between the technological means and the underlying theory caters to machine interpretability, ontological interoperability, knowledge maintainability, and human-computer interfacing capabilities of MI-EDNA.

MI-EDNA is a mixed-initiative learning environment that employs ontological representations to capture the semantics of the models of self-regulated learning. Further, MI-EDNA maps specific learner interactions onto SRL model variables. By analyzing the mapped model, one can reflect on learners' regulation skills with respect to the theories of SRL and can be in a position to explain the correlation, if any, that exists between performances and

[1] 'MI' in MI-EDNA stands for Mixed-Initiative; 'EDNA' stands for the mixed-initiative movie character, Edna Mode, who remains the inspiration for this work!

interactions observed in specific learner activities. This section highlights an overall architecture, the ontological underpinnings, the process of capturing and populating the ontology, recognizing the SRL tactics and strategies in an ontological formalism, and how ontological representation enables inference mechanisms to infer knowledge on learners' cognitive models.

3.2 System Architecture

The architecture of MI-EDNA is geared towards addressing the goals of enabling both the system and the learner to be able to explain why a particular interaction has been initiated and how such opportunities have been recognized based on the principles of SRL. It promotes modularity of system development and adaptability to the needs of the learners since the rules and facts can be fed into the system in OWL format and reasoned with, *at run time*. The system architecture comprises technical architecture and functional architecture.

3.2.1 Technical Architecture

The technical design of the system consists of four main components: the underlying ontology (CILT.owl, TTS.owl, CILT-Instantiated.owl), the ontology instantiator (OntoParser), the inference engine (Gessie), and the interface (gStudy, Query tools). The technical architecture of MI-EDNA is presented in Figure 3.1.

The ontology acts as the connector or as a blackboard[2] between different modules of the architecture. The ontology acts as an area of information exchange for the other three components. Further, the ontology also coordinates the actions arising from the other three components with the goal of providing value-added, theory-centric feedback to the learners.

The data instantiation into the ontological structure can be fully- or semi-automated; that is, the instances can either populate the ontology without any human intervention or with minimal human manipulation. Presently, MI-EDNA's fully-automated instantiators are used to transfer data from learner interactions into the ontology. The learner interaction data, captured from within an online study environment [33] as part of an experiment in a real classroom setting, is stored in XML format. An XML parser is used to browse the XMLized data and create the corresponding ontology instances in OWL format. The XML

[2] http://en.wikipedia.org/wiki/Blackboard(computing)

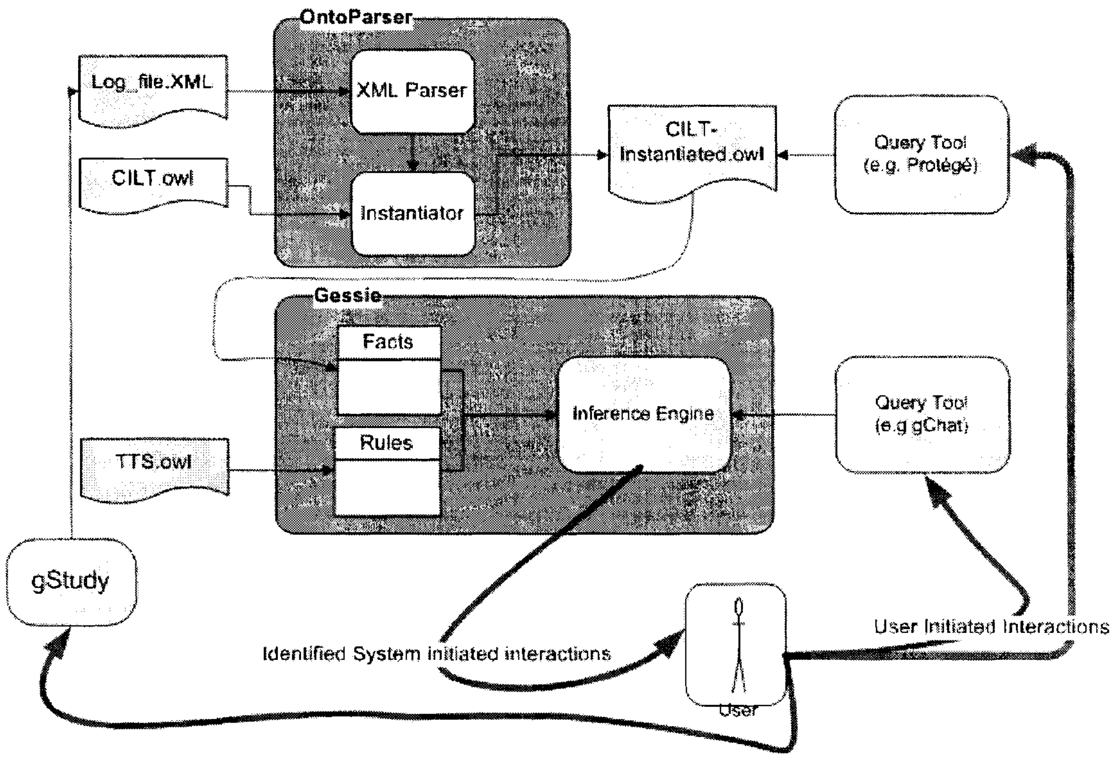

Figure 3.1: Technical Architecture

parser instantiates the concepts and establishes the relations between instances that have just been created in the ontology based on the constraints and the restrictions predefined in the ontology.

MI-EDNA uses two types of inference engines: one based on Description Logic and the other on Production Rules. The inference engines provide a gateway for the system to reason with the encoded knowledge. The encoded knowledge contains the interaction data of the learners and the self-regulated learning patterns. In addition, the encoded knowledge also contains rules that interpret patterns in learner interactions.

The learners, using the query interface, extract information about their learning styles and their learning patterns to help them self-regulate. Aside from the learners, educators and researchers can also query MI-EDNA for summary information on how learners attempt to self-regulate. In addition to the query interfaces, MI-EDNA also provides channels of communication for a variety of users, learners, educators, and researchers to investigate the ontology.

3.2.2 Functional Architecture

The technical architecture of MI-EDNA displays the system components and their relationships with each other. On the other hand, the functional architecture of the system describes the flow of information across the components within the system. This functional architecture is depicted in Figure 3.2.

MI-EDNA's design is centered around the notions of SRL, MII, and Ontology. The functional architecture aims at the utility and sustainability of these notions within MI-EDNA.

Learners evolve their learning strategies as they progress through their learning process. MI-EDNA is designed to adapt to the observable changes in learners' knowledge and the strategies they employ over a period. Self-regulation transforms mental abilities to academic skills, which involves selective use of specific processes that must be personally adapted to each learning task. Such cognitive transformations are explicitly captured in the ontology and are shared with other key modules in MI-EDNA for adaptation.

SRL processes include: a) setting proximal goals; b) adapting strategies to attain goals; c) monitoring performance for signs of progress; d) restructuring contexts to make them compatible with goals; e) managing time; f) self-evaluating ones methods; g) attributing

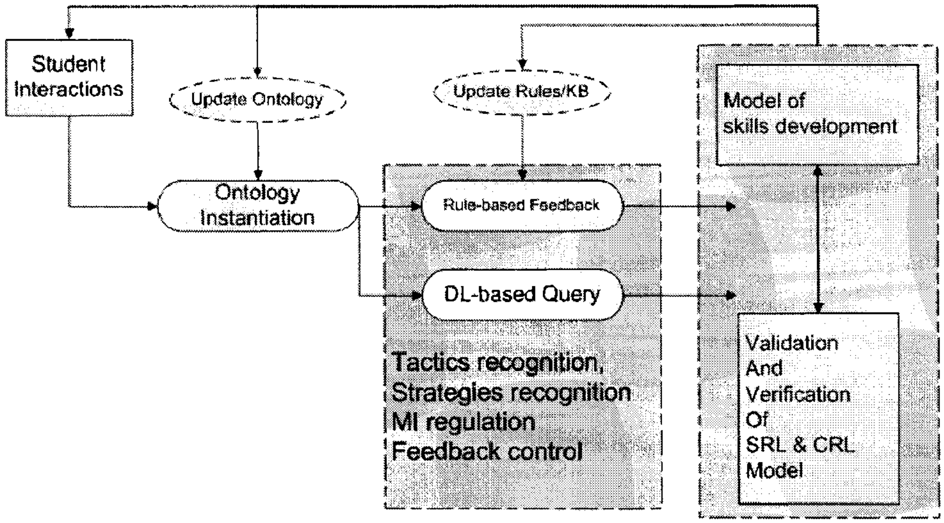

Figure 3.2: Functional Architecture

causation to results; and h) adapting future methods[3]. SRL students focus on how they activate, alter, and sustain specific learning practices in social contexts as well as solitary contexts, and this process is functionally captured within MI-EDNA's data flow. For instance, in helping learners to adapt strategies to attain goals, MI-EDNA provides interfaces for learners to set their goals, to monitor their progress, to identify tactics and strategies they use in attaining specific goals, to compare their strategies with the strategies employed with their peers, to explore strategies advocated by the instructor, and to compare their performances after they adopted a new set of strategies. MI-EDNA tracks the flow of information and control throughout the learning process that a learner engages in, and identifies opportunities for non-intrusive system intervention. Similarly, the flow of each one of the SRL processes identified earlier can be traced in MI-EDNA's functional architecture with respect to specific patterns that are observed in learner interactions.

The flow in the functional architecture starts with learner interactions. As explained in Section 3.2.1, these interactions are fed into the ontology as instances.

Certain patterns of instances invoke specific rules in the Production Rules inference engine. These rules help recognize opportunities for Mixed-Initiative interaction and also

[3]http://www.learningkit.sfu.ca/

recognize tactics/strategies employed by the learners in order to provide feedback to learners.

Further, the rules and the DL-based inference engine are designed to capture the *skills development processes* corresponding to each learner. The elements of the *skills development* model includes meta-cognitive information pertaining to meta-memory, meta-comprehension, schema-training, and self-regulation[4]. Meta-memory tracks the process (e.g., recall or prompted) learners used to select tactics and strategies. Meta-comprehension traces how learners estimate how much they know or don't know in a subject, and how much they know about the remedial actions. Schema-training tracks whether the learners recognize that they are developing or not developing new cognitive structures. Finally, self-regulation attempts to gauge the effectiveness of learners' remedial actions.

Finally, the *skills development* model validates and updates the ontology and the rules based on external feedback on the validation of what the model represents. The validation is normally obtained from the learner or the instructor.

3.2.3 Implementation Domain

The architecture of MI-EDNA is implemented as part of the Learning Kit Project[5]. Learning Kit aims to develop a generic study tool named gStudy[6]. The project investigates how learners cognitively strategize ways to classify, index, annotate, analyze, organize, evaluate, and cross-reference information as part of their study habits. The project is built on and extends the theory of self-regulation. The main research aspects of Learning Kit are how learners develop learning skills and how they develop expertise in using learning skills effectively. The theory arises from two key observations: learners goals for learning take precedence, and learners are in charge of how they learn.

Inducing self-regulated learning habits through coaching is the current philosophy behind gStudy. The system is designed to monitor the current state of the learner through activity patterns, to compare patterns with ideal self-regulated actions to encourage learners to self-regulate, and to elaborate the learners model based on interactions. The interactions between the system and the learner are based on the system making meta-cognitive suggestions during dialogues with the learner, responding when the learner adapts their behavior,

[4]Judy Atkins, University Of Saskatchewan,
http://www.usask.ca/education/coursework/802papers/Adkins/ADKINS.PDF

[5]http://www.learningkit.sfu.ca/index.html

[6]gStudy - The online learning tool developed for LearningKit project.

and having the ability to handle questions related to help, tests, assignments, and planning. Figure 3.3 on the gStudy architecture design layers displays the various factors addressed in the design of the Learning Kit project.

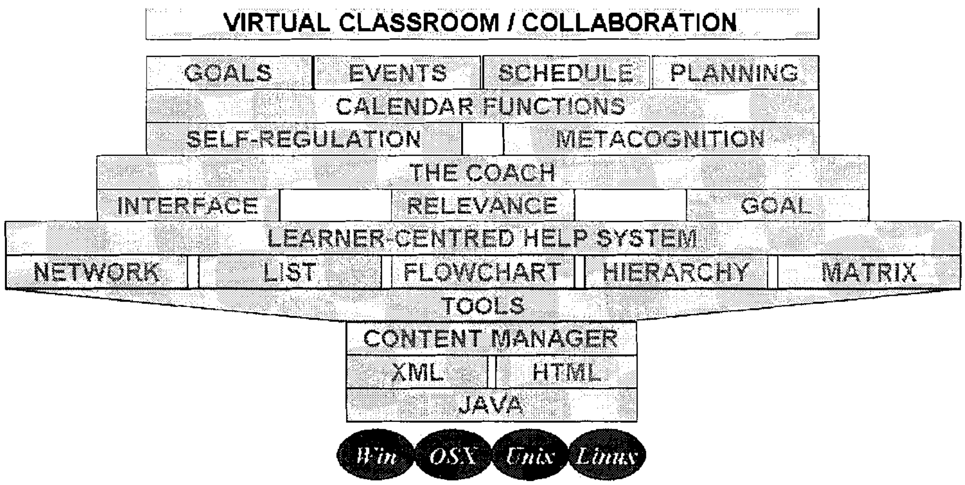

Figure 3.3: gStudy Architecture Layers

gStudy allows the learner interactions to be captured and analyzed to recognize tactics and strategies that students use to reach their goals during the learning process as described in [91]. [33] outline some of the generic strategies and tactics students used in the domain of Reading. Learner interactions with gStudy are captured in a log file. The interactions that we currently target in gStudy include browsing, highlighting, compiling code, text chatting, indexing, concept mapping, note taking, reviewing, and collaborating. Using the log as the source, one can record various tactics and strategies when students are engaged in performing tasks related to reading, composition, or problem-solving (e.g., Java Programming).

MI-EDNA provides the second channel of ontology-guided coaching to students engaged in gStudy. By interconnecting the ontologies with gStudy's tools and the patterns of self-regulated learning, MI-EDNA provides contextualized support for learners "on the fly" as they study though gStudy. This research collects data on how such explainable and theory-oriented prompting and feedback promote significant, transferable, and enduring changes to learner study skills and problem-solving abilities.

3.3 System Components

This section elaborates on the components of the system architecture and the foundational aspects of the design decisions.

3.3.1 Ontological Underpinnings

The ontological representation of the domain knowledge and the interaction knowledge allows for a formal representation of the concepts and the relationships between them in Description Logic as formulated in OWL-DL [41]. OWL-DL allows one to represent concepts and relations with restrictions. This section describes an OWL-DL based representation of the Content-Interaction-Learner-Time (CILT) and the Teaching Tactics and Strategies (TTS) ontologies in detail.

Content-Interaction-Learner-Time Ontology

Content, learner, interactions, and time are computational entities that are fundamental to representing the learner interactions within the gStudy tool. The objective in building this application ontology is twofold: first, it provides a sharable and interoperable framework; second, it provides flexibility in terms of plug-and-play components. The four major components (Content, Learner, Interaction, and Time) of the ontology can be used interchangeably across different applications/domains. As identified in Chapter 2, domain ontologies are a collection of interconnected ontologies which define the details of general concepts and their features in each sub-domain. The CILT ontology comprises four specific domain ontologies listed below.

Content This ontology represents the structure of the document that is represented in the gStudy tool. The content consists of two main concepts: *documentFragments* and *documentComment*.

The fragments consist of various elements of the documents such as *paragraphs, images, glossary, quiz, and so on.* Fragments by themselves do not have any association with any domain. The domain ontology can be imported as part of the Content ontology. With the use of object type properties, the fragments can establish associations with domain specific topics. Each of the fragments as in Figure 3.4[7] is associated with one or more domain topics and their relations to other document elements in the content.

The comments are an interesting variation, as they represent document elements that are created/added by the learner. During the process of learning, the learner creates notes and concepts in relation to the fragments. The comments as shown in Figure 3.5 explicitly represent the topics they relate to and also the relationships between the comment and the fragment.

Figure 3.6 illustrates the ontological structure of the content ontology. A complete listing of the content and the corresponding ontology are presented in Appendix A. The shaded classes *DocumentElement* and *Note* are defined classes [78] in this ontology. The content ontology predominantly uses 'is-a' relation to connect document elements. In addition, document elements can also have relations defined through $owl : objectTypeProperties$. For example, an instructor meta-tagging the document fragment as *important* can be represented as a relation between the instructor and the document fragment.

For instance, the content ontology has a relation named '*hasElement*' that has been established between concepts '*documentFragment*' and '*Image*'. This relation is tagged with '*owl:TransitiveProperty*', and the relation '*isPartOf*' has been established as the '*inverse property*' of '*hasElement*'. Instantiating the ontology, the '$chapter1_section1$' has a relation '*hasElement*' of type '*Image*', which in turn is a part of the '*Media*'. As a result of this, DL can be used to conclude that '$chapter1_section1$' has an element of type '*Media*'.

[7]Screenshots from Protege software. Used with permission based on the open-source Mozilla Public License.
http://protege.stanford.edu/download/download.html
http://www.mozilla.org/MPL/

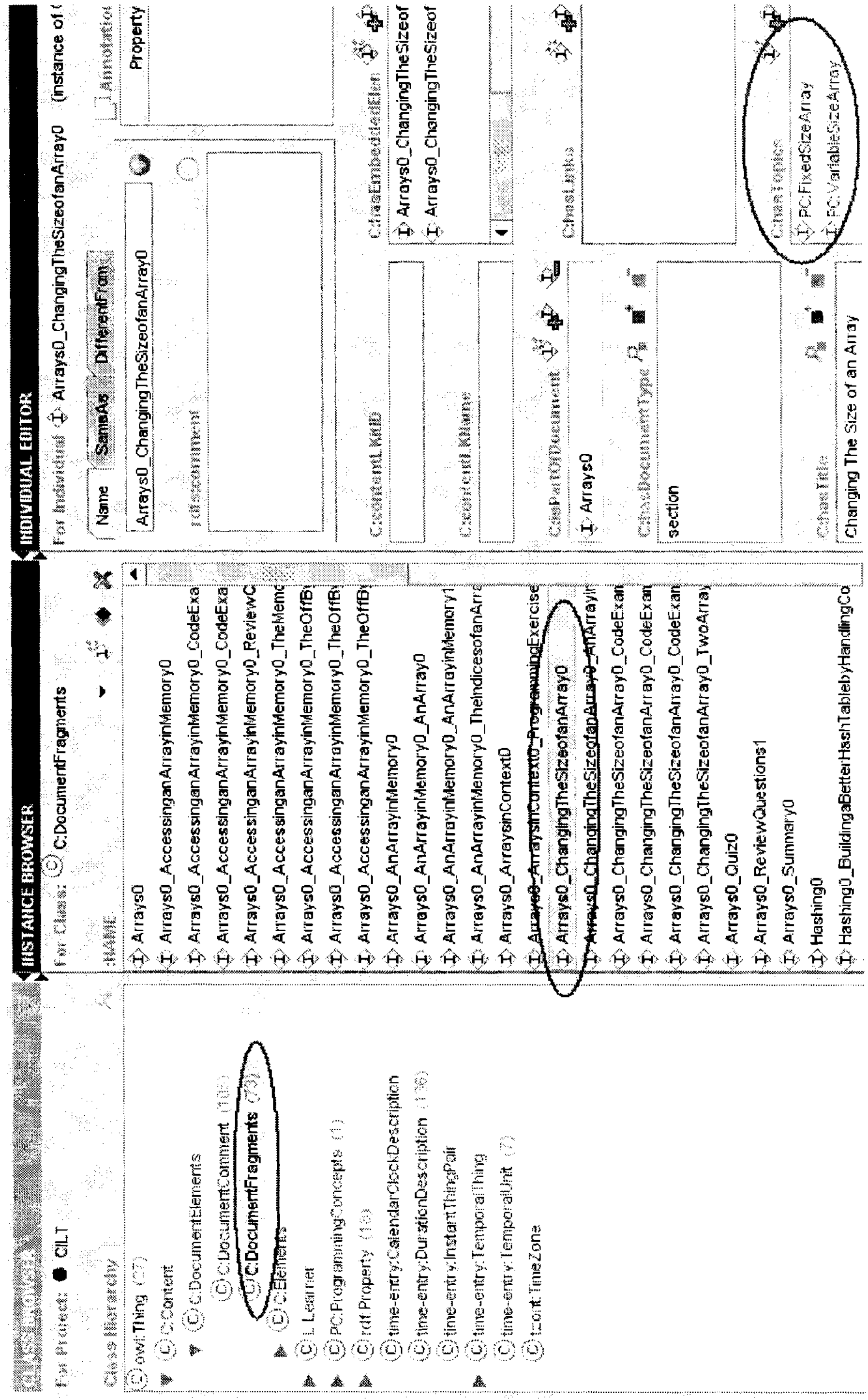

Figure 3.4: Document Fragment mapped onto one or more domain specific topics

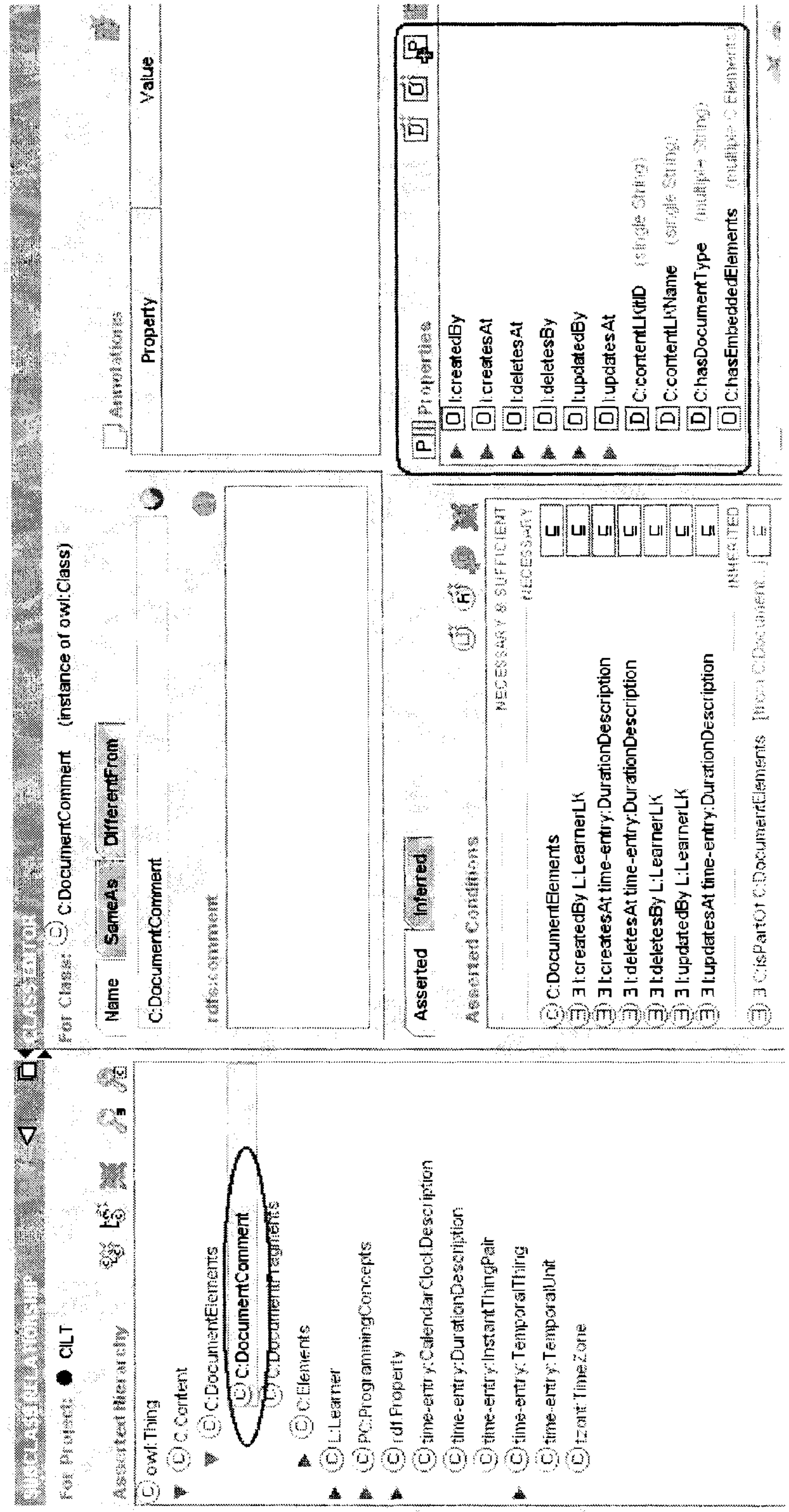

Figure 3.5: Document Comment mapped onto Document Element

Figure 3.6: Content Ontology

Interaction This ontology is mainly about the learner interactions, specifically focusing on learner interactions that are observed within gStudy. This ontology does not consist of concepts but rather consists of interactions represented as *owl* : *objectTypeProperty* with the goal of being able to use these object type properties as relations between concepts in the overall ontology. Keeping the interaction ontology with only the object property makes it more modular as the developers and researchers can add the interactions to only one ontology and still be able to use it across the components of the system for various purposes.

Name	P...	Range	Domain	Inverse	Other Characteristics
createdBy			owl:Thing	creates	
creates			owl:Thing	createdBy	
createsAt			owl:Thing		
debugs			owl:Thing		
debugsAt					Super properties: {debugs}
debugsCode					Super properties: {debugs}
deletes			owl:Thing	deletesBy	
deletesAt			owl:Thing		
deletesBy			owl:Thing	deletes	
deletesTarget			owl:Thing		
hasTargetObject		string	owl:Thing		
highlightedBy			owl:Thing	highlights	
highlights			owl:Thing	highlightedBy	
highlightsAt			owl:Thing		
highlightsTarget			owl:Thing		
implementedBy			owl:Thing	implements	InverseFunctional
implements			owl:Thing	implementedBy	
linkedAt			owl:Thing		
linkedBy			owl:Thing		Functional, Transitive
links			owl:Thing	links	Functional, InverseFunctional, Symmetric
logIn					Super properties: {systemEventType}
logOut					Super properties: {systemEventType}

Figure 3.7: Excerpts from the Interaction Ontology

The interactions are centered around learning tasks in gStudy such as *creates - createdBy, highlights - highlightedBy, links - linkedBy, takesQuiz - takenQuizBy, browses - browsesBy,* and so on. Learning tasks in the domain of problem-solving (e.g., Java Programming) includes *implements - implementedBy, compiled - compiledBy, debugs - debuggedBy,* and so on. This interaction ontology on its own is not very useful. It acts as a glue between the collection of ontologies - Content, Learner, and Time. Figure 3.7 displays some of the *object properties* in the interaction ontology.

Learner This ontology represents information related to the learner. Considering there are various representation standards for user information (e.g., LIP[8], PAPI[9]), this ontology only provides a skeleton into which other structures (based on standards such as PAPI and LIP) can be imported. The learner ontology only represents the minimal amount of data that is essential for representing the gStudy learner.

The gStudy learner information is restricted to variables that are associated with the notion of self-regulation. Some of the key information that the Learner ontology represents includes *goals of the learner, domain knowledge of the learner, self reliance*

[8]http://www.imsglobal.org/
[9]http://edutool.com/papi/

of the learner, and the social aspects of the learner. Specifically, the learner ontology infers information related to the aforementioned variables using data obtained from learner interactions with gStudy. The learner interactions with the content are related through the use of objectTypeProperty from the interaction ontology. Figure 3.8 shows the properties in the learner ontology as obtained from the gStudy tool.

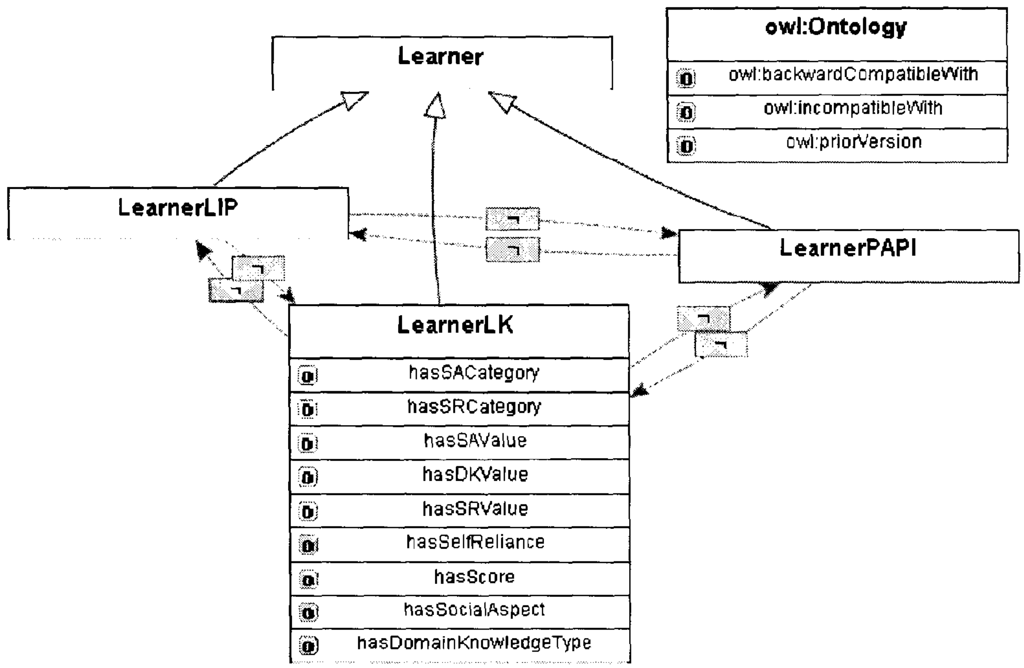

Figure 3.8: Learner attributes

Time The time ontology is needed mainly for the tracking of learner interactions with respect to time. There are many published time ontologies that can be imported for use in MI-EDNA. Currently, MI-EDNA uses the DAML-Time[10] ontology. Though the time ontology is too extensive for the purposes of our research, the same has been imported with the intention of instantiating only selected concepts such as *time-entry:DurationDescription.*

All these four ontologies are imported into a single, integrated ontology, named the CILT ontology. Concepts across these ontologies are connected with each other using the

[10]http://www.isi.edu/ pan/damltime/time-entry.owl

objecttype properties from the interaction ontology. The CILT ontology is represented in OWL-DL format. CILT ontology is enhanced by the restrictions on concepts and relations based DL constructs. For example, for every note taken by a learner, the note has to be linked to at least one *documentFragment*. Thus, the use of *Cardinality* synopsis featured in OWL-DL enriches the representation of CILT. It captures the essence of the learners' interactions in the application at any given time frame.

Teaching Tactics and Strategies Ontology

Educational Psychologists have identified and advocated a number of models of learning corresponding to self-regulated learning (SRL). One such model promotes that SRL consists of the phases of forethought, performance, and self-reflection [95]. The forethought phrase corresponds with preplanning what a learner would undergo prior to engaging in the learning activity. The self-reflection phase corresponds with the post-learning process. The performance phase reflects the processes that occur during learning. Learners engage themselves in learning tactics to achieve self-control and self-reflection during this phase. Another model promotes that SRL consists of four different phases knowledge, goal, tactic and strategies and product [92]. However, this model emphasizes more on the each different phases being driven by the goal and motivation of the learner. The goal and motivation of the learner determines the the tactics and strategies the learner adopts to achieve the product. The prior knowledge of the learner determines the goal set by the learner. Learners practice self-control and self-reflection strategies to incorporate specific cognitive tactics to eventually achieve their goal. Since the learner knowledge changes with their progress, the tactics and strategies they adopt also changes. Winne and Hadwin's SRL model, emphasis on the cognitive tactics and learning strategies chosen by the learners to be based on the learners knowledge and learners goals.

Student learning activities, their interactions, and typical SRL tactics and strategies are represented in the TTS ontology. The TTS ontology formally captures SRL-specific teaching tactics and strategies in addition to other human-oriented teaching tactics and strategies. An excerpt of the TTS ontology is presented in 3.9.

The TTS ontology can represent different SRL models and their components, independent of each other. OWL-DL axiom synopsis such as *disjointWith* and *oneOf*, and OWL-DL boolean combinations of class expressions synopsis such as *unionOf* and *intersectionOf*, form

```
<owl:ObjectProperty rdf:about="#hasPlannedStrategies">
  <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isCollectionOf"/>
<owl:ObjectProperty rdf:about="#teaches">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#IndependdentSeatWorkTT"/>
        <owl:Class rdf:about="#DoIReallyKnowItTS"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <owl:inverseOf rdf:resource="#isTaughtBy"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#doesTaskAnalysis">
  <rdfs:domain rdf:resource="#SRLForeThoughtPhase"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasLearnGoalOriented">
  <rdfs:range rdf:resource=
          "http://www.w3.org/2001/XMLSchema#boolean"/>
  <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasGoal">
  <rdfs:range rdf:resource=
              "http://www.w3.org/2001/XMLSchema#boolean"/>
  <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
  <rdf:type rdf:resource=
              "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasStrategicPlan">
  <rdfs:range rdf:resource=
             "http://www.w3.org/2001/XMLSchema#boolean"/>
  <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
  <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

Figure 3.9: Excerpts from the Teaching Tactics and Strategies Ontology (OWL)

the basis of the expressive power in the TTS ontology. Presently, TTS represents a combination of two SRL models: Zimmerman's three-phase model [95] and Winne and Hadwin's four-phase model [92].

Different types of constraints have been built corresponding to '*owl:DataTypeProperty*' and '*owl : ObjectProperty*' with respect to the phases/states of the SRL models. For example, the *SelfRegulatedLearningET* in Zimmerman's SRL model consists of '*DataTypeProperties*' such as '*hasGoal*', '*hasExpectedOutcome*', '*hasSelfEfficiency*', '*hasStrategicPlan*', and so on. The concept constructs for *SelfRegulatedLearningET* define the uniqueness of the three phases of the model. As an extension of the concept, the concept constructs further define the concept *OptimalSelfRegulatedLearningET* by imposing restrictions on these DataTypeProperties. In comparison to Winne and Hadwin's SRL model, the '*DataTypeProperties*' such as '*hasGoal*' and '*DataTypeProperties*' such as '*hasLearnerKnowledge*', affects the '*hasStrategicPlan*' the learner creates before proceeding with the learning.

Learner transitions across phases are enabled based on values of certain dataTypeProperties. For example, figure 3.10 shows that in order to reach optimality, the SRLForeThoughtPhase enforces the restriction on dataTypeProperties such as *hasGoal, hasPlannedStrategy* to be set to True. This indicates that the learner must define his/her goal prior to starting the learning process.

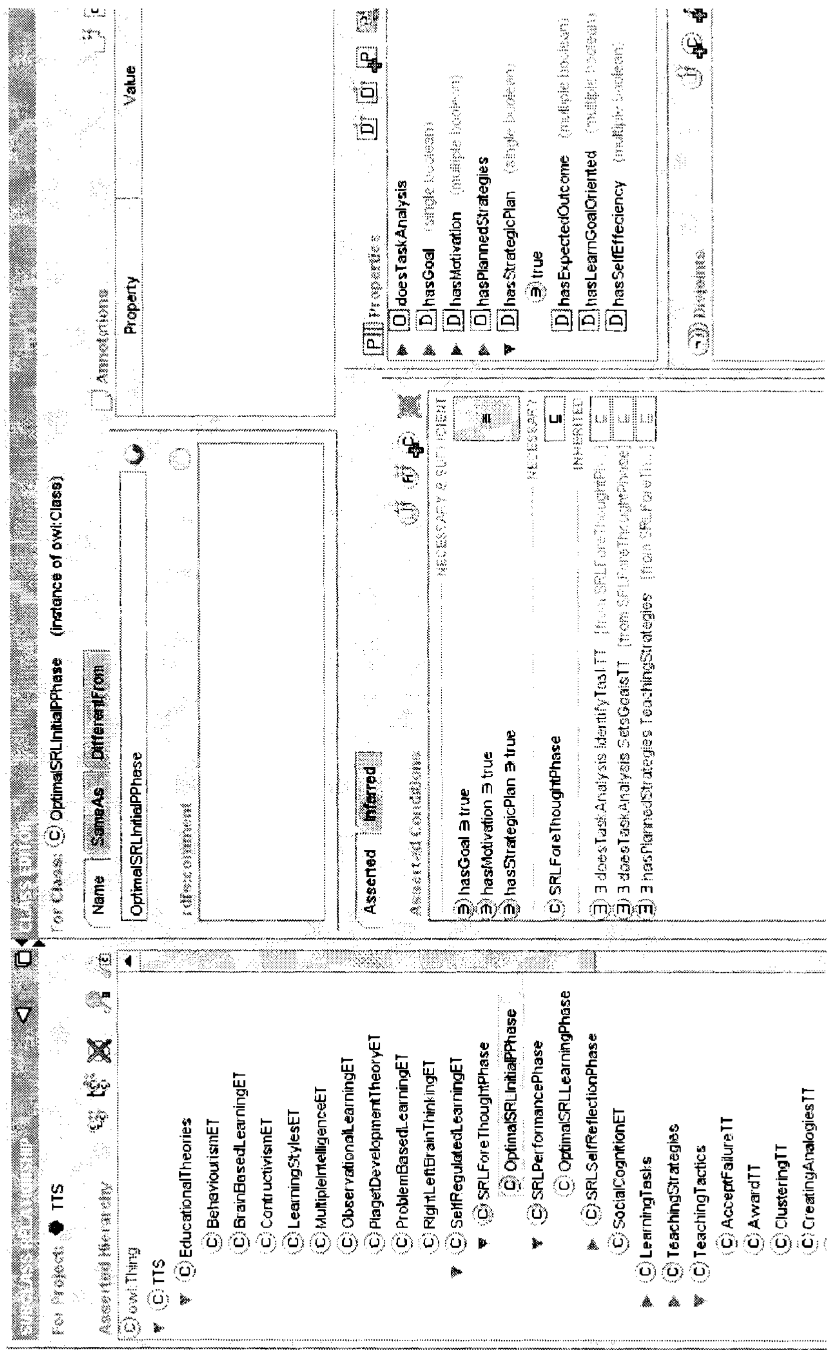

Figure 3.10: Optimal SRL Forethought Phase

3.3.2 Ontology Instantiation

Ontology of any domain or application serves as the blueprint for knowledge sharing and representation in that domain or application. The basis of the framework that underlies ontology supports the key capabilities, namely representing, acquiring, sharing, and utilizing knowledge. However, knowledge representation and sharing capabilities need to be transparently interwoven at the user interaction level and the ontological representation, recognition, and reasoning at the meta level. Automated instantiation of the ontology still remains one of the crucial aspects of ontological knowledge engineering. We contend that the strength of ontology-oriented systems rely heavily on the automatic or semi-automatic instantiation of knowledge in the ontology.

Manual knowledge instantiation is tedious, cumbersome, and error prone. Ideally, ontologies should be instantiated in an automated fashion. However, not many systems have been designed to fully automate instantiation of the underlying ontologies. As described below, this research utilizes an automatic and semi-automatic instantiation of ontologies.

In MI-EDNA, the content is meta-tagged and instantiated in the ontology in a semi-automatic manner. For instance, the online contents for Java Programming is developed and tagged using DocBook[11] platform. XSLT[12] style sheets are used to automate the process of instantiating the content ontology.

The learner interactions, however, are fully automated in MI-EDNA. While learner interactions are being logged in an XML file, in parallel, the real-time interaction data can also be fed into the ontology instantiator. The ontology instantiator creates the corresponding OWL instances in an OWL file using the raw XML data and the CILT ontology (in OWL format). Essentially, the instantiation mechanism automatically maps the log of trace data of learner interactions captured within gStudy onto an ontological formalism.

3.3.3 Reasoning and Inference

Computational reasoning and inferencing refers to computer-based emulation of the human capability to arrive at a conclusion by reasoning [81]. The reasoning capacity is dependent on the formalism in which the information is stored in the system. There are various schemas and methods that are available to retrieve information. Information can be simply

[11] www.docbook.org

[12] http://www.w3.org/TR/xslt

extracted based on algorithms, as exemplified in procedural processing, assuming that the information is stored as data structures. If the information is represented in frames, then various schemas and scripts can be used to retrieve knowledge based on associations and constraints explicitly established across slots. Network representation of information uses activity propagation methods to extract knowledge. Backward and forward chaining methods can be used to extract knowledge in rules-based systems. Similarly, there are other representation schemas that utilize formalism-specific reasoning mechanisms to extract knowledge. Hence, the reasoning mechanism one would consider is completely dependent on the type of formalism of the information.

As explained in the previous section, MI-EDNA uses ontology as the knowledge representation formalism. Knowledge represented in an ontological form can be inferred though a number of reasoning mechanisms. Ontologies explicitly represent the meaning contained within the stored information. The meaning, otherwise known as the semantics of the represented information, is explicitly captured in terms of the description of concepts and their interrelationships. Given this structural representation, one can employ Description Logic based reasoning mechanisms assuming that the structure is constrained by a class of Description Logic formalisms.

MI-EDNA uses a DL-based formalism and a rule-based formalism to represent information and infer knowledge. Multiple forms of representation pave the way for multiple knowledge processing mechanisms. The rule-based representation increases the scope of the inference by providing a means to separate units of knowledge and by providing capability to add or modify the knowledge independent of the representation. The rule-based representation also brings in a naturalness to knowledge processing - a more human-oriented method to view, query, and infer knowledge. Thus, the formal representation of information in the CILT ontology and the SRL principles in the TTS ontology enable knowledge to be processed with logical reasoning mechanisms as well as with rule-based reasoning mechanisms.

Reasoning with Description Logic

The ontological representation in MI-EDNA uses the OWL-DL sub-language. OWL-DL ontology consists of two parts, an intensional part and an extensional part [6]. The intensional part is known as TBox and contains knowledge about concepts and relationships between concepts. The extensional part is known as ABox and contains knowledge about instances and how they relate to the concepts and relationships. The instantiated CILT and the TTS ontologies of MI-EDNA contain both TBox and ABox knowledge. Thus, OWL-DL representation of the CILT and the TTS ontologies permits the use of reasoners based on Description Logic (DL). DL reasoners[13] help build and maintain shareable ontologies by revealing inconsistencies, hidden dependencies, redundancies, and misclassifications [77]. Some of the basic DL reasoning techniques are: class consistency, concept subsumption, instance checking, and concept satisfiability [6], [50].

CILT ontology represented in OWL-DL formalism uses a restricted set of first order logic called DL-constructors. OWL-DL provides a set of axioms to infer knowledge from the represented ontology. Some of the examples of the DL-based reasoning is shown in Table 3.1.

This table exemplifies the capabilities of DL-based inferences. The DL based ontological representation permits MI-EDNA to use logic based reasoners. Rule-based reasoning extends the expressivity of OWL at the expense of the decidability of query answering operations [42]. The next subsection presents the rules-based approach to reasoning.

Reasoning with Production Rules

MI-EDNA extends the OWL-DL representation of the knowledge to rule-based representation that extends Mixed-Initiative interaction (MII) features of the system. Figure 3.11 shows the flow of information from the ontological representation to the rule-based inference engine.

The system uses the JESS inference engine and the corresponding production rules to cater to the requirements of MII. The OWL metamodel and the instantiated CILT ontology are translated into *facts* for consumption by the JESS engine. The translation is accomplished by means of XSLT stylesheets[14].

JESS rules play a major role in the analysis and dissemination of SRL specific knowledge. The TTS ontology defines the SRL principles. The rules related to the tactics and strategies are formulated in Semantic Web Rule Language (SWRL) [42], which in turn are transformed into JESS rules with the help of XSLT stylesheets[15]. This production rule mechanism

[13] e.g. Racer - http://www.racer-systems.com/

[14] http://mycampus.sadehlab.cs.cmu.edu/public-pages/OWLEngine.html

[15] www.inf.fu-berlin.de/inst/ag-nbi/research/owltrans/

Table 3.1: DL-based reasoning rules

DL Axioms	**Explicit Representation** **Inferred Knowledge**
subClassof (?a rdfs:subClassOf ?b) ^ (?b rdfs:subClassOf ?c) => (?a rdfs:subClassOf ?c)	<owl:Class rdf:ID="DocumentComment"> <rdfs:subClassOf> <owl:Class rdf:ID="DocumentElements"/> </rdfs:subClassOf> </owl:Class> <owl:Class rdf:about="#DocumentElements"> <rdfs:subClassOf> <rdfs:subClassOf rdf:resource="#Content"/> </rdfs:subClassOf> </owl:Class>
	<owl:Class rdf:ID="DocumentComment"> <rdfs:subClassOf> <rdfs:subClassOf rdf:resource="#Content"/> </rdfs:subClassOf> </owl:Class>
TransitiveProperty (?TP rdf:type owl:TransitiveProperty) ^ (?a ?TP ?b) ^ (?b ?TP ?c) => (?a ?TP ?c)	<owl:TransitiveProperty rdf:ID="hasReference"> <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/> </owl:TransitiveProperty> <owl:Class> <rdf:Description rdf:about="...#Algorithms"> <owl:onProperty rdf:resource="#hasReference"/> <rdf:Description rdf:about="...#DesignPatterns"/> </rdf:Description> </owl:Class> <rdf:Description rdf:about="...#DesignPatterns"> <owl:onProperty rdf:resource="#hasReference"/> <owl:someValuesFrom rdf:resource="...#Efficiency"/> </rdf:Description> </owl:Class>
	<owl:Class> <rdf:Description rdf:about="...#Algorithms"> <owl:onProperty rdf:resource="#hasReference"/> <owl:someValuesFrom rdf:resource="...#Efficiency"/> </rdf:Description> </owl:Class>

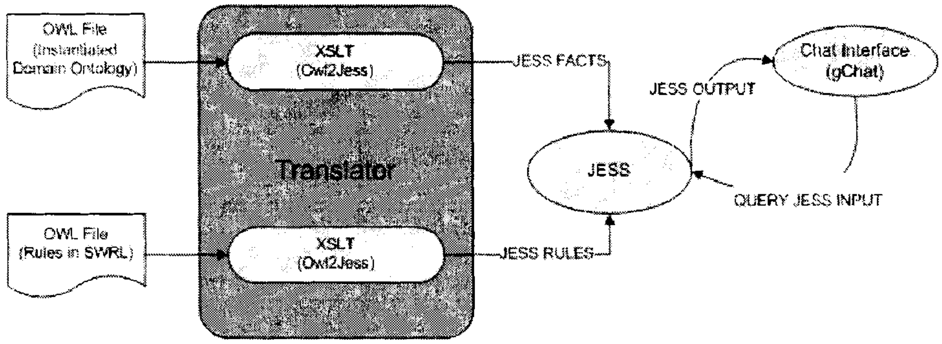

Figure 3.11: Reasoning with Production rules in MI-EDNA

permits the system to detect the SRL principles that the learners are engaged in and provides the logical point of information dissemination to the learners.

The JESS engine consists of data models defined as templates, the user-defined/pre-defined rules, the working memory consisting of the facts, functions, and global variables. The data model is defined in the Jess template as a triple, (predicate - subject - object), as shown below. Building the template as triplets instead of the complete OWL tree makes building the rules simpler and easier.

```
(deftemplate triple "Template representing a triple"
(slot predicate (default ""))
(slot subject   (default ""))
(slot object    (default "")))
```

The working memory has all the facts in the form of triplets as defined in the template. OWL ontology is transformed via XSLT into the triplets. An example of the triplets is shown here that explicitly represents the subject $O1418C11documentComment$ having the property of "hasEmbeddedElements" which points to an object $O1418C11$. This example captures a *fact*, where the document comment contains a note as an embedded element.

```
(MAIN::triple
(predicate  "http://www.sfu.ca/../Content.owl#hasEmbeddedElements")
(subject  "http://www.sfu.ca/../CILT.owl#01418C11documentComment")
(object  "http://www.sfu.ca/../CILT.owl#01418C11"))
```

MI-EDNA consists of two different variants of production rules - **pre-defined rules** and **user-defined rules**. The pre-defined rules consist of all the OWL meta-model primitives. The XSLT stylesheet as developed by Gandon and Sadeh[16] defines the OWL primitives for use in Jess. An example of the pre-defined rule such as *transitivity* in OWL meta-model is shown here.

```
(defrule MAIN::transitivity
(declare (salience 100))
(MAIN::triple (predicate ?p) (subject ?x) (object ?y))
(MAIN::triple (predicate ?p) (subject ?z) (object ?x))
(MAIN::triple (predicate
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
        (subject ?p)
        (object "http://www.w3.org/2002/07/owl#TransitiveProperty"))
=>
(assert (MAIN::triple (predicate ?p) (subject ?z) (object ?y))))
```

MI-EDNA has two different types of user-defined rules: the core rules concentrating on the SRL principles and the five different scaffolds for interaction; and the learner-centric rules originating from the learner. The SRL rules attempts to first match the patterns identified by Hadwin et. al. [33]. Assuming that the learning tasks have been identified by the initial candidate set of rules, additional candidate rules match the learning task and the learning task sequences into tactics and strategies of SRL. MI-EDNA, aside from building the working memory with additional inferred knowledge, also interacts with the learner as dictated by the consequents in the rules.

MI-EDNA has two distinguishing features. First, system-initiated queries and responses are based on data that gStudy gathers on the fly. For instance, a student can inquire about the number of times she has reviewed a glossary term in a session where she studied the technical material. Second, the topic of queries can be about the content and about study tactics as traced by gStudy when learners studied the content.

The details of rule implementation can be read in Appendix C. Example of rule that tracks the learning tasks and writes to the Working Memory.

```
(defrule Learner_CC (declare (salience 100))
(triple (predicate
    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#createdBy")
(subject   ?DocumentComment1)(object   ?Learner))
```

[16] http://www.cs.cmu.edu/ sadeh/MyCampusMirror/OWLEngine.html

```
(triple (predicate
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasEmbeddedElements")
(subject   ?D2&:(eq ?D2 ?DocumentComment1)) (object    ?Element1))
(triple (predicate
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasGlossaryTemplate")
(subject   ?E2&:(eq ?Et2 ?Element1)) (object    ?Template1))
(triple (predicate
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#glossaryElement")
(subject   ?T2&:(eq ?T2 ?Template1)) (object    "concept"))
(triple (predicate
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasGlossaryTVersion")
(subject   ?T&:(eq ?T ?Template1)) (object    "0"))
=>
(assert (tmp_cnt (LearningTask "gCC")(Learner ?Learner)
    (Template  ?Template1)))
(assert (tmp_LT_ref (LearningTask "gCC")(Learner ?Learner)
(Template  ?Template1)))
(assert (triple
(predicate
    "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#IdentifyingNewConceptsTT")
(subject  "gCC")(object   ?Learner)))
(assert (triple
(predicate
    "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#NamingItemsToLearnTT")
(subject  "gCC")(object   ?Learner)))
(printout t ?Learner "Took Concept" crlf))
```

Example of rule that maps it back to the SRL phases and interacts with the learner.

```
(defrule Learner_SRLPerformancePhase (declare (salience 100))
(triple (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject  ?Learner)(object
        "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
?fid <-  (tmp_TS (Strategies ?Strategy)
    (Learner ?L&:(eq ?L ?Learner))(Tactics ?Tactic))
(triple (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject ?LearnerName)(object ?L1&:(eq ?L1 ?Learner)))
```

```
=>
(bind ?LS (run-query* search-template-tmp_TS ?Learner))
(bind ?tempS "")
(while (?LS next)
    (foreach ?item
        (create$ "Organization_TS" "Elaboration_TS" "CriticalThinking_TS")
        (if   (eq ?item (?LS getString s))
        then
            (bind ?tempS (str-cat ?tempS " "?item))
        )
    )
)
(miedna_informs (str-cat ?LearnerName"
        is using these Strategies - " ?tempS "
        during the SRL Performance Phase")))
```

This rule-based reasoning expands the inferencing capabilities of MI-EDNA beyond DL-based reasoning. The rule-based reasoning leads to a more natural mode of communication and interaction between the learner and the system.

3.3.4 Interfaces

MI-EDNA, in combination with gStudy, provides various interfaces for learner interactions and for the system to initiate interaction with the learner. The learner in gStudy figure 3.12[17] can view, browse, label the highlights with quicknote, take detailed annotated notes, link concepts with glossary, and create other links within the content.

Learners can interact with the system or with other learners using a chat tool that is provided with gStudy and MI-EDNA. The chat tool figure 3.13 allows learners to collaborate with other learners working on the same material or the same learning task. This chat interface also acts as a mode of communication for the system to initiate interaction with the learner based on their observed learning styles and patterns. The system initiates interactions based on the SRL patterns of the learner to help the learner self-regulate.

For example, when the learner creates notes that are of the same type, then MI-EDNA can non-intrusively inform the learner about other patterns of note-taking. This interface can also be used by the learner to post queries to interact directly with MI-EDNA in order to self-regulate their

[17]gStudy software used with permission from Principle Investigator and as part of the research team

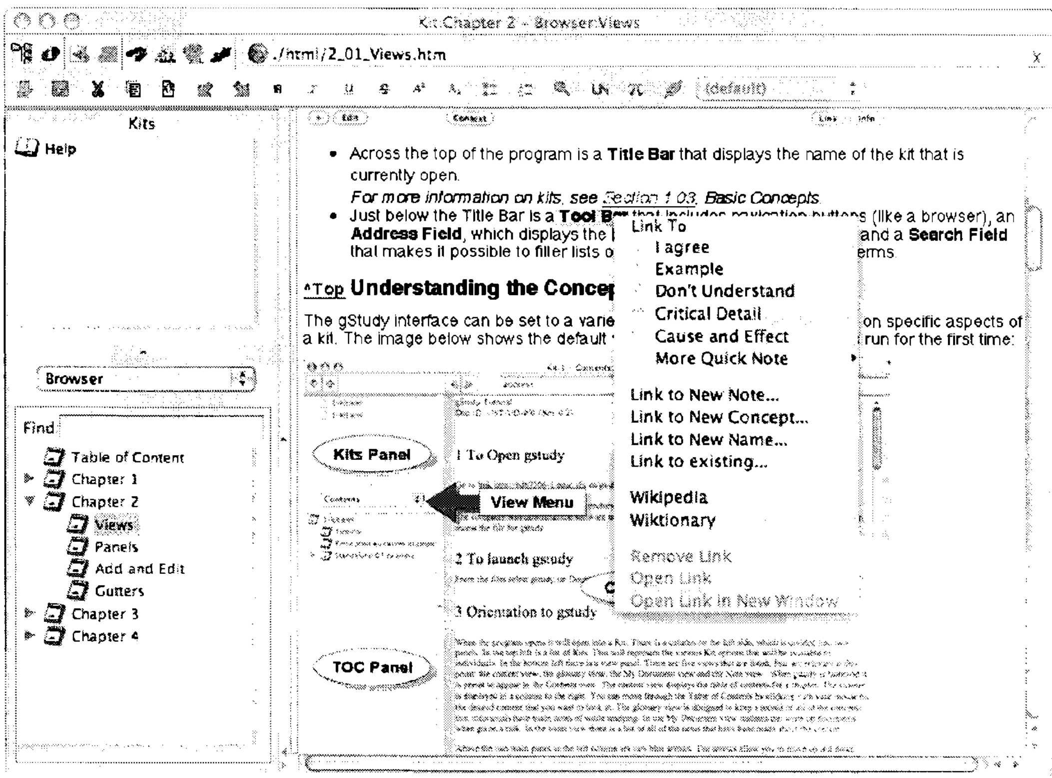

Figure 3.12: gStudy Interface

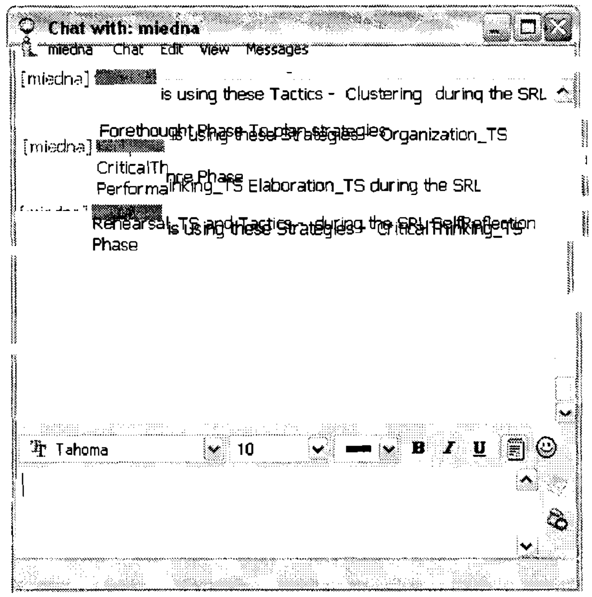

Figure 3.13: gChat Interface

learning style. For example, a learner can build complex queries, such as what percentage of students in his/her class i) highlight text, ii) immediately make a note, iii) link that note to relevant glossary items, and iv) score more than 85 percent on the test covering the assignment, in that order.

These interfaces from gStudy and MI-EDNA provide a window for interaction and communication between the learner and the system. Through these interfaces, Mixed-Initiative interactions in MI-EDNA occurs with the goal of helping learners to self-regulate their learning.

3.3.5 Mixed-Initiative Recognition and Dissemination

The objective of MI-EDNA is to have a system with MII characteristics. MI-EDNA aims for adaptivity at multiple levels of granularity such as *learner interaction level*, *learner task level*, *learner goal level*, and so on. As seen in the functional architecture Section 3.2.2, the system aims to track the cognitive model of the learner to trace the learning patterns with respect to the pedagogical strategies related to content, learner goals, and learner preferences.

Mixed-Initiative interaction in MI-EDNA follows the pair-programming model [90], where the expert is mainly an observer with open-ended opportunities to initiate interaction. There are no specified cases or specified situations for the expert to initiate interaction. The interaction, in the case of pair-programming, is mostly dependent on the expert programmer and the knowledge level of the novice [47]. In an analogical approach, MI-EDNA **passively observes** learner interactions,

recognizes opportunities for initiatives, and **actively initiates** interactions that are based on the principles of Self-Regulated Learning.

As a **passive observer**, learner interactions with the content are instantiated into the ontology. As explained in Section 3.3.3, the interaction data yields learning patterns of learners. MI-EDNA performs an analysis of these patterns based on the formally represented tactics and strategies. The SRL patterns that are represented as rules act as the knowledge processor and MI-EDNA uses these rules to **recognize opportunities** to initiate interaction with the learner. The dissemination of these opportunities are based on the scaffolding/fading techniques [34] [5]. Some very specific scaffolding/fading techniques are listed below, which provide a structural support for the interactions.

1. Guidance to learners on navigation of content (*content scaffold*),
2. Methods they use to study/solve problems (*process scaffold*),
3. How much they have learned (*knowledge of results*),
4. How learner's peers study and what they score on tests (*normative scaffold*), and
5. Supporting learners based on the context of interaction (*context scaffold*)

Thus, MI-EDNA recognizes the exact opportunities when the system (or the users) should take control of the interaction and the initiator is in a position to provide explanations. Some of the production rules that characterize MI-EDNA's ability to recognize opportunities for initiative-taking were discussed in Section 3.3.3.

Explicitly recognizing opportunities for initiatives by the system or by the user is imperative to the success of mixed-initiative systems. Employing production rules to recognize opportunities for initiatives based on well-founded theories, MI-EDNA is able to formally ground and analyze learner-system interactions.

SRL based initiatives encourage learners to plan their learning process, to monitor their emerging understanding, to use different strategies to learn, to handle task difficulties and demands, and to assess their emerging understanding in comparison with other learners. In essence, by **actively initiating** interactions, MI-EDNA helps a learner to self-regulate as well as to co-regulate with fellow learners. This active initiation is based on the scaffolding/fading technique as shown the table 3.2.

Azevedo and Hadwin [5] emphasis on the scaffolding/fading being the gradual control of support calibrated for the learner and the task. The scaffolded interaction involves three main characteristics - diagnosis, calibration and fading. The notion of the scaffolding is to have appropriate control in representing, managing and enacting an interaction at an appropriate pace for the respective learner [34]. Diagnosing the appropriate needs for scaffolded support for complexity of the task, calibrating the scaffolded interaction to the learner and fading the interactions with the learner knowledge leads

Table 3.2: Scaffolding Opportunities

Categories	Sample System Scaffold Initiation Opportunities	Rules example	Tactic/Strategy Suggested
Content Scaffolds	Goal setting strategy suggestions	The system evaluates **the topic** under study and helps the learner form a set of short- and long-term **goals**.	Goal Formation
	Providing reference material	The system provides sources of information that are **relevant to the set of goals** established for the given content.	Goal Oriented Resource Identification
	Help learner focus attention on important sections	If the learner takes **notes only on certain topics,** then system recommends **other document locations** that are linked to the same topic, which the learner may not be aware of.	Linked Resource Identification
Process Scaffolds	Reminders for incomplete tasks	If the learner takes notes but doesn't write anything in the notes, the system recommends filling in short descriptions to help the learner recall.	Self Study Improvisation Tactics
		Students may be prompted to go back to **'question', 'to do', 'don't understand'** and **'debate'** notes on which they have not elaborated.	Task Recollection
	Based on the learner's interaction, the system can guide him/her specific SRL tactics	If the learner is only **highlighting**, then the system recommends the use of **taking notes** to the highlighted text.	Self Study Improvisation Tactics
Learner Scaffolds	System compares results of learner interactions with goals	System evaluates learning outcomes and compares the progress with respect to the established goals, giving a performance appraisal.	Goal Based Evaluation
	The system provides feedback on the learner's current knowledge level	The system can show the **learners' knowledge status** based on the learner's model.	Progress Level Reflective Motivation
Normative Scaffolds	Based on peer interactions and accomplishments, the system can initiate comparative accomplishment statistics	The system shows the learners their standings in regards to the score or the learning style with respect to their peers.	Comparative Motivation
		If a majority of learners spent a considerable amount of time reading a section, then associate the speed of reading with the corresponding content and the performances of learners in evaluation exercises related to that content.	Collaborative Emulation
Context Scaffolds	Context-Based Help	If the learner highlights a word and makes a **'Don't Understand'** note then the system searches in the Glossary for that word and then provides the reference to the learner.	Knowledge of Correct Response

to an adaptive interaction. Hence, to provide appropriate scaffolded support to the individualized learners creates a gradual shift from *system-directed regulation* in SRL to *learner-directed regulation.*

The mixed-initiative interaction environment creates the grounds for scaffolding/fading of interaction. Scaffolded knowledge dissemination paves the way to build a dynamic and robust interactive system that offers recommendations based not only on the asynchronous actions of the learner but also on how these actions map on to the principles of SRL. Thus, inherently the system gains a *degree of initiative taking*[18], while being dependent on the actions of the learner. The calculation of the degree of initiative taking corresponding to a set of interactions determines the fading effect (systematic and graceful degradation of support) of the system with respect to the requirements and goals set by the learner. Thus, the recognition of system initiatives is dictated by the production rules that encode the principles of SRL, while the dissemination is based on the scaffolding/fading techniques.

3.4 Synthesis

This section described the architecture, the design, and the implementation details of MI-EDNA. The technical architecture identified the components of the system, while the functional architecture displayed the information flow in the system. The third section elaborated on the individual system components that capture knowledge in DL-based ontological formalism, transformation of the CILT ontology into JESS facts and rules, system recognition of initiation points (opportunities for MII) based on the TTS ontological rules, system actively initiating interactions with the learners based on the scaffolding/fading techniques, and various interfaces that are available for learners to engage in these interactions. The next chapter presents an analysis of how and why MI-EDNA recognizes opportunities for mixed-initiatives and identifies application areas where such theory-centric mixed-initiative approaches can be deployed successfully.

[18]Currently, the degree of initiative taking is dependent on the frequency of opportunities for system initiations and the importance of the category of the scaffold feedback.

Chapter 4

Analysis

This chapter will concentrate on the analysis of the effectiveness of implementation of research objectives. The research objectives and the manner in which they are addressed in MI-EDNA highlight four major endeavours of this research: knowledge engineering, knowledge dissemination, mixed-initiative interaction, and domain knowledge representation. An ontological framework has been chosen as the underlying formalism with the aspiration of formally capturing the domain specific knowledge and to provide interoperable and shareable access to the knowledge. The ontological framework enriched with Description Logic and the expressive power of production rules have equipped MI-EDNA with a strong basis to recognize opportunities for mixed-initiation based on educational theories. Finally, the scaffolding/fading techniques of MI-EDNA accommodate the dissemination of the principles of these theories with the intent of helping learners regulate their learning habits.

In essence, the research outcomes in each of these objectives will be presented and analyzed with regards to the benefits and drawbacks of the approaches.

4.1 Ontological Representation of Domain Knowledge

As mentioned in Chapter 1, the first and the fourth objectives of the thesis aim to develop an ontological framework to represent various types of knowledge in the domain of education.

4.1.1 Knowledge Engineering

The foremost knowledge required to achieve the goals of MI-EDNA were obtained from two sources:

1. the theory of self-regulated learning in Educational Psychology, and

2. a theory-centric analysis of learner interactions that were observed when they executed well-defined learning tasks

The literature review (Section 2.3.2) on SRL reveals that knowledge of SRL theory and the associated principles can be explicitly mapped onto specific tactics and strategies. Aside from the literature review, discussions with the experts[1] in the field lead to clarification on the philosophy advocated by SRL, an interpretation of the application of SRL, and various aspects of the learning patterns promoted by SRL.

Obviously, SRL is very conceptual in nature and different researchers have different perspectives on the interpretation of the metacognitive theories behind SRL. Encoding the knowledge embedded in an abstract theory such as SRL, first of all, requires explicit definitions of the educational concepts observed in the theory and restricting the scope of the defined concepts with respect to what can be stored in a knowledge representation scheme. The accuracy of such a mapping of the educational concepts to the computational concepts depends heavily on intensive negotiations between the educational experts who provide the basis of the knowledge and the computational experts who provide the basis for the representation of the knowledge. These negotiations govern the intricate balance between the knowledge that needs to be captured and the knowledge that can be captured. Learning Kit meetings are held every week, where the principal investigators, the software developers, and the student researchers can engage in these negotiations.

The process of formally representing the vital elements of the theory captures the meaning of the theory in terms of the interpretations of learner interactions. A schema that observes and classifies learner interactions observed within the gStudy environment provides the basic information required to build the components of the ontology. The process of knowledge engineering that resulted in the development of the components of the ontology benefitted tremendously from close consultations with the domain experts.

Discussion

The ontological representations of the SRL theory and the observed learner interactions form the core of MI-EDNA's knowledge. The validity in the ***accuracy*** of the representation underwent a process of:

- continual discussion and presentation of the conceptual ideas to the domain experts,
- demonstrations of versions of the ontology to the domain experts, and
- publication of the ideas through academic channels (e.g., [51])

[1]e.g., tutorial on SRL by Dr Dianne Jamieson-Noel on SRL, and personal meetings with the Principal Investigators of the Learning Kit project

The use of DL constructs and axioms in representing SRL in an ontology has resulted in an *explicit* representation of the theory. This explicit representation has two major benefits: first, a computable representation and second, a shareable blueprint of the theory. Thus, different parts of the ontology corresponding to different aspects of SRL were verified and validated with the help of the domain experts. Importantly, the representation of learner interactions as ontological properties creates a glue between the *learner* ontology, the *content* ontology, and the *time* ontology. Again, the validity of these connectivities were validated by the domain experts.

Thus, the knowledge engineering process that encoded the SRL theories and the gStudy learner interactions, and the resultant pieces of the knowledge were verified and validated.

4.1.2 Ontology Management

The analysis of Ontology Management in MI-EDNA concerns two types of data that are being used for the instantiation of the ontology - *static data* and *dynamic data*.

The *static data* is developed once and as the name implies, it rarely changes. Also, any change in static data occurs in an organized and methodical process since it may impact the quality of the content. Primarily, the kit[2] content developed for gStudy is considered as the static data. The kit content is parsed and instantiated in MI-EDNA's application ontology (CILT).

The process of creating the static data involves human experts who create not only the content but also the meta-tags that identify the structure of the content. Importantly, the meta-tags can also include instructional tags such as 'important paragraph to read', 'subsection that provides arguments in support of the hypothesis', 'key concept sentence', and so on. The meta-tags are used to parse the content and appropriately instantiate the ontology. Two chapters have been created in the domain of "Java Programming" for the purposes of this analysis, using DocBook[3] specifications and its editor. Further, the chapters have been tagged based on the structural information[4] of the content. Then, the content of these chapters were instantiated into the ontology.

The *dynamic data* continually evolves. The learner interactions form the core of the dynamic data. Interactions between the learners and the content (using gStudy or other associated toolkits), with respect to time, are captured as log files. These log data files are then used to instantiate the ontology. Excerpts of the log files of the interaction data are attached in Appendix B.

For this analysis, the log data that were obtained from the interactions of 8 students with gStudy were filtered through the automated ontology instantiator in MI-EDNA. The instantiator extracted interaction-specific data (e.g., createdBy, updatedBy, highlights, deletedBy, and so on) and stored it in the ontology. The following excerpts shows the log file on the top and the instantiated ontology

[2]Kit – > The learning content packaged for gStudy learning tool

[3]http://www.docbook.org/

[4]we did not include any instructional tags

on the lower half.


```
- <ModelEvent action="created" target="concept"
timeStamp="2004-10-29T09:15:14.587"> - <targetObject
kitID="dca7d0_ffd66f03b9__7fff" kitName="NEW Educ 220"> - <Concept
author="instructional" dateCreated="2004-10-29T09:15:14.501-07:00"
dateModified="2004-10-29T09:15:14.501-07:00" id="1286"
name="Concept_1" templateRef=""
xmlns="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/glossary">
  <Description />
  <Definition />
  <Examples />
  </Concept>
  </targetObject>
  </ModelEvent>
```


instantiated into $-->$


```
<C:DocumentComment rdf:ID="01286_C12_concept_documentComment">
 <I:createsAt>
   <time-entry:DurationDescription rdf:ID="01286_C12_concept_duration">
    <time-entry:hours
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">09</time-entry:hours>
    <time-entry:minutes
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15</time-entry:minutes>
    <time-entry:seconds
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">14</time-entry:seconds>
    <time-entry:years
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2004</time-entry:years>
    <time-entry:months
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">10</time-entry:months>
    <time-entry:days
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">29</time-entry:days>
  </time-entry:DurationDescription>
 </I:createsAt>
 <I:createdBy rdf:resource="#C12"/>
    <C:hasEmbeddedElements rdf:resource="#01286_C12_concept"/>
    <C:contentLKitID rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        dca7d0_ffd66f03b9__7fff</C:contentLKitID>
    <C:contentLKName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        NEW Educ 220</C:contentLKName>
        ...
```


Maintaining the ontology in a consistent state requires consultations with the end users (the researchers and the learners) as well as consultations with ontology engineers. End users verify whether the ontology stores the right data and the experts validate whether the ontology stores the data right.

Performance Analysis of the Instantiator

Semi-automatic and automatic instantiators are essential components of MI-EDNA since data obtained from a variety of gStudy toolkits needs to be populated into MI-EDNA as and when the data becomes available. MI-EDNA can only work as effectively as the accuracy of knowledge represented and captured, and as ***efficiently*** as the knowledge capturing process. The nature and purpose of the static data leads to the content data being instantiated in the ontology *semi-automatically* because the author of the content and the ontology engineer will have to add or modify the tags using docBook. Also, any changes to the content and the corresponding changes in the ontology will have to go through a *semi-automatic* process that involves an ontology engineer. The efficiency of the instantiation is not as critical as the validity (correctness) of the instantiated knowledge. In the context of semi-automatic instantiation, involvement of the content creators, in addition to the ontology engineers, is preferable. The validity of the Java Programming content by the domain experts (Researchers in Learning Kit) is one major aspect of the validity of the instantiated content ontology. Further, the instantiated ontology in MI-EDNA also underwent a validation process that involved an ontology engineer reviewing the instantiated ontology.

Learner interactions can be instantiated into the ontology at real-time and MI-EDNA can provide feedback to the learners at real-time. However, for the purposes of this analysis only data obtained from gStudy logs were used. Log files of varying sizes were obtained from gStudy and were subjected to independent instantiation within MI-EDNA.

In order to analyze the performance of the ontology instantiator log files of varying sizes obtained from authentic learner interactions were used. The size of the initial CILT ontology was $58822KB$; then, each user log file with varying file-sizes as listed in Table 4.1 was fed into the ontoParser[5]. The ontoParser parsed these files and updated data corresponding to each user interaction in the CILT ontology. The size of the instantiated ontology based on the input log files from eight learners is $1814279KB$. This graph shows that the performance of the ontoParser varies in time taken to instantiate with respect to the file size. For instance, the instantiator took 31 milliseconds to upload a data file of size $15717KB$ while another data file of size larger size $65224KB$ took only 25 milliseconds to load.

Figure 4.1 represents the graph of the time it took for the ontoParser to load each file. This graph shows that the time to instantiate grows *relatively* linear with respect to the file

[5] second component of MI-EDNA

Table 4.1: Data for Time Analysis

learner	file size(kb)	time($millisec$)
C12	4746	4
C12	21229	11
C11	21805	6
C11	39282	110
C11	65224	25
C11	111416	48
C21	27988	44
C21	48219	20
C21	52468	21
C22	230530	90
C22	579212	283
C22	1128468	409
C31	15715	31
C31	33321	39
C32	558989	287
C32	585157	314
C32	589179	407
C31	728887	245
C42	8084	2
C42	21217	8
C41	25167	13
C42	43762	16
C42	46798	17
C42	51636	21

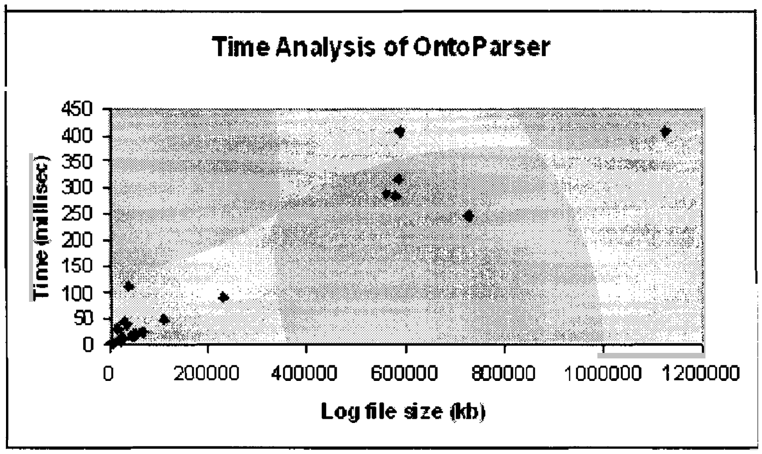

Figure 4.1: Ontology Instantiation

size. The calculated correlation coefficient of 0.94, between the *log file size* and the *time* it took to instantiate, indicate that the two variables are strongly associated with one another. Though there is strong relation between the two variable, some variance that exists are because of the type of interactions that exists in the log file. The log file consists of different learner patterns, the time it takes for the ontoParser to instantiate is relative to the number of learner interactions in the log file. Thus, the data being populated into the ontology needs to be handled in a modular manner to achieve a required operational efficiency in MI-EDNA. Some of the approaches that one can adopt to improve the performance of MI-EDNA with an increasing number of instantiated data will be discussed in the next chapter.

The discussion so far has been concerned with instantiating the log data from files. The instantiator is also able to handle real-time data, which significantly reduces the issues related to the size of the file. With real-time data coming in as a data-stream and the instantiated data being generated as stream-data, the time consumption for instantiating the interaction data will be dependent more on the types of interaction rather than the number of interactions. This time will be significantly less in comparison to the delay caused by the size of the file. Presently, gStudy is not capable of collecting and sending stream-data.

4.2 Recognition of Opportunities for Mixed-Initiative

The third objective is to design a mixed-initiative system and analyze the conditions under which it can be operational.

4.2.1 Rules for Initiatives

The instantiated ontology provides the theoretical knowledge for the system to recognize opportunities for initiatives. Through logical interpretations of the learner interactions that have been instantiated into the ontology, MI-EDNA is able to recognize the learning tasks identified in [33]. Interactions corresponding to a set of eight students have been used to verify whether MI-EDNA is capable of recognizing learning opportunities. These learning tasks are listed in Appendix E with explanation as to how these tasks are related to the learner interactions.

Further, MI-EDNA also maps the learning tasks onto tactics and strategies that learners have used in their learning process. These tactics and strategies are then mapped onto SRL phases to recognize additional opportunities for system initiatives. Production rules have been used to perform all these mappings–learning tasks to tactics, tactics to strategies, and strategies to SRL phases. In the process of self-regulating their learning styles learners can initiate interactions either by querying MI-EDNA or by interacting with gStudy.

MI-EDNA draws on the representation of learner interactions and the self-regulated model for system initiatives as well as on factors that promote mixed-initiatives. Although the theoretical background of initiative-recognition is based on SRL, the decisions for initiative-taking is based on the principles of mixed-initiatives. The factors of mixed-initiatives in MI-EDNA include *control*, *reactivity*, *communicative competence*, *negotiative ability*, *non-intrusivity*, *grounding*, and *affirmation recognition*. Presently, the mixed-initiative framework implemented in MI-EDNA does not take these factors into consideration.

Effectiveness of the Production Rules

The strength of mixed-initiatives in MI-EDNA is dependent on the ***accuracy*** and the ***frequency*** of how the system is able to recognize the learning tasks, tactics, and strategies.

The *accuracy* of MI-EDNA is analyzed through a *marginal* comparison of the results of MI-EDNA with the results observed in Hadwin et.al(2005). [33] populated the database with learner interactions log file and are able to track learner patterns, count the frequency of the patterns, classify the patterns, and apply various statistical measures to visualize learner interactions associated with these patterns. The matrix 4.2, of the learner versus the learning tasks, presents a list of learning tasks MI-EDNA was able to recognize. These are opportunities for MI-EDNA to take the initiative to interact with the Learner, or to take the initiative on tasks related to coaching learners in self-regulation.

These learning tasks as identified in [33] provides significant indication of the learner's learning method. The learning task of highlighting has not been summarized in this table because of the limitation of the log file data. All the learning tasks recognized manually by Hadwin et.al's [33] are also recognized by the rules in MI-EDNA. Recognition of these learning tasks lead to recognition of learner tactics and strategies. The TTS ontology discussed in Section A is used to track learner engagement in various tactics and strategies. Tables 4.3 and 4.4 present the occurrence of some of the tactics and strategies identified in Zimmerman's SRL model.

The tables clearly show that MI-EDNA is able to recognize and count the occurrences of

Table 4.2: Learner and Learner Interaction Matrix

LearnerID	C11	C12	C21	C22	C31	C32	C41	C42
gCCNL	3	4	0	0	0	1	2	0
gCL	264	372	89	152	48	63	244	254
gCQLA	0	8	0	2	0	0	0	0
gCQLCE	0	1	0	0	0	0	0	0
gCQLCS	0	10	0	5	2	0	7	0
gCQLD	0	35	0	0	0	0	0	0
gCQLDA	0	2	0	0	0	0	1	0
gCQLDU	0	0	0	0	0	0	1	0
gCQLE	18	24	7	12	5	6	16	17
gCQLI	28	47	8	27	7	0	0	0
gCQLM	1	0	0	0	0	0	0	0
gCQLNR	0	2	0	0	0	0	0	0
gCQLP	1	5	0	0	0	0	0	0
gCQLR	85	98	0	5	1	0	80	82
gCQLRR	0	7	0	0	0	0	0	2
gCQLS	6	0	0	5	2	0	0	0
gCQLTB	0	1	0	0	0	0	0	0
gCQLTH	0	3	0	0	0	0	0	0
gCQNL	0	2	0	0	0	0	1	0
gCSN	153	156	67	135	59	66	142	149
gUCN	3	4	0	0	0	1	2	0
gUDN	0	0	0	0	0	6	0	0
gUQDU	0	0	0	0	0	0	4	0
gUQE	0	0	0	0	4	0	6	0
gUQI	0	2	0	0	1	0	0	0
gUQR	0	0	0	0	1	0	0	0
gUQRR	0	1	0	0	1	0	0	0
gUQS	0	0	0	0	0	0	0	0
gUSN	145	148	0	132	59	64	135	142

Table 4.3: Learner Vs. recognized Teaching Tactics

Learner	C11	C12	C21	C22	C31	C32	C41	C42
Clustering	1	2	1	1	1	1	1	1
Creating Analogies	1	2	1	2	2	1	2	1
Identify Main Ideas	2	2	1	3	3	0	1	0
Identify New Concepts	1	1	0	0	0	1	1	0
Making Decision	1	1	0	1	1	0	0	0
Naming Items To Learn	1	2	1	1	1	0	0	1
Outlining	0	0	0	0	0	0	0	0
Paraphrasing	2	2	1	1	1	2	2	1
Questions	0	1	0	0	0	0	2	0
Summarizing	1	1	1	1	1	1	1	1

Table 4.4: Learner Vs. recognized Teaching Strategies

Learner	C11	C12	C21	C22	C31	C32	C41	C42
Critical Thinking	*	*	–	*	*	*	*	–
Elaboration	*	*	*	*	*	*	*	*
Organization	*	*	*	*	*	*	*	*
Rehearsal	*	*	*	*	*	*	–	–

learning tasks, tactics, and strategies observed in learner interactions. At this time, gStudy tracks only a limited set of learner interactions and hence the production rules of MI-EDNA are designed to recognize only a limited set of tactics and strategies. However, MI-EDNA introduces a novel methodology to map patterns of interactions all the way to the phases (states) of models of SRL as shown in Figure 4.2. Importantly, MI-EDNA is now capable of delivering theory-centric feedback through mixed-initiative dialogues at a higher level of abstraction using these inferred tactics and strategies.

MI-EDNA initiates interaction with the learner on the basis of the MI opportunities recognized by the system. That is, MI-EDNA proactively provides feedback to the learners only after recognizing the opportunities. However, MI-EDNA's initiatives are designed to be non-intrusive; that is, learners can choose to completely ignore MI-EDNA's feedback. On the other hand, MI-EDNA allows learners to query the system to reflect on their learning patterns based on the feedback information provided by MI-EDNA; that is, the preconditions for the recognition of MI opportunities are also stored within MI-EDNA and learners can query these preconditions to understand the theoretical basis of MI-EDNA's initiatives.

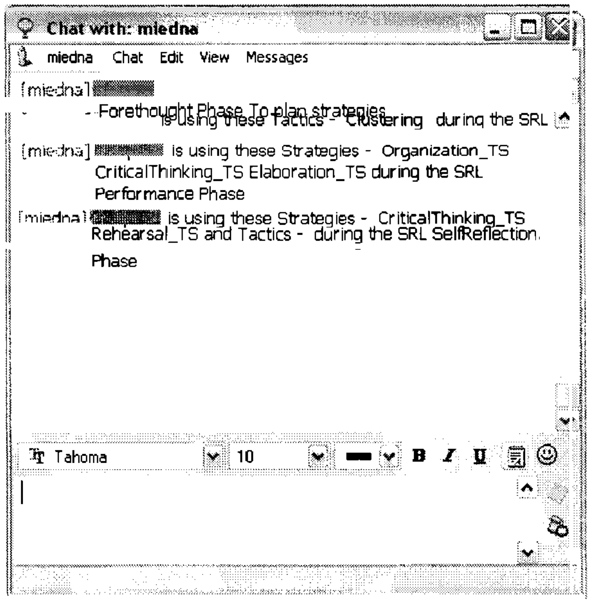

Figure 4.2: MI-EDNA responds while observing SRL Phases

As mentioned earlier, the theoretical basis of MI-EDNA's initiatives could also be augmented with factors (control, reactivity, communicative competence, negotiative ability, non-intrusivity, grounding, affirmation recognition, and so on) that govern the principles of mixed-initiatives.

4.3 Theory-centric Dissemination Based on Scaffold Principles

This section discusses how MI-EDNA recognizes and utilizes theory-centric and explanation-aware knowledge dissemination opportunities.

4.3.1 Dissemination

The first two sections of this chapter analyzed the representation of SRL and recognition of the components of SRL models. This section concentrates on appropriate dissemination of the SRL principles, which is one of the major goals of MI-EDNA.

As discussed in Section 3.3.5, MI-EDNA uses five different scaffold categories to support the learner to self-regulate their learning. The antecedents of the production rules recognize opportunities for mixed-initiative interaction. The consequents of the production rules disseminate aspects of SRL principles.

The dissemination is based on the philosophies of scaffolding [34] in which the interaction by the system is based on the diagnosis of when the learner needs support, calibrated support based on the learner and task, and fading of the scaffolding is based on the learner knowledge. This dissemination evolves gradually alerting the learners to take responsibility to self-regulate by providing educational resources, problems or tasks when needed, cognitive guidance, and feedback on their learning habits and performances.

Discussion

This research presents a framework of scaffolding the type of self-regulated learning [5] for theory-centric dissemination with an analysis on the type of scaffolds from which learners can benefit. Production rules present a flexible mechanism to disseminate(provide feedback) information based on the theory of SRL.

Table 4.5: Scaffold/Fading Techniques triggered for Learners

Scaffold	$C11$	$C12$	$C21$	$C22$	$C31$	$C32$	$C41$	$C42$
1a. ProvideReference	–	*	–	–	–	–	*	–
1b. FocusImportantSection	*	*	*	–	*	*	–	*
2a. StudyImprovisation	–	*	*	–	*	–	*	*
2b. TaskRecollection	*	–	–	*	–	*	–	–
3a. LearnerKnowledge	*	*	*	*	*	*	*	*
4a. ComparativeMotivation	*	–	*	–	*	*	–	*
5a. ContextReference	–	*	*	–	–	*	–	–

For each of the eight learners, MI-EDNA triggers a combination of these scaffolds based on the learner's prior knowledge, the learning task and interactions. Table 4.5 lists the learners and the scaffolds that were triggered based on learners interaction patterns[6]. The scope of this research was to identify the dissemination process and to develop a simple framework to enable the process. The scaffolds present one such framework. The validity or the effectiveness of the scaffolded dissemination is a potential future research direction that

[6]Some of the query examples for the scaffolded categories are mentioned in Chapter 3 and the details of the rules and queries are in Appendix E

one can pursue in light of the fact that the area of Intelligent Tutoring Systems identifies many of the scaffolding and fading principles [34].

4.4 Application

This section presented an analysis of MI-EDNA's *process and validity* regarding knowledge engineering, the *accuracy* of ontological representations of online content, content-oriented interactions, learner characteristics, time, teaching tactics, teaching strategies, and self-regulatory aspects. The section also explored the *process* of instantiating the assertional knowledge in the ontology, in an automatic or a semi-automatic fashion. Further, an analysis of the application of the reasoners based on Description Logic and Productions Rules was presented. This analysis identified techniques that recognize the *frequency* of regulatory behaviour of learners and initiation opportunities for mixed-initiative interactions. Finally, the process of knowledge dissemination based on a theory-centric approach was analyzed and the results are presented with respect to data obtained from the interactions of 8 learners.

A number of ideas that center around this research have been published and they speak for the academic rigour of the research approach [84], [83], [51], [61]. This process of analysis with respect to the degree to which learner interactions can be mapped onto the models of self-regulation provide a solid foundation for further research in the field of sharing learning experiences across multiple domains including programming, profiling, reading, composition, collaboration, and task understanding.

Chapter 5

Conclusion

This chapter presents a summary of the research accomplishments, discusses the scope and limitations, and highlights the future work.

5.1 Summary

This research explores the application of the formal representation of the SRL principles. Ontological Representation creates a feasible, shareable, and easily expandable knowledge base. MI-EDNA successfully captures and disseminates the tactics and strategies of SRL using ontologies. Using the same underlying instantiated ontology, inferencing is performed by engines that operate on Description Logic and production rules. This research contends that through MI-EDNA learners will have more opportunities to reflect on and regulate their learning processes. These opportunities for Mixed-Initiative interactions are formally (using ontologies) recognized based on the sequences of strategies and tactics used by the learners.

MI-EDNA offers scaffolded feedback using production rules. System-initiated interactions are aimed at the content of learning, the process of learning, the domain knowledge, the normative comparison of learning activities and performances, and, finally the context of learning. The system can initiate interactions with the learner to promote specific strategies and tactics with respect to the content and/or the context. It can also initiate interactions with the learner when it finds gaps in learner strategies and tactics. Further, it can initiate interactions with the learner with respect to the strategies and tactics employed by other students. Using these five recognizable opportunities, MI-EDNA provides contextualized feedback to learners, on the fly, as they study or solve problems in specific learning activities. Using a rule based inference mechanism for scaffolding paves the way for building a dynamic and robust interactive system that offers recommendations based on the asynchronous actions of the learner. Recognizing tactics that are enacted by the learners, the underlying strategy that spawned these tactics leads to recognizing opportunities for the delivery of SRL-based feedback.

In summary, MI-EDNA observes the fine-grained interactions of the learner with the online material; populates these interactions in an ontology; automatically translates these interactions into fine-grained tactics; predicts the coarse-grained strategies; matches these observed tactics and strategies against the optimal tactics and strategies prescribed by SRL; triggers system-initiated interactions to prompt and guide the learner who has strayed away from optimal SRL tactics and strategies; enables a logic-based query interface for learner initiated interactions; develops a cognitive model of skills of the learner; and attempts to revise the ontology based on the model.

5.2 Scope and Limitations

This research presents an approach for computational representation of the theories in educational domains using ontologies. Specifically, this research focuses on the ontological representation of the models of SRL. The representation of a theory in its entirety is a tremendous task. This research presents a viable solution for the same.

Validation of the completeness of the representation of SRL principles in an ontological format has not been one of the goals of this research. The research focused only on representing a limited subset of the principles advocated by an SRL model to exemplify the possibility of representing a complete model of SRL. Other aspects of SRL principles can be added to the ontology and further validation with the experts can be done as future work.

Another limitation of this research was the availability of the data for analysis. With the gStudy application not having the ability to provide streamed log data, the analysis relied on the real-world log data. Although this data provided the basis for analyzing the knowledge engineering, inferencing, and dissemination aspects of the research, the analysis on the performance of the complete system has not been done. This is a part of the future work that will be discussed in the next section.

Another factor is the limits on the interaction data that could be observed and recorded in gStudy. Presently, this interaction data does not address the complete set of factors and variables that contribute to the SRL model. This thesis indicates and proves MI-EDNA's ability to track the learner's cognitive model with respect to the SRL phases and to base the system initiatives based on such opportunistic recognition. However, the SRL model mapping is not complete and the system is only able to track certain states of SRL with the limitations on the interaction data.

Finally, the scope of this research project is to explore the recognition of opportunities for mixed-initiation. The interfaces used for MI interactions in MI-EDNA are presented and briefly discussed. However, neither the design factors of mixed-initiative interface nor the evaluation of this interface have been completely explored as part of this research. This is one of the core opportunities for future work in this research, which will be discussed in the next section.

5.3 Future work

As mentioned, this thesis only presents one piece of the puzzle in this research of *knowledge engineering of educational domain in an ontological format for dissemination based on the principles of MI.* With this initial presentation of the prototype development and conceptual presentation of the ideas through MI-EDNA, the future of this research is extensive. The feedback from the academia regarding the publications and presentations based on this research has been extremely positive. Some of the future work on this research has already been taken up by graduate students in the MI3 research group at Simon Fraser University. Some of the immediate future research directions include:

1. knowledge engineering of applied educational psychology domains such as reading [33], composition, and problem-solving [18],
2. modelling the self-regulatory capabilities of learners [11],
3. evaluating the influences of Mixed-Initiative interactions and interfaces
4. developing a cognitive model of the self-regulatory skills of the learner
5. employing MI-EDNA for co-regulated learning [19]
6. verifying and validating the underlying SRL model based on a cognitive model

7. providing a common ontological SRL framework for geographically-distributed learners and instructors in a blended online learning environment, and

8. explanation-aware SRL modeling and scaffolding [76]

MI-EDNA explored the initial phases of the domain of studying and how proactive SRL prompting influences the studying habits. Further research in the domains of composition and problem-solving are already underway.

This research focused on tracking learners' self-regulatory patterns. Future research, with the combination of user-modeling, can enhance MI-EDNA's rule-based pattern matching to accurately infer learners' self-regulatory capabilities. This would require complete learner interaction data and the factors that affect all states of self-regulatory behaviour.

MI-EDNA examined and presented a novel methodology to recognize opportunities for system initiatives to enable Mixed-Initiative interactions. MI-EDNA identified the various system initiative opportunities based on the principles of self-regulated learning. These initiatives are enacted by the system based on the scaffolding/fading techniques. This ground work of MI-EDNA creates an exciting opportunity for exploration of the various interfaces and factors related to Mixed-Initiative interactions.

MI-EDNA focused only on the principles of self-regulated learning. Employing MI-EDNA for the purposes of co-regulated learning is an interesting research expansion since the interfaces for interaction are set in a collaborative environment. The ongoing attempts to represent models of task analysis and collaborative learning in an ontology indicates the potential of this research.

Although the validation and verification of the SRL models based on the underlying cognitive model was not the focus of this thesis, it is an important piece of research that needs to be done in the future. The validation research would require extensive involvement from researchers working in the domain of cognitive sciences.

MI-EDNA provides the framework for explanation-aware dissemination of the SRL model. The dissemination based on the scaffolding/fading techniques provide explanation to the learners based on the underlying SRL model. The theory-centric representation and dissemination provides the framework for the explanation. Future work with respect to explanation involves engaging the SRL model and the process of scaffolding to explicitly or surreptitiously integrate corresponding explanations as part of the interactions and system-oriented initiatives.

In its entirety, MI-EDNA and the ongoing future research to enhance MI-EDNA aim at explicitly representing domain knowledge, integrating pedagogical knowledge, and creating shareable learner experiences. The instantiated ontologies, the observed learning patterns, and the corresponding production rule inferences fossilize learning experiences. These imprints of learning can be classified according to the criteria set forth by the institution or by the individual who owns these explicitly represented experiences. These experiences can be shared across geographical distances, recommended by institutions, and custom evaluated by individuals.

Experiences from a competent learner's problem-solving patterns, or group of experiences from an accomplished program at an institution, or a set of experiences of learning styles are some of the examples of shareable imprints of learning. These observable learning patterns can be captured through ontology engineering and management. The value and power of these learning experiences can bridge the geographical disparity in learning as well as increase the quality of learning for the individuals and for the institutions.

MI-EDNA along with the gStudy toolkit offer the infrastructure needed for capturing the experiences of learning patterns related to reading, writing, and problem-solving, among other domains. The learner experiences in the domain of reading that MI-EDNA makes available is currently being used for research in evaluating the learning styles and the competencies of learners.

This research focused on the opportunities MI-EDNA creates in the frontier of quality of education in institutions or amongst individuals. The core strength of MI-EDNA resides with its ability to enforce a tight integration between the learning practices and the educational theories. With the deployment of Mixed-Initiative interactions, based on Self-Regulated Learning principles, a learner-conducive communication occurs between the gStudy toolkit and the learner. Inferring the instantiated ontology yields theory-oriented explanations for Mixed-Initiative interactions. These experiences in learning patterns can be shared across individuals and institutions for balancing the quality of learning.

Appendix A

Ontology Files

This appendix lists two major ontologies that were created as part of this research. The ontologies are also posted on the web[1] [2] for reuse.

CILT OWL file:


```
<rdf:RDF
 xmlns:tzont="http://www.isi.edu/~pan/damltime/timezone-ont.owl#"
 xmlns:L="http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#"
 xmlns:time-entry="http://www.isi.edu/~pan/damltime/time-entry.owl#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:PC="http://www.sfu.ca/~ldoherty/RA/Jurika/ProgrammingOntologyOwl.owl#"
 xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
 xmlns:I="http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#"
 xmlns="http://www.sfu.ca/~shakya/ontology_lib/CILT.owl#"
 xmlns:C="http://www.sfu.ca/~shakya/ontology_lib/Content.owl#"
 xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xml:base="http://www.sfu.ca/~shakya/ontology_lib/CILT.owl"
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl"/>
<owl:imports rdf:resource="http://www.isi.edu/~pan/damltime/time-entry.owl"/>
<owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>3.2 Jurika Shakya</owl:versionInfo>
<owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Content.owl"/>
```


[1] *http : //www.sfu.ca/ shakya/ontologylib/CILT.owl*

[2] *http : //www.sfu.ca/ shakya/ontologylib/TTS.owl*

```
<owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Learner.owl"/>
</owl:Ontology>
<rdf:Description rdf:about=
"http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK">
<rdfs:subClassOf>
    <owl:Restriction>
    <owl:someValuesFrom>
        <rdf:Description rdf:about=
        "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentFragments">
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:someValuesFrom rdf:resource=
            "http://www.isi.edu/~pan/damltime/time-entry.owl#DurationDescription"/>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlightsAt"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:someValuesFrom rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"/>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlightedBy"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        </rdf:Description>
    </owl:someValuesFrom>
    <owl:onProperty rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlights"/>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#creates"/>
    <owl:someValuesFrom>
        <rdf:Description rdf:about=
        "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentComment">
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:someValuesFrom rdf:resource=
            "http://www.isi.edu/~pan/damltime/time-entry.owl#DurationDescription"/>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#createsAt"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
```

```
            <owl:Restriction>
            <owl:someValuesFrom rdf:resource=
            "http://www.isi.edu/~pan/damltime/time-entry.owl#DurationDescription"/>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#deletesAt"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#createdBy"/>
            <owl:someValuesFrom rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#updatesAt"/>
            <owl:someValuesFrom rdf:resource=
            "http://www.isi.edu/~pan/damltime/time-entry.owl#DurationDescription"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#updatedBy"/>
            <owl:someValuesFrom rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
            <owl:someValuesFrom rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"/>
            <owl:onProperty rdf:resource=
            "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#deletesBy"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        </rdf:Description>
    </owl:someValuesFrom>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#deletes"/>
```

```
    <owl:someValuesFrom rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentComment"/>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
    <owl:someValuesFrom rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentComment"/>
    <owl:onProperty rdf:resource=
    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#updates"/>
    </owl:Restriction>
</rdfs:subClassOf> </rdf:Description> </rdf:RDF>
```

TTS OWL file:[3]

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#"
  xml:base="http://www.sfu.ca/~shakya/ontology_lib/TTS.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >1.0 Jurika Shakya</owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Teaching Tactics and Strategies - SRL only</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="DoIReallyKnowItTS">
    <owl:disjointWith>
      <owl:Class rdf:ID="ConsiderAllFactorTS"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="BaggageClaimTS"/>
    </owl:disjointWith>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:ID="QuestionsTT"/>
              <owl:Class rdf:ID="IndependdentSeatWorkTT"/>
              <owl:Class rdf:ID="AcceptFailureTT"/>
            </owl:unionOf>
          </owl:Class>
```

[3]only excerpt from the ontology listed here

```
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="consistsOf"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
    <owl:disjointWith>
      <owl:Class rdf:ID="GalleryWalkTS"/>
    </owl:disjointWith>
    <owl:disjointWith>
...
  <owl:ObjectProperty rdf:about="#rehearsesBy">
    <rdfs:domain rdf:resource="#Rehearsal_TS"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#doesSelfControl">
    <rdfs:domain rdf:resource="#SRLPerformancePhase"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#thinksCriticallyBy">
    <rdfs:domain rdf:resource="#CriticalThinking_TS"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#doesSelfJudgement">
    <rdfs:domain rdf:resource="#SRLSelfReflectionPhase"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#hasPlannedStrategies">
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isCollectionOf"/>
  <owl:ObjectProperty rdf:about="#teaches">
    <rdfs:range>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#IndependdentSeatWorkTT"/>
          <owl:Class rdf:about="#DoIReallyKnowItTS"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:range>
    <owl:inverseOf rdf:resource="#isTaughtBy"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#doesTaskAnalysis">
    <rdfs:domain rdf:resource="#SRLForeThoughtPhase"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="hasLearnGoalOriented">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#hasGoal">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
```

```
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#hasStrategicPlan">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasSelfEffeciency">
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#hasMotivation">
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBenefits">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#AwardTT"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasOutcome">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasExpectedOutcome">
    <rdfs:domain rdf:resource="#SelfRegulatedLearningET"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  </owl:DatatypeProperty>
  <owl:FunctionalProperty rdf:ID="learns">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <owl:inverseOf>
      <owl:InverseFunctionalProperty rdf:ID="learntBy"/>
    </owl:inverseOf>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="hasAward">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:InverseFunctionalProperty rdf:about="#learntBy">
    <owl:inverseOf rdf:resource="#learns"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </owl:InverseFunctionalProperty>
</rdf:RDF>
```

Appendix B

Sample Log file

Researchers from the Learning Kit Project conducted an experiment on the utility of gStudy and collected usage data from over 200 student participants. MI-EDNA uses the log files generated from gStudy to instantiate the ontologies. This appendix only presents an excerpt from one of the log files, since the log files are rather long. The user name has been replaced with a random ID.

Raw XML Log File Sample:


```
 - <Events user="C12"> - <ModelEvent action="updated"
target="toc" timeStamp="2004-10-29T08:55:24.925"> - <targetObject
kitID="dca7d0_ffd66f03b9__7fff" kitName="NEW Educ 220"> - <ToCEntry
author="instructional" dateCreated="2004-10-13T16:37:54.657-07:00"
dateModified="2004-10-29T08:55:24.469-07:00" id="1004" name="Chapter
7" templateRef="" type="Section"
xmlns="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/toc">
- <ns1:links
xmlns:ns1="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model">
- <ns2:link author="LK1" dateCreated="2004-10-13T16:37:54.889-07:00"
dateModified="2004-10-13T16:37:54.889-07:00" id="1005"
templateRef=""
xmlns:ns2="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns2:destination>
  <ns2:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0700_Ch07.htm" targetKitID="Sample"
  useRange="false" />
  </ns2:destination>
  </ns2:link>
  </ns1:links>
```

```
- <ToCEntry author="LK1" dateCreated="2004-10-13T16:39:54.441-07:00"
dateModified="2004-10-26T12:08:36.990-07:00" id="1028" name="6.
Summary" templateRef="" type="Section"> - <ns3:links
xmlns:ns3="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model">
- <ns4:link author="LK1" dateCreated="2004-10-13T16:39:54.479-07:00"
dateModified="2004-10-13T16:39:54.479-07:00" id="1029"
templateRef=""
xmlns:ns4="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns4:destination>
  <ns4:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0706.htm" targetKitID="Sample"
  useRange="false" />
  </ns4:destination>
  </ns4:link>
  </ns3:links>
- <ns5:link author="LK1" dateCreated="2004-10-13T16:39:54.479-07:00"
dateModified="2004-10-13T16:39:54.479-07:00" id="1029"
templateRef=""
xmlns:ns5="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns5:destination>
  <ns5:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0706.htm" targetKitID="Sample"
  useRange="false" />
  </ns5:destination>
  </ns5:link>
  </ToCEntry>
- <ToCEntry author="LK1" dateCreated="2004-10-13T16:38:17.562-07:00"
dateModified="2004-10-26T12:07:14.561-07:00" id="1008" name="1.
Overview" templateRef="" type="Section"> - <ns6:links
xmlns:ns6="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model">
- <ns7:link author="LK1" dateCreated="2004-10-13T16:38:17.630-07:00"
dateModified="2004-10-13T16:38:17.630-07:00" id="1009"
templateRef=""
xmlns:ns7="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns7:destination>
  <ns7:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0701.htm" targetKitID="Sample"
  useRange="false" />
  </ns7:destination>
  </ns7:link>
  </ns6:links>
- <ns8:link author="LK1" dateCreated="2004-10-13T16:38:17.630-07:00"
dateModified="2004-10-13T16:38:17.630-07:00" id="1009"
templateRef=""
xmlns:ns8="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns8:destination>
  <ns8:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
```

```
  rangeStart="-1" target="html/0701.htm" targetKitID="Sample"
  useRange="false" />
  </ns8:destination>
  </ns8:link>
  </ToCEntry>
- <ToCEntry author="LK1" dateCreated="2004-10-13T16:38:35.935-07:00"
dateModified="2004-10-26T12:07:29.026-07:00" id="1012" name="2.
Elements of the Cognitive Perspective" templateRef=""
type="Section"> - <ns9:links
xmlns:ns9="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model">
- <ns10:link author="LK1"
dateCreated="2004-10-13T16:38:36.041-07:00"
dateModified="2004-10-13T16:38:36.041-07:00" id="1013"
templateRef=""
xmlns:ns10="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns10:destination>
  <ns10:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0702.htm" targetKitID="Sample"
  useRange="false" />
  </ns10:destination>
  </ns10:link>
  </ns9:links>
- <ns11:link author="LK1"
dateCreated="2004-10-13T16:38:36.041-07:00"
dateModified="2004-10-13T16:38:36.041-07:00" id="1013"
templateRef=""
xmlns:ns11="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns11:destination>
  <ns11:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0702.htm" targetKitID="Sample"
  useRange="false" />
  </ns11:destination>
  </ns11:link>
  </ToCEntry>
- <ToCEntry author="LK1" dateCreated="2004-10-13T16:38:53.060-07:00"
dateModified="2004-10-26T12:07:47.791-07:00" id="1016" name="3.
Information Processing Model of Memory" templateRef=""
type="Section"> - <ns12:links
xmlns:ns12="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model">
- <ns13:link author="LK1"
dateCreated="2004-10-13T16:38:53.132-07:00"
dateModified="2004-10-13T16:38:53.132-07:00" id="1017"
templateRef=""
xmlns:ns13="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns13:destination>
  <ns13:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0703.htm" targetKitID="Sample"
```

```
  useRange="false" />
  </ns13:destination>
  </ns13:link>
  </ns12:links>
- <ns14:link author="LK1"
dateCreated="2004-10-13T16:38:53.132-07:00"
dateModified="2004-10-13T16:38:53.132-07:00" id="1017"
templateRef=""
xmlns:ns14="http://www.sfu.ca/edu/gStudy/generated/xml/gstudy/model/link">
- <ns14:destination>
  <ns14:doc QTmedia="false" image="false" moClass="htmlDoc" rangeEnd="-1"
  rangeStart="-1" target="html/0703.htm" targetKitID="Sample"
  useRange="false" />
  </ns14:destination>
  </ns14:link>
  </ToCEntry>
```

Appendix C

Instantiated OWL File

The instantiated CILT ontology consists of data corresponding to learner interaction and the content. The fully instantiated CILT ontology is rather long. Hence, only an excerpt of the ontology is presented here.

Excerpt of Instantiated OWL file:

```
  <?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:tzont="http://www.isi.edu/~pan/damltime/timezone-ont.owl#"
xmlns:L="http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#"
xmlns:time-entry="http://www.isi.edu/~pan/damltime/time-entry.owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:PC="http://www.sfu.ca/~ldoherty/RA/Jurika/ProgrammingOntologyOwl.owl#"
xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
xmlns:I="http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#"
xmlns="http://www.sfu.ca/~shakya/ontology_lib/CILT.owl#"
xmlns:C="http://www.sfu.ca/~shakya/ontology_lib/Content.owl#"
xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xml:base="http://www.sfu.ca/~shakya/ontology_lib/CILT.owl">
 <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl"/>
   <owl:imports rdf:resource="http://www.isi.edu/~pan/damltime/time-entry.owl"/>
   <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
         3.2 Jurika Shakya</owl:versionInfo>
   <owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Content.owl"/>
   <owl:imports rdf:resource="http://www.sfu.ca/~shakya/ontology_lib/Learner.owl"/>
```

```
    <rdfs:comment rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">corrected the default namespace
added deleted by interactiosns.</rdfs:comment>
  </owl:Ontology>
  <rdf:Description rdf:about="http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <rdf:Description rdf:about=
            "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentFragments">
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:someValuesFrom rdf:resource=
                    "http://www.isi.edu/~pan/damltime/time-entry.owl#DurationDescription"/>
                <owl:onProperty rdf:resource=
                    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlightsAt"/>
              </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:someValuesFrom rdf:resource=
                    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"/>
                <owl:onProperty rdf:resource=
                    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlightedBy"/>
              </owl:Restriction>
            </rdfs:subClassOf>
          </rdf:Description>
        </owl:someValuesFrom>
        <owl:onProperty rdf:resource=
                    "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#highlights"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    ....
    <C:ProgrammingCode rdf:ID=
"LinkedLists0_ProgrammingLinkedLists0_ProgrammingExerciseTheNode0_Node_Unfinished.java_ProgramListing">
        <C:isPartOf rdf:resource=
"#LinkedLists0_ProgrammingLinkedLists0_ProgrammingExerciseTheNode0_Node_Unfinished.java0">
        </C:isPartOf>
    </C:ProgrammingCode>
    <C:ProgrammingCode rdf:ID=
"LinkedLists0_ProgrammingLinkedLists0_CreatingaGenericLinkedList0_GenericStack.java_ProgramListing">
        <C:isPartOf rdf:resource=
"#LinkedLists0_ProgrammingLinkedLists0_CreatingaGenericLinkedList0_GenericStack.java0">
        </C:isPartOf>
    </C:ProgrammingCode>
    <owl:Thing rdf:ID="PsuedoRandomProbing">
    </owl:Thing>
```

```
<C:Question rdf:ID="Arrays0_ReviewQuestions1_Question1">
    <C:isPartOfQuestionSet rdf:resource="#Arrays0_ReviewQuestions1_QuestionSet">
    </C:isPartOfQuestionSet>
    <C:isPartOf rdf:resource="#Arrays0_ReviewQuestions1">
    </C:isPartOf>
</C:Question>
<C:DocumentFragments rdf:ID="Hashing0_ReviewQuestions3">
    <C:hasTitle>Review Questions
    </C:hasTitle>
    <C:isPartOfDocumentElement rdf:resource="#Hashing0">
    </C:isPartOfDocumentElement>
    <C:hasDocumentType>section
    </C:hasDocumentType>
</C:DocumentFragments>
<C:DocumentFragments rdf:ID="Arrays0_Summary0">
    <C:isPartOfDocumentElement rdf:resource="#Arrays0">
    </C:isPartOfDocumentElement>
    <C:hasTitle>Summary
    </C:hasTitle>
    <C:hasDocumentType>section
    </C:hasDocumentType>
</C:DocumentFragments>
<C:Question rdf:ID="Hashing0_Quiz2_Question1">
    <C:isPartOfQuestionSet rdf:resource="#Hashing0_Quiz2_QuestionSet">
    </C:isPartOfQuestionSet>
    <C:isPartOf rdf:resource="#Hashing0_Quiz2">
    </C:isPartOf>
</C:Question>
<owl:Thing rdf:ID="FixedSizeArray">
</owl:Thing>
.....
<I:highlightsAt><time-entry:DurationDescription rdf:ID=
        "001027_C11_highlight_h207_duration">
    <time-entry:hours rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">10</time-entry:hours>
    <time-entry:minutes rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">06</time-entry:minutes>
    <time-entry:seconds rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">55</time-entry:seconds>
    <time-entry:years rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">2004</time-entry:years>
    <time-entry:months rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#string">11</time-entry:months>
    <time-entry:days rdf:datatype=
            "http://www.w3.org/2001/XMLSchema#string">17</time-entry:days>
    </time-entry:DurationDescription></I:highlightsAt></C:DocumentFragments>
<C:DocumentFragments rdf:ID=
```

```
"Hashing0_BuildingaBetterHashTablebyHandlingCollisions0_ReHashingAlgorithms0_PsuedoRandomProbing0">
        <C:hasDocumentType>media
        </C:hasDocumentType>
        <C:isPartOfDocumentElement rdf:resource=
            "#Hashing0_BuildingaBetterHashTablebyHandlingCollisions0_ReHashingAlgorithms0">
        </C:isPartOfDocumentElement>
        <C:hasTitle>Psuedo-Random Probing
        </C:hasTitle>
        <C:hasTopics rdf:resource=
            "http://www.sfu.ca/~ldoherty/RA/Jurika/ProgrammingOntologyOwl.owl#PsuedoRandomProbing">
        </C:hasTopics>
    </C:DocumentFragments>
    <owl:Thing rdf:ID="LinearProbing">
    </owl:Thing>
    <C:DocumentFragments rdf:ID="Hashing0_WhatisaHashTable0">
        <C:isPartOfDocumentElement rdf:resource="#Hashing0">
        </C:isPartOfDocumentElement>
        <C:hasTitle>What is a Hash Table?
        </C:hasTitle>
        <C:hasDocumentType>section
        </C:hasDocumentType>
    </C:DocumentFragments>
```

Appendix D

MI-EDNA Program Details

MI-EDNA uses a Java program to automatically instantiate the ontologies and to obtain JESS facts and JESS rules from the ontology. The Rete algorithm in JESS is called from within the Java program to facilitate a real-time environment for the execution of MI-EDNA. The conversion of the OWL files (ontology) into JESS facts is accomplished through XSL stylesheets[1]. Minor modifications were made to the XSL file to extract the rdf:ID into the facts.

Ontology Instantiator

Program D.1: Sample code from Ontology Instantiator

```
package ca.sfu.iat.edtech.ontoparser;

import ca.sfu.iat.edtech.helperClasses.xmlclasses.SAXHelper; import
ca.sfu.iat.edtech.helperClasses.xmlclasses.XMLTreeNode; import
ca.sfu.iat.edtech.helperClasses.TimeMarker; import
ca.sfu.iat.edtech.ontoparser.mapperObjects.NoteMapper; import
org.w3c.dom.Document; import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder; import
javax.xml.parsers.DocumentBuilderFactory; import
javax.xml.parsers.ParserConfigurationException; import
javax.xml.transform.*; import javax.xml.transform.dom.DOMSource;
```

[1]http://www.cs.cmu.edu/ sadeh/MyCampusMirror/OWLEngine.html

```
import javax.xml.transform.stream.StreamResult; import java.io.*;
import java.util.*; import java.util.logging.*;
 /**
 * Created by IntelliJ IDEA.
 * User: mayo
 * Date: May 16, 2005
 * Time: 10:15:18 PM
 */

public class OntoParser {
    //private String ontFile = null;
    private String logFileDirectory = null;
    private Document ontDocument = null;
    private File logFileList;
    private static Logger logger = Logger.getLogger
        (OntoParser.class.getPackage().getName());
    private static TimeMarker tm = new TimeMarker();

    /**
     * @param ontFile          ontology owl file
     * @param logFileDirectory directory where the logs are
     * @param logFileListFile  xml file with the logs that will be parsed
     */
    public OntoParser(String ontFile, String logFileDirectory,
        String logFileListFile) {
        logger.setLevel(Level.INFO);

        this.logFileList = new File(logFileListFile);
        this.logFileDirectory = logFileDirectory;

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        factory.setNamespaceAware(true);
        DocumentBuilder builder;

        try {
            File f = new File(ontFile);
            long size = f.length();
            System.out.println("Loading source ontology ...");
            System.out.print("file: " + ontFile + "\tsize: " + size + "\ttime: ");

            builder = factory.newDocumentBuilder();

            tm.mark();

            ontDocument = builder.parse(f);

            System.out.println(tm.status());
```

```
    } catch (ParserConfigurationException e) {
        e.printStackTrace();
    } catch (SAXException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}


public void traverse() throws IOException, SAXException,
    ParserConfigurationException {
    //LogContentHandler lch = new LogContentHandler();
    //SAXParserFactory spf = SAXParserFactory.newInstance();
    //XMLReader xmlReader = null;
    //SAXParser saxParser = spf.newSAXParser();
    //xmlReader = saxParser.getXMLReader();

    NoteMapper nm = new NoteMapper(ontDocument);
    File logFileDir = new File(logFileDirectory);

    readAllFiles(logFileDir, nm);
    logger.info(nm.totalLinks + " valid links made");
    logger.info(nm.modelObjects.size() + " model objects made");
    logger.info("c:" + c);

    //xmlReader.setContentHandler(nm);
    //xmlReader.setErrorHandler(SAXHandler);
    //xmlReader.parse(logFile);
}

int c = 0;

/**
 * read the xml file and convert the indexes into global indexes from their clusters
 * @param dir
 * @param to
 * @throws IOException
 * @throws SAXException
 */
private void readAllFiles(File dir, NoteMapper to) throws IOException, SAXException {
    InputStream is = new FileInputStream(logFileList);
    XMLTreeNode fileListRoot = SAXHelper.parseElements(is, null);
    Map filenamesToIndexes = new HashMap();
    Set clusters = fileListRoot.getAllNodesOfType("Cluster");
    int offset = 0;
```

```
for (Iterator i = clusters.iterator(); i.hasNext();) {
    int offsetAmount = 0;
    XMLTreeNode cluster = (XMLTreeNode) i.next();
    Set files = cluster.getAllChildNodes();

    for (Iterator j = files.iterator(); j.hasNext();) {
        XMLTreeNode file = (XMLTreeNode) j.next();
        String name = file.getAttribute("name");
        int index = Integer.parseInt(file.getAttribute("index"));

        filenamesToIndexes.put(name, new Integer(index + offset));

        ++offsetAmount;
    }
    offset += offsetAmount;
}

is.close();

//get the actual files
List files = new ArrayList();
recursiveRead(dir, files);
//System.out.println("read " + files.size() + " log files");
//read the files in order of their index in the loglist xml file we read earlier
File[] orderedFiles = new File[files.size()];
for (int i = 0; i < files.size(); ++i) {
    Integer index = (Integer) filenamesToIndexes.get(((File) files.get(i)).getName());
    if (index != null) { //ignore files that don't exist
        orderedFiles[index.intValue()] = (File) files.get(i);
    }
}

System.out.println("Processing log files ...");

for (int i = 0; i < orderedFiles.length; ++i) {
    String name = null;

    //VERY VERY ugly workaround :) This should be rewritten to something mose sensible
    try {
        name = orderedFiles[i].toString();
    } catch (NullPointerException e) {
    }

    if (name != null) {
        File f = new File(name);
        long size = f.length();
```

```
            tm.mark();

            to.fromXML(name);

            long time = tm.status();
            System.out.println("file: " + name + "\tsize: " + size + "\ttime: " + time);
        }
    }
}

private void recursiveRead(File dir, List to) {
    File[] logFiles = dir.listFiles();
    for (int i = 0; i < logFiles.length; ++i) {
        if (logFiles[i].isFile()) {
            to.add(logFiles[i]);
            ++c;
        } else if (logFiles[i].isDirectory()) {
            recursiveRead(logFiles[i], to);
        }
    }
}

public String documentToString() throws TransformerException {
    TransformerFactory factory = TransformerFactory.newInstance();
    Transformer transform = factory.newTransformer();
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    StreamResult output = new StreamResult(baos);
    DOMSource domDoc = new DOMSource(ontDocument);

    transform.transform(domDoc, output);

    return baos.toString();
}


public void documentToFile(String filename) throws TransformerException {
    Source source = new DOMSource(ontDocument);
    File file = new File(filename);
    Result result = new StreamResult(file);

    Transformer xformer = TransformerFactory.newInstance().newTransformer();
    xformer.transform(source, result);
}


public static void main(String[] args) {
```

```
try {
    FileHandler fh = new FileHandler("log.txt");
    fh.setFormatter(new SimpleFormatter());

    //silence the logging into stdOut;
    Handler[] handlers = Logger.getLogger("").getHandlers();
    for (int i = 0; i < handlers.length; i++) {
        Logger.getLogger("").removeHandler(handlers[i]);
    }

    Logger.getLogger("").addHandler(fh);

    //logger.addHandler(fh);
} catch (IOException e) {
    e.printStackTrace();
}

String ontFile = null;
String logDir = null;
String outFile = null;
String logFileList = null;

if (args.length >= 4) {
    ontFile = args[0];
    logDir = args[1];
    outFile = args[2];
    logFileList = args[3];

} else {
    System.out.println("Usage: ontoparser ontologySourceFile " +
            "logDirectory ontologyOutputFile logFileList.\nlogFileList is " +
            "usually loglist.xml");
    System.exit(0);
}

OntoParser op = new OntoParser(ontFile, logDir, logFileList);

try {
    op.traverse();
} catch (SAXException ex) {
    ex.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (ParserConfigurationException e) {
    e.printStackTrace();
}
```

```
        long time = 0;

        try {
            tm.mark();

            op.documentToFile(outFile);

            time = tm.status();
        } catch (TransformerException e) {
            e.printStackTrace();
        }

        File o = new File(outFile);
        long size = o.length();

        System.out.println("Writing instantiated ontology ...");
        System.out.println("file: " + outFile + "\tsize: " + size + "\ttime: " + time);

        //ps.close();
    }
}
----------------------------------------------------------------------------
package ca.sfu.iat.edtech.ontoparser;

import java.util.logging.Logger;

public class SaxMapperLog {
    private static Logger logger = Logger.getLogger("ca.sfu.iat.edtech.ontoparser");

    static boolean doTraceLogging =
        Boolean.getBoolean("ca.sfu.iat.edtech.ontoparser.saxMapper.trace");

    public static void trace(String msg) {
        if (doTraceLogging) {
            logger.info(msg);
        }
    }


    public static void error(String msg) {
        logger.warning(msg);
    }
}

----------------------------------------------------------------------------
package ca.sfu.iat.edtech.ontoparser.util;
```

```
public class OntoDate {
    private String year;
    private String month;
    private String day;
    private String hour;
    private String minute;
    private String second;
    private String milisecond;
    private String timezoneOffset;

    //date pattern: 2004-11-09T08:27:35.513-08:00
    public static final OntoDate parse(String dateString) {
        OntoDate date = new OntoDate();

        String year = dateString.substring(0, 4);
        String month = dateString.substring(5, 7);
        String day = dateString.substring(8, 10);
        String hour = dateString.substring(11, 13);
        String minute = dateString.substring(14, 16);
        String second = dateString.substring(17, 19);
        String milisecond = dateString.substring(20, 23);
        //String timezoneOffset = dateString.substring(23, 29);
        //date.set(year, month, day, hour, minute, second, milisecond, timezoneOffset);
        date.set(year, month, day, hour, minute, second, milisecond);
        return date;
    }

    public void set(String year, String month, String day,
            String hour, String minute, String second, String milisecond)
    {
        this.year = year;
        this.month = month;
        this.day = day;
        this.hour = hour;
        this.minute = minute;
        this.second = second;
        this.milisecond = milisecond;
        this.timezoneOffset = timezoneOffset;
    }

    public String toString() {
        return (year + "-" + month + "-" + day + "T" + hour + ":" + minute
                       + ":" + second + "." + milisecond + timezoneOffset);
    }

    public String getYear() {
        return year;
```

```
}

public void setYear(String year) {
    this.year = year;
}

public String getMonth() {
    return month;
}

public void setMonth(String month) {
    this.month = month;
}

public String getDay() {
    return day;
}

public void setDay(String day) {
    this.day = day;
}

public String getHour() {
    return hour;
}

public void setHour(String hour) {
    this.hour = hour;
}

public String getMinute() {
    return minute;
}

public void setMinute(String minute) {
    this.minute = minute;
}

public String getSecond() {
    return second;
}

public void setSecond(String second) {
    this.second = second;
}

public String getMilisecond() {
```

```
        return milisecond;
    }

    public void setMilisecond(String milisecond) {
        this.milisecond = milisecond;
    }

    public String getTimezoneOffset() {
        return timezoneOffset;
    }

    public void setTimezoneOffset(String timezoneOffset) {
        this.timezoneOffset = timezoneOffset;
    }
}

---------------------------------------------------------------------------
package ca.sfu.iat.edtech.ontoparser.util;

public class Namespaces {
    public static final String NS_Content = "C";
    public static final String NS_Learner = "L";
    public static final String NS_time = "time-entry";
    public static final String NS_timezone = "tzont";
    public static final String NS_Interaction = "I";
}

---------------------------------------------------------------------------
package ca.sfu.iat.edtech.ontoparser.util;

public class Notes {
    public static final String TYPE_QuickNote = "QuickNote";
}

---------------------------------------------------------------------------
package ca.sfu.iat.edtech.ontoparser.util;

import org.w3c.dom.*;

public class DocumentHelper {
    public static Node createSimpleTextNode(Document doc, String nodeName, String nodeValue) {
        return createSimpleNodeWithType(doc, nodeName, nodeValue,
                    "http://www.w3.org/2001/XMLSchema#string");
    }

    public static Node createSimpleNodeWithType(Document doc, String nodeName,
                    String nodeValue, String type) {
```

```
        Element node;
        node = doc.createElement(nodeName);
        node.setAttribute("rdf:datatype", type);
        Text node_text = doc.createTextNode(nodeValue);
        node.appendChild(node_text);
        return node;
    }

    public static Node createSimpleTextNodeWithAttrs(Document doc, String nodeName,
                    String nodeValue, Attr attrs) {
        Element e = (Element) createSimpleTextNode(doc, nodeName, nodeValue);
        e.setAttributeNode(attrs);
        return e;
    }

    public static String createTagWithNS(String ns, String tagName) {
        return ns + ":" + tagName;
    }

    public static Node createCalendarClockDescription(Document doc, String id, OntoDate date) {
        Element calendarClockDescription =
                doc.createElement(createTagWithNS(Namespaces.NS_time, "DurationDescription"));
        calendarClockDescription.setAttribute("rdf:ID", id);
        Node hour =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "hours"), date.getHour());
        calendarClockDescription.appendChild(hour);
        Node minute =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "minutes"), date.getMinute());
        calendarClockDescription.appendChild(minute);
        Node second =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "seconds"), date.getSecond());
        calendarClockDescription.appendChild(second);
        Node year =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "years"), date.getYear());
        calendarClockDescription.appendChild(year);
        Node month =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "months"), date.getMonth());
        calendarClockDescription.appendChild(month);
        Node day =
        createSimpleTextNode(doc, createTagWithNS(Namespaces.NS_time, "days"), date.getDay());
        calendarClockDescription.appendChild(day);
        return calendarClockDescription;
    }
}
```

ontoTranform2Jess

Program D.2: Sample code from Ontology to Jess Transformer

```
import javax.xml.transform.TransformerFactory; import
javax.xml.transform.Transformer; import
javax.xml.transform.TransformerException; import
javax.xml.transform.stream.StreamSource; import
javax.xml.transform.stream.StreamResult; import
java.io.FileOutputStream; import java.io.FileNotFoundException;

/**
 * Created by IntelliJ IDEA.
 * User: mayo
 * Date: Oct 25, 2005
 * Time: 12:25:36 PM
 */
public class Transform {

    public Transform() {
    }

    public void transform(String stylesheet, String ontology, String output)
                throws TransformerException, FileNotFoundException {
        TransformerFactory tFactory = TransformerFactory.newInstance();
        Transformer transformer = tFactory.newTransformer(new StreamSource(stylesheet));
        transformer.transform(new StreamSource(ontology),
                new StreamResult(new FileOutputStream(output)));
    }

    public static void main(String[] args) {
        if (args.length != 3) {
            System.out.println("Usage: Transform xsl_stylesheet instantiated_owl output_file");
            System.exit(1);
        }

        try {
            new Transform().transform(args[0], args[1], args[2]);
        } catch (TransformerException e) {
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```
--------------------------------------------------------------------------------
import jess.*; import org.jivesoftware.smack.XMPPConnection; import
org.jivesoftware.smack.XMPPException; import
org.jivesoftware.smack.Chat;

import java.util.Hashtable;

public class SendMessage implements Userfunction {
    private Messenger test;
    public SendMessage(Messenger test) {
        this.test = test;
    }

    public String getName() {
        return "miedna-send-message";
    }

    public Value call(ValueVector valueVector, Context context) throws JessException {
        String learner = valueVector.get(1).stringValue(context);
        String message = "";
        int valVSize = valueVector.size();
        for (int i = 2; i < valVSize; i++) {
            message += valueVector.get(i).stringValue(context) + " ";
        }

        try {
            test.sendMessage(learner, message);
        } catch (XMPPException e) {
            e.printStackTrace();
        }

        return null;
    }
}

--------------------------------------------------------------------------------
import java.io.Reader; import java.io.Writer;

public class ReaderWriterThread extends Thread {
    private Reader reader;
    private Writer writer;

    public ReaderWriterThread() {
    }

    public void run() {
```

```
        while (true) {
        }
    }
}

----------------------------------------------------------------------------
import org.jivesoftware.smack.packet.Message;

import java.util.EventListener;

public interface MessageListener extends EventListener {
    void messageReceived(Message message);
}

----------------------------------------------------------------------------
import org.jivesoftware.smack.*; import
org.jivesoftware.smack.filter.PacketFilter; import
org.jivesoftware.smack.filter.PacketTypeFilter; import
org.jivesoftware.smack.packet.Message; import
org.jivesoftware.smack.packet.Presence; import
org.jivesoftware.smack.packet.IQ; import
org.jivesoftware.smack.packet.Packet; import
org.jivesoftware.smackx.MessageEventManager; import
org.jivesoftware.smackx.DefaultMessageEventRequestListener; import
org.jivesoftware.smackx.MessageEventNotificationListener;

import java.util.*; import java.io.IOException;

public class Messenger implements Runnable {
    private static Messenger instance = null;
    private XMPPConnection connection;
    private static String server = "209.87.56.80";
    private static int port = 5222;
    private static String username = "miedna";
    private static String password = "miednapwd";
    private static String resource = "miedna";
    private static int priority = 5;
    private Hashtable openclients;
    private Thread thread;
    private Vector messageListeners;

    public Messenger() {
        messageListeners = new Vector();
        openclients = new Hashtable();

        try {
            Properties props = new Properties();
```

```
            props.load(this.getClass().getResourceAsStream("miedna.properties"));
            server = props.getProperty("server");
            port = Integer.parseInt(props.getProperty("port"));
            username = props.getProperty("username");
            password = props.getProperty("password");
            resource = props.getProperty("resource");
            priority = Integer.parseInt(props.getProperty("priority"));
        } catch (IOException e) {
            System.out.println("Didn't find a property file. Using defaults.");
        }
    }

    public Messenger getInstance() {
        if (instance == null) {
            instance = new Messenger();
        }
        return instance;
    }

    private void connect() throws XMPPException {
        connection = new XMPPConnection(server, port);
        connection.login(username, password, resource);
    }

    private void disconnect() {
        connection.close();
    }

    public void sendMessage(String learner, String message) throws XMPPException {
        Chat chat = (Chat) openclients.get(learner);
        if (chat == null) {
            chat = connection.createChat(learner);
            openclients.put(learner, chat);
            System.out.println("create chat with " + learner);
        }
        chat.sendMessage(message);
    }

    public void start() {
        thread = new Thread(this);
        thread.start();
    }

    public void stop() {
        thread = null;
        disconnect();
    }
```

```
public void run() {
    try {
        connect();
    } catch (XMPPException e) {
        e.printStackTrace();
    }

    if (connection == null || !connection.isConnected()) {
        System.out.println("Could not connect to the server ...");
        return;
    }

    //setup listener stuff
    PacketListener messageListener = new PacketListener() {
        public void processPacket(Packet packet) {
            Message message = (Message) packet;
            if (message.getType() != Message.Type.CHAT
                    && message.getType() != Message.Type.HEADLINE
                    && message.getType() != Message.Type.NORMAL) {
                return;
            }

            if (message.getType() == Message.Type.NORMAL) {
                if (message.getBody() == null) {
                    return;
                }
            }
            fireMessageListener(message);
        }
    };
    PacketFilter messageFilter = new PacketTypeFilter(Message.class);
    connection.addPacketListener(messageListener, messageFilter);
    connection.getRoster().setSubscriptionMode(Roster.SUBSCRIPTION_ACCEPT_ALL);
    connection.sendPacket(new Presence(Presence.Type.AVAILABLE,
            "Online", priority, Presence.Mode.AVAILABLE));
}

public void addMessageListener(MessageListener ml) {
    messageListeners.add(ml);
}

public void removeMessageListener(MessageListener ml) {
    messageListeners.remove(ml);
}

void fireMessageListener(final Message message) {
```

```
        new Thread() {
            public void run() {
                Iterator i = messageListeners.iterator();
                while (i.hasNext()) {
                    ((MessageListener) i.next()).messageReceived(message);
                }
            }
        }.start();
    }
}
```

Appendix E

Rules for Mixed-Initiative Recognition

The essence of MI-EDNA lies in its ability to recognizing system-initiation points/opportunities for interactions. These initiation points are recognized by the JESS rules embedded within MI-EDNA. These rules are fired based on the interaction data of the learners captured in the ontology and the SRL principles that were captured in the ontology. This appendix lists a few of these rules.

OWL Template:

MI-EDNA uses the OWL meta model template developed by the Carnegie Mellon University. This meta-model captures the OWL structures and the owl axioms in the form of triplets.

MI-EDNA specific Template:

Program E.1: Sample Jess MI-EDNA `templates`

```
;;;Gobal Variables----------------------
(defglobal ?*gcounter* = 0)

;;;; Template ----------------------------
(deftemplate tmp_lt_cnt
```

```
   (slot LearningTask (default ""))(slot Learner  (default ""))(slot LTCount  (default 0)))
(defquery search-template-tmp_lt_cnt
  (declare (variables ?lt) )
     (tmp_lt_cnt ( LearningTask ?lt)( Learner ?l)( LTCount ?cnt)))

;;;------------------------------------------
(deftemplate tmp_cnt
   (slot LearningTask (default ""))(slot Learner  (default ""))(slot Template  (default "")))
(defquery search-template-tmp_cnt
  (declare (variables ?lt) )
     (tmp_cnt ( LearningTask ?lt) ( Learner ?l)( Template ?t)))

;;----------------------------------------------

(deftemplate tmp_LT_ref
   (slot LearningTask (default ""))(slot Learner  (default ""))(slot Template  (default "")))
(defquery search-template-tmp_LT_ref
  (declare (variables ?lt1) )
     (tmp_LT_ref ( LearningTask ?lt1)( Learner ?l1)( Template ?t1)))

;;; Template with time----------------------
(deftemplate tmp_time
   (slot Learner  (default ""))(slot Template  (default ""))(slot Hr (default 0))
   (slot Min (default 0))(slot Sec (default 0))(slot Yr (default 0))
   (slot Mnth (default 0))(slot Day (default 0)) )
(defquery search-template-tmp_time
  (declare (variables ?l) )
 (tmp_time (Learner ?l)(Template  ?t)(Hr ?h)(Min ?m)(Sec ?s)(Yr ?y)(Mnth ?mn)(Day ?d)))

;;;; Template ----------------------------
(deftemplate tmp_TT
   (slot Tactics  (default ""))(slot Learner  (default "")) )
(defquery search-template-tmp_TT
  (declare (variables ?l) )
     (tmp_TT (Tactics ?t)(Learner ?l)))

;;;; Template ----------------------------
(deftemplate tmp_TS
   (slot Strategies (default ""))(slot Learner  (default ""))(multislot Tactics))
(defquery search-template-tmp_TS
  (declare (variables ?l) )
     (tmp_TS(Strategies ?s)(Learner ?l)(Tactics $?t)))

;;;; Template ----------------------------
(deftemplate tmp_link
   (slot Topic  (default ""))(slot DocCmt  (default ""))(slot DocFrgmnt  (default ""))
       (slot Template (default "")))
```

```
(defquery search-template-tmp_link
  (declare (variables ?tm) )
 (tmp_link (Topic  ?tp) (DocCmt  ?dc)(DocFrgmnt  ?df)(Template ?tm)))

;;; Template ----------------------------

(deftemplate tmp_type_noteStyle
   (slot LearningTask  (default ""))(slot Notetyp  (default "")))
(defquery search-template-tmp_type_noteStyle
  (declare (variables ?lt) )
 (tmp_type_noteStyle(LearningTask ?lt)(Notetyp ?nt)))
```

Functions:

Program E.2: Sample Jess Rules and Functions in MI-EDNA

```
;;Initiatilizer--------------------------

(defrule Initlize (declare (salience 100))
 (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject   ?Learner)(object    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
=>
    (foreach ?item
    (create$ "gCC" "gCCNL" "gCL" "gCN" "gCQLA" "gCQLABEL" "gCQLC" "gCQLCE" "gCQLCS" "gCQLD" "gCQLDA"
    "gCQLDU" "gCQLE" "gCQLI" "gCQLM" "gCQLMEM" "gCQLNR" "gCQLP" "gCQLR" "gCQLRR" "gCQLS" "gCQLTB"
    "gCQLTH" "gCQNL" "gCSN" "gUC" "gUCN" "gUDN" "gUQC" "gUQDU" "gUQE" "gUQI" "gUQR"
    "gUQRR" "gUQS" "gUSN")
        (assert (tmp_lt_cnt (LearningTask ?item)(Learner  ?Learner)(LTCount 0))))
        (foreach ?TSitem (create$ "CriticalThinking_TS" "Elaboration_TS" "Organization_TS"
        "Rehearsal_TS")
            (assert (tmp_TS(Strategies ?TSitem) (Learner  ?Learner) (Tactics "")))))

;; InformLearnerStrategies ----------------------------

(defrule InformLeanerStrategies (declare (salience 100))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject   ?Learner)(object    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
 (tmp_TT(Tactics ?Tactic)(Learner ?L&:(eq ?L ?Learner)))
=>
(miedna_informs (str-cat "You have been only using Tactics "
?Tactic))
```

```
;;; rules for Linker ----------------------------

(defrule Linker (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#links")
  (subject   ?Link1)
  (object    ?DocumentFragment1))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#links")
  (subject   ?L2&:(eq ?L2 ?Link1))
  (object    ?DocumentComment1))
=>
(assert (triple  (predicate Linked)
  (subject  ?DocumentFragment1) (object    ?DocumentComment1))))

;;; QuickNoter ----------------------------

(defrule QuickNoter (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#createdBy")
  (subject   ?DocumentComment1) (object   ?Learner))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasEmbeddedElements")
  (subject   ?DocumentComment2) (object    ?Element1))
 (test (eq ?DocumentComment1 ?DocumentComment2))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasTemplate")
  (subject   ?Element2) (object    ?Template1))
  (test (eq ?Element1 ?Element2))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#templateRef")
  (subject   ?Template2) (object    "QuickNote"))
(test (eq ?Template1 ?Template2))
(triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#templateVersion")
  (subject   ?T&:(eq ?T ?Template1))(object    "0"))
=>
(assert (triple (predicate ?Template1)
    (subject   ?Learner)(object    "QuickNote"))))

;;; QuickNoterUpdater----------------------------

(defrule QuickNoteUpdater (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#updatedBy")
  (subject   ?DocumentComment1) (object   ?Learner))
```

```
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasEmbeddedElements")
  (subject   ?DocumentComment2) (object    ?Element1))
 (test (eq ?DocumentComment1 ?DocumentComment2))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasTemplate")
  (subject   ?Element2)(object    ?Template1))
  (test (eq ?Element1 ?Element2))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#templateRef")
  (subject   ?Template2)(object    "QuickNote"))
(test (eq ?Template1 ?Template2)) (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#templateVersion")
  (subject   ?T&:(eq ?T ?Template1))(object    ?v&:(not (eq ?v "0"))) )
=>
(assert(triple(predicate ?Template1)
   (subject   "QuickNote")(object    ?Learner))))

;;; TopicMapper ----------------------------

(defrule TopicMapper (declare  (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasTemplate")
  (subject   ?Element)(object    ?Template))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasEmbeddedElements")
  (subject   ?DocumentComment)(object    ?E&:(eq ?E ?Element)))
  (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#links")
  (subject   ?Link)(object    ?E2&:(eq ?E2 ?Element)))
  (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#links")
  (subject   ?L&:(eq ?L ?Link))(object    ?DocumentFragment))
   (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject   ?DF&:(eq ?DF ?DocumentFragment))
    (object "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#DocumentFragments"))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject  ?DFName)(object    ?DF1&:(eq ?DF1 ?DocumentFragment)))
  (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasTitle")
  (subject   ?DF1&:(eq ?DF1 ?DocumentFragment))(object    ?topic))
=>
(assert(tmp_link (Topic  ?topic) (DocCmt  ?DocumentComment)
   (DocFrgmnt  ?DFName)(Template ?Template))))
```

```
;; InformerLearnerTaskcount ---------------------------

(defrule InformLeanerTaskCount (declare (salience 100))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject?Learner)(object "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
 ?fid <- (tmp_cnt( LearningTask ?Type)( Learner ?Learner1)( Template ?t))
  (test (eq ?Learner ?Learner1))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject   ?LearnerName)(object    ?L&:(eq ?L ?Learner)))
=>
    (if  (eq ?*gcounter* 0)
    then (bind ?*gcounter*  (learner_task_cnt ?LearnerName ?Type ?fid))
    else (bind  ?*gcounter*  (learner_task_retractor  ?fid) ) ))

;;; rules for learner_task_count ---------------------------

(deffunction learner_task_cnt (?learner ?type ?fid)
        (bind ?result_num (count-query-results search-template-tmp_cnt ?type))
        (miedna_informs (str-cat ?learner " created " ?type " notes " ?result_num " times"))
        (retract ?fid)
    (return 1))

(deffunction learner_task_retractor (?fid ?type)
    (retract ?fid)
    (return 1))
```

Rules - Learning Tasks:

Program E.3: Sample Jess `Rules` for Learning Tasks in MI-EDNA

```
;;; rules for gCC ---------------------------

(defrule Learner_CC (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Interaction.owl#createdBy")
  (subject   ?DocumentComment1)(object ?Learner))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasEmbeddedElements")
  (subject   ?DocumentComment2&:(eq ?DocumentComment2 ?DocumentComment1))
  (object    ?Element1))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasGlossaryTemplate")
```

```
  (subject   ?Element2&:(eq ?Element2 ?Element1)) (object ?Template1))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#glossaryElement")
  (subject   ?Template2&:(eq ?Template2 ?Template1))(object "concept"))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasGlossaryTVersion")
  (subject   ?T&:(eq ?T ?Template1))(object    "0"))
=>
(assert (tmp_cnt
   (LearningTask "gCC") (Learner ?Learner)(Template  ?Template1))
(assert (tmp_LT_ref
   (LearningTask "gCC")(Learner ?Learner)(Template  ?Template1)))
(assert
 (triple(predicate "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#IdentifyingNewConceptsTT")
  (subject  "gCC")(object   ?Learner)))
(assert
 (triple (predicate "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#NamingItemsToLearnTT")
  (subject  "gCC")(object   ?Learner)))
(printout t ?Learner "Took Concept" crlf) )

;;; rules for gCQLDA ----------------------------

(defrule Learner_CQLDA (declare (salience 100))
  (triple
    (predicate ?Template1)(subject ?Learner)(object "QuickNote"))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/Content.owl#hasValue")
  (subject   ?Template2)(object "I disagree"))
  (test (eq ?Template1 ?Template2))
=>
(assert (tmp_cnt
   (LearningTask "gCQLDA")(Learner ?Learner)(Template ?Template1)))
(assert (tmp_LT_ref
   (LearningTask "gCQLDA")(Learner ?Learner)(Template  ?Template1)))
   (printout t ?Learner "Took QuickNote I disagree" crlf) )

;;; EOF
```

Rules - Teaching Tactics and Strategies:

Program E.4: Sample Jess `Rules` for Teaching Tactics and Strategies in MI-EDNA

```
;;; rules for ParaphrasingTT ----------------------------
```

```
(defrule Learner_ParaphrasingTT (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#ParaphrasingTT")
  (subject  ?LearningTask)(object   ?Learner))
 (test (or (eq ?LearningTask "gCCNL") (eq ?LearningTask "gCSN")))
 (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject   ?LearnerName)(object    ?L&:(eq ?L ?Learner)))
=>
(assert
 (tmp_TT
   (Tactics "ParaphrasingTT")(Learner ?Learner)))
(miedna_informs (str-cat ?LearnerName "practiced Tactics -
Paraphrasing ")))

;;; rules for CreatingAnalogiesTT ----------------------------
(defrule Learner_CreatingAnalogiesTT (declare (salience 100))
 (triple
  (predicate "http://www.sfu.ca/~shakya/ontology_lib/TTS.owl#CreatingAnalogiesTT")
  (subject  ?LearningTask)(object   ?Learner))
 (test (or (eq ?LearningTask "gCQLCS")(eq ?LearningTask "gCQLE")(eq ?LearningTask "gCQLMEM")))
 (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject   ?LearnerName) (object    ?L&:(eq ?L ?Learner)))
=>
(assert
 (tmp_TT
   (Tactics "CreatingAnalogiesTT")(Learner ?Learner)))
(miedna_informs (str-cat ?LearnerName "practiced Tactics -  Creating
Analogies ")))

;;; rules for Elaboration_TS ----------------------------

(defrule Learner_Elaboration_TS (declare (salience 100))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject   ?Learner)
    (object    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
  ?fid <-  (tmp_TS
   (Strategies "Elaboration_TS")(Learner ?L&:(eq ?L ?Learner))(Tactics ?Tactic))
  (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
    (subject   ?LearnerName) (object    ?L&:(eq ?L ?Learner)))
=>
 (bind ?LT (run-query* search-template-tmp_TT ?Learner))
 (bind ?tempT "")
 (while (?LT next)
```

```
   (foreach ?item (create$ "CreatingAnalogiesTT" "ParaphrasingTT" "SummarizingTT")
    (if   (eq ?item (?LT getString t))
       then (bind ?tempT (str-cat ?tempT " "?item)))))
 (modify ?fid (Tactics ?tempT)) (miedna_informs (str-cat
?LearnerName "  is using the Critical Thinking Strategies - "
?tempT)) )

;;; rules for CriticalThinking_TS ---------------------------
(defrule Learner_CriticalThinking_TS (declare (salience 100))
  (triple
   (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
   (subject   ?Learner)
   (object    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
?fid <-  (tmp_TS
   (Strategies "CriticalThinking_TS")(Learner ?L&:(eq ?L ?Learner))(Tactics ?Tactic))
  (triple
   (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
   (subject   ?LearnerName)(object    ?L&:(eq ?L ?Learner)))
=>
 (bind ?LT (run-query* search-template-tmp_TT ?Learner))
 (bind ?tempT "")
 (while (?LT next)
   (foreach ?item (create$ "IdentifyNewConceptsTT" "MakingDecisionTT"
       "EvaluationTT" "QuestionsTT")
    (if   (eq ?item (?LT getString t))
       then (bind ?tempT (str-cat ?tempT " "?item)) ) ))
 (modify ?fid (Tactics ?tempT)) (miedna_informs (str-cat
?LearnerName "  is using the Critical Thinking Strategies - "
?tempT)) )
```

Rules - Self Regulated Learning Phases:

Program E.5: Sample Jess Rules for SRL phases recognition in MI-EDNA

```
;; rules for SRLPerformancePhase ---------------------------
(defrule Learner_SRLPerformancePhase (declare (salience 100))
 (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (subject   ?Learner)
    (object    "http://www.sfu.ca/~shakya/ontology_lib/Learner.owl#LearnerLK"))
   ?fid <-  (tmp_TS
   (Strategies ?Strategy)(Learner ?L&:(eq ?L ?Learner))(Tactics ?Tactic))
   (triple
    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#rdfID")
```

```
    (subject   ?LearnerName)(object    ?L1&:(eq ?L1 ?Learner)))
=>
 (bind ?LS (run-query* search-template-tmp_TS ?Learner))
 (bind ?tempS "")
 (while (?LS next)
    (foreach ?item  (create$ "Organization_TS" "Elaboration_TS" "CriticalThinking_TS")
         (if   (eq ?item (?LS getString s))
            then(bind ?tempS (str-cat ?tempS " "?item)))))
 (miedna_informs (str-cat ?LearnerName"  is using these Strategies
- " ?tempS " during the SRL Performance Phase")))
```

Bibliography

[1] J. F. Allen. Mixed-initiative interaction. *IEEE Intelligent Systems*, pages 14 – 16, September 1999.

[2] C. Andersen, D. Traum, K. Purang, and D. Purushothaman. Mixed initiative dialogue and intelligence via active logic. In *Working Notes of the AAAI-99 Workshop on Mixed-initiative Intelligence*, pages 99 – 106, Orlando, Florida, 18 July 1999. AAAI.

[3] G. Antoniou and F. V. Harmelen. Web ontology language: Owl. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*. Springer-Verlag, 2003.

[4] L. Aroyo, D. Dicheva, and A. Cristea. Ontological support for web courseware authoring. In *ITS02 Intelligent Tutoring Systems*, volume LNCS 2363, pages 270 – 280. Springer, 2002.

[5] R. Azevedo and A. Hadwin. *Instructional Science*, chapter Scaffolding self-regulated learning and metacognition Implications for the design of computer-based scaffolds. Springer, 2005.

[6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[7] M. Boicu, G. Tecuci, D. Marcu, C. Boicu, and B Stanescu. Mixed-initiative control for teaching and learning in disciple. In *In Proceedings of the IJCAI-03 Workshop on Mixed-Initiative Intelligent Systems*, Acapulco, Mexico,, August 2003.

[8] A. Borgida and P.F. Patel-Schneider. A semantics and complete algorithm for subsumption in the classic description logic. In *Journal of Artificial Intelligence Research*, volume 1, pages 277 – 308. AI Access Foundation and Morgan Kaufmann Publishers, 1994.

[9] J. Bourdeau and R. Mizoguchi. Ontological engineering of instruction: Aperspective. In *Proceedings of AIED'99*, pages 620 – 623, 1999.

[10] R. J. Brachman and J. G. Schmolze. An overview of the kl-one knowledge representation system. In *Cognitive Science*, volume 9, pages 171 – 216. Ablex Publ Corp, Schlumberger Palo Alto Res, Palo Alto, CA USA, 1985.

[11] D. Brokenshire, J. Shakya, and V. Kumar. Providing information for mixed initiative interaction via interoperable user modelling. In *AAAI Fall Symposia on Mixed-Initiative Problem-Solving Assistants*, pages 16 – 21, 2005.

[12] M. Burstein, G. Ferguson, and J. Allen. Integrating agent-based mixed-initiative control with an existing multi-agent planning system. Technical Report 72, Computer Science Dept., University of Rochester, 2000.

[13] P. R. Cohen. The pragmatics of referring and the modality of communication. *Computational Linguistics*, 10:97 – 146, 1984.

[14] R. Cohen, C. Allaby, C. Cumbaa, M. Fitzgerald, K. Ho, B. Hui, C. Latulipe, F. Lu, N. Moussa, D. Pooley, A. Qian, and S. Siddiqi. *User Modeling and User-Adapted Interaction*, chapter What is Initiative?, pages 171 – 214. Kluwer Academic Publishers., 1998.

[15] J. Davies, R. Studer, Y. Sure, and P. W. Warren. *Technology Journal*, volume 23, chapter Next generation knowledge management, pages 175 – 190. July 2005.

[16] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI Magazine*, Spring:17 – 33, Spring 1993.

[17] D. Dicheva, L. Aroyo, and A. Cristea. Cooperative courseware-authoring support. *International Journal of Computers and Applications*, 25(3):179 – 187, 2003.

[18] L. Doherty, J. Shakya, M. Jordanov, P. Lougheed, D. Brokenshire, S. Rao, and V.S. Kumar. Recognizing opportunities for mixed-initiative interactions in novice programming. In *AAAI Fall Symposium on Mixed-Initiative Problem-Solving Assistants*, pages 51 – 56, 2005.

[19] L. Doherty, M. Whitney, J. Shakya, M. Jordanov, P. Lougheed, D. Brokenshire, S. Rao, S. Menon, and V.S. Kumar. Mixed-initiative in computer games: Algorithmic content creation in open-ended worlds. In *AAAI Fall Symposium on Mixed-Initiative Problem-Solving Assistants*, pages 46 – 56, 2005.

[20] B. Du Boulay and R. Luckin. Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education*, 12:235 – 256, 2001.

[21] I. Horrocks F. Baader and U. Sattler. Description logics as ontology languages for the semantic web. In D. Hutter and W. Stephan, editors, *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2003.

[22] G. Ferguson, J. Allen, and B. Miller. Trains-95: Towards a mixed-initiative planning assistant. In *Proceedings of the Third Conference on Artificial Intelligence Planning Systems (AIPS-96)*, pages 70 – 77, Edinburgh, Scotland, 1996.

[23] C. Forgy. Ops5: Rete-based expert system shell. Technical report, 2005.

[24] C. L. Forgy. Rete: a fast algorithm for the many pattern/many object pattern match problem. pages 324 – 341, 1990.

[25] E. Friedman-Hill. *Jess in Action.* Manning Publications, 2003.

[26] A.C. Graesser, N. Person, D. Harter, and TRG. Teaching tactics and dialog in autotutor. In *International Journal of Artificial Intelligence in Education*, volume 12, pages 257 – 279. 2001.

[27] J.E. Greer, G.I. McCalla, J. Cooke, J.A. Collins, V.S. Kumar, A. Bishop, and J. Vassileva. *Computers as Cognitive Tools: The Next Generation*, chapter Integrating Cognitive Tools for Peer Help: the Intelligent IntraNet Peer Help-Desk Project, pages 69 – 96. Lawrence Erlbaum Publishers, 1999.

[28] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199 – 200, 1993.

[29] N. Guarino and C. Welty. Towards a methodology for ontology based model engineering. In *Proceedings of ECOOP-2000 Workshop on Model Engineering*, 2000.

[30] C. I. Guinn. Mechanisms for mixed-initiative human-computer collaborative discourse. *Annual Meeting of the ACL Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 7:278 – 285, June 24 - 2 1996. Santa Cruz, California.

[31] K. M. & David W. Aha Gupta. A framework for incremental query formulation in mixed-initiative case-based reasoning. In *ICCBR'03 Workshop on Mixed-Initiative Case-Based Reasoning*, 2003.

[32] V. Haarslev and R. Mller. Racer system description. In T. Nipkow R. Gor, A. Leitsch, editor, *Proceedings of International Joint Conference on Automated Reasoning, IJ-CAR'2001*, pages 701 – 705, Siena, Italy, June 2001. Springer-Verlag.

[33] A. Hadwin, J. Nesbit, D. Jamieson-Noel, P.H. Winne, and V. Kumar. Tracing self-regulated learning in an e-learning environment. In *AERA05*, Montreal.Canada, April 2005.

[34] A. Hadwin, L. Wozney, and O. Pontin. *Instructional Science*, chapter Scaffolding the appropriation of self-regulatory activity: A socio-cultural analysis of changes in teacherstudent discourse about a graduate research portfolio, pages 1 – 38. Springer, 2005.

[35] I. Hatzilygeroudis and J. Prentzas. Knowledge representation requirements for intelligent tutoring systems. In *7th International Conference, ITS 2004*, volume 3220, pages 87 – 97. ITS, August-September 2004.

[36] F. Hayes-Roth. Rule-based systems. *Commun. ACM*, 28(9):921 – 932, 1985.

[37] M. A. Hearst. Trends and controversies - mixed-initiative interaction. *IEEE Intelligent Systems*, pages 14 – 24, September/October 1999.

[38] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe. A practical guide to building owl ontologies using the protege-owl plugin and co-ode tools edition 1.0. Technical report, The University Of Manchester, Standford University, August 2004.

[39] I. Horrocks. Using an expressive description logic: FaCT or fiction? pages 636 – 647, 1998.

[40] I. Horrocks and P. F. Patel-Schneider. Dl systems comparison. In R. M. MacGregor W. Nutt C. A. Welty E. Franconi, G. De Giacomo and F. Sebastiani, editors, *Collected Papers from the International Description Logics Workshop (DL'98)*, pages 55 – 57. CEUR, May 1998.

[41] I. Horrocks, P. F. Patel-Schneider, and F. Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1), 2003.

[42] I. Horrocks and P. F. PatelSchneider. A proposal for an owl rules language. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 723 – 731. ACM, 2004.

[43] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rom-melse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256 – 265, Madison, WI, 1998.

[44] C. S. Hunt. Organizational and end-user information systems (oeis) model curriculum. Technical report, National Organizational & End-User Information Systems Curriculum Committee Task Force, 2004.

[45] IEEE-CS. Computing curricula 2001 computer science. Technical report, Joint Task Force of ACM and IEEE-CS, 2001.

[46] A. Inaba and R. Mizoguchi. Learning design palette: An ontology-aware authoring system for learning design. In *Proc. of International Conference on Computers in Education (ICCE2004)*, Melbourne, Australia, Nov. 30-Dec. 3 2004.

[47] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7 – 38, 1998.

[48] L. Karen, W. Myers, M. Tyson, M. J. Wolverton, P. A. Jarvis, T. J. Lee, and M. desJardins. Passat: A user-centric planning framework. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*, 2002.

[49] N. A. Kinshuk and A. Patel. Adaptive tutoring in business education using fuzzy backpropagation approach. In D. Harris & R. J. Koubek M. J. Smith, G. Salvendy, editor, *Usability Evaluation and Interface Design: Cognitive Engi-neering, Intelligent Agents and Virtual Reality*, New Orleans, U, August 2001. Proceedings of the 9th International Conference on Human- Computer Interaction.

[50] H. Knublauch, M. A. Musen, and A. L Rector. Editing description logic ontologies with the protege owl plugin. In *International Workshop on Description Logics - DL2004*. Whistler, BC, Canada, 2004.

[51] V. Kumar, J. Shakya, C. Groeneboer, and S. Chu. Toward ontology of teaching strategies. In *Workshop Paper in ITS2004*, Brazil, August 2004.

[52] V.S. Kumar, J.C. Nesbit, P.H. Winne, A.F. Hadwin, and K.N. Han. *E-Learning Networked Environments and Architectures*, chapter Quality Rating and Recommendation of Learning Objects. 2005.

[53] R. Kusters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer, 1998.

[54] C. Lacave and F. J. Diez. A review of explanation methods for heuristic expert systems. *Knowl. Eng. Rev.*, 19(2):133 – 146, 2004.

[55] H. J. Levesque. *Artificial Intelligence*, chapter Foundations of a Functional Approach to Knowledge Representation, pages 155 – 212. Elsevier Science Publishers, 1984.

[56] D. K. Lidtke and G. E. Stokes. An information systems-centric curriculum iscc 991. In *1999 Frontiers in Education Conference*. IEEE, 1999.

[57] R.M. MacGregor. Inside the loom description classifier. In *ACM SIGART Bulletin*, volume 2, pages 88 – 92. ACM, 1991.

[58] C. Massey and J. Curry. Online post-secondary education: A competitive analysis. Technical report, Industry Canada, March 1999.

[59] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learner. In *Journal of Interactive Media in Education*, volume 7. KMI, 2004.

[60] B. L. McCombs and R. J Marzano. Putting the self in self-regulated learning: The self as agent in integrating will and skill. *Educational Psychologist*, 25:51 – 69, 1990.

[61] S. Menon, J. Shakya, and V.S. Kumar. Rule-based mixed-initiative scaffolding. International workshop on applications of semantic web technologies for e-learning (sw-el), Semantic Web E-Learning, Banff, Canada, October 2005.

[62] M. Minsky. *The Psychology of Computer Vision*, chapter A Framework for Representing Knowledge. McGraw-Hill, 1975.

[63] R. Mizoguchi, K. Sinitsa, and M. Ikeda. Knowledge engineering of educational systems forauthoring system design -a preliminary results of task ontologydesign. In *Proceedings of EAIED*, pages 329 – 335, 1996.

[64] F. Mulder and T.M. Weert. Ifip/unesco's informatics curriculum framework 2000 for higher education. *ACM SIGCSE Bulletin*, 33(4):75 – 83, 2001.

[65] Weert T. V. Mulder F. *UNESCO:Informatics Curriculum Framework 2000 for Higher Education.* International Federation for Information Processing (IFIP), Paris, 2000.

[66] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10:98 – 129, 1999.

[67] N. Noy and D. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University, Stanford, CA, 2003.

[68] S. Ohlsson. Some principles of intelligent tutoring. *Artificial Intelligence and Education*, 1:203 – 238, 1987.

[69] S. Paris and A. Paris. Classroom applications of research on self-regulated learning. *Educational Psychologist*, 36(2):89 – 101, 2001. Lawrence Erlbaum Associates, Inc.

[70] C. Pateras, N. Chapados, R. Kwan, D. Lavoie, and R. Tremblay. A mixed-initiative natural dialogue system for conference room reservation. In *Proc. EuroSpeech'99*, volume 3, pages 1275 – 1278. 1999.

[71] R. Plant and R. Gamble. Methodologies for the development of knowledge-based systems, 1982 - 2002. *Knowl. Eng. Rev.*, 18(1):47 – 81, 2003.

[72] O. Popescu. *Logic-Based Natural Language Understanding in Intelligent Tutoring Systems.* PhD thesis, School of Computing Science, Carnegie Mellon University, 2005.

[73] J. Prentzas, I. Hatzilygeroudis, and J. Garofalakis. A web-based intelligent tutoring system using hybrid rules as its representational basis. In *ITS 2002*, volume LNCS 2363, pages 119 – 128, 2002.

[74] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic information Processing*, pages 216 – 270. MIT Press, 1968.

[75] N. Ramakrishnan, R. Capra, and M. N. PerezQuinones. Mixedinitiative interaction = mixed computation. In *Proceedings of the 2002 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation*, volume Volume 37. Virginia Tech, January 2002.

[76] S. Rao, S. Menon, J. Shakya, M. Jordanov, L. Doherty, and V. Kumar. The importance of contexts in mixed-initiative interactions for online help. In *AAAI'05 Fall Symposia on Mixed-Initiative Problem-Solving*, pages 100 – 105, 2005.

[77] A. Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including owl. In *Proc K-CAP: 2003*, pages 121 – 129, 2003.

[78] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *Lectures Notes on Computing Science*. Springer, 2004.

[79] R. Rickel, N. Lesh, C. Rich, C. L. Sidner, and A. Gertner. Using a model of collaborative dialogue to teach procedural tasks. Technical report, AIED-2001 Workshop on Tutorial Dialog Systems, San Antonio, TX, 2001.

[80] C. Rose, P. Jordan, M. Ringenberg, S. Siler, K. VanLehn, and A. Weinstein. Interactive conceptual tutoring in atlas-andes. In *AI in Education 2001 Conference*, 2001.

[81] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2002.

[82] SAGE. Sage for learning: http://www.sageforlearning.ca/. Technical report.

[83] J. Shakya and V.S. Kumar. Capturing and disseminating the principles of self-regulated learning in an ontological framework. In *SW-EL workshop held in conjuction with K-CAP*, pages 27 – 35. K-CAP, October 2005.

[84] J. Shakya, S. Menon, L. Doherty, M. Jordanov, and V.S. Kumar. Recognizing opportunities for mixed-initiative interactions based on the principles of self-regulated learning. In *AAAI'05 Fall Symposia on Mixed-Initiative Problem-Solving*, number Workshop on Mixed-initiative interactions for Problem-solving, pages 117 – 122, Arlington, VA, Nov 2005.

[85] E. H. Shortliffe, R. Davis, and B. G. Buchanan. Production systems as a representation for a knowledge- based consultation program. *Artifical Intelligence*, 8:15 – 45, 1977.

[86] P. L. Smith and T. J. Ragan. *Instructional Design*. John Wiley & Sons. Inc., 1999.

[87] J. F. Sowa. *Knowledge Representation - Logical, Philosophical and Computational Foundations*. Brooks/Cole, 2000.

[88] G. Tecuci, M. Boicu, C. Ayers, and D. Cammons, editors. *Personal Cognitive Assistants for Military Intelligence Analysis: Mixed-Initiative Learning, Tutoring, and Problem Solving*, Proceedings of the First International Conference on Intelligence Analysis, McLean, VA,, May 2005. McLean, VA,.

[89] G. Teege. Using description logics in intelligent tutoring systems. *Working Notes of the 1994 Description Logic Workshop*, 1994.

[90] L. Williams and R. Kessler. *Pair Programming Illuminated.* Addison-Wesley, 2003.

[91] P. H Winne. Self-regulated learning viewed from models of information processing. In B. J. Zimmerman & D. H. Schunk, editor, *Self-regulated learning and academic achievement: Theoretical perspectives*, pages 153 – 189. Hillsdale, NJ: Erlbaum, 2 edition, 2001.

[92] P.H. Winne. Experimenting to bootstrap self-regulated learning. In *Journal of Educational Psychology*, volume Vol. 89, pages 397 – 410. American Psychological Association, September 1997.

[93] P.H. Winne, J.C. Nesbit, V.S. Kumar, and A.F. Hadwin. Supporting self-regulated learning with gstudy software: The learning kit project. In *International Journal of Technology, Instruction, Cognition and Learning.* OCP, 2005.

[94] B. Woolf. *Encyclopedia of Artificial Intelligence*, chapter AI in Education, pages 434 – 444. John Wiley & Sons, Inc, 1992.

[95] B. Zimmerman. *Becoming a Self-Regulated Learner: An Overview. Theory Into Practice*, volume Volume 41. Spring 2002.

Index