5-2014

Procedural-Reasoning Architecture for Applied Behavior Analysis-based Instructions

Edmon Begoli
University of Tennessee - Knoxville, ebegoli@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Edmon Begoli entitled "Procedural-Reasoning Architecture for Applied Behavior Analysis-based Instructions." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Bruce J. MacLennan, Major Professor

We have read this dissertation and recommend its acceptance:

David F. Cihak, Lynne E. Parker, Bradley Vander Zanden

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

5-2014

Procedural-Reasoning Architecture for Applied Behavior Analysis-based Instructions

Edmon Begoli
University of Tennessee - Knoxville, ebegoli@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Edmon Begoli entitled "Procedural-Reasoning Architecture for Applied Behavior Analysis-based Instructions." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Bruce J. MacLennan, Major Professor

We have read this dissertation and recommend its acceptance:

David F. Cihak, Lynne E. Parker, Bradley Vander Zanden

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Procedural-Reasoning Architecture for Applied Behavior Analysis-based Instructions

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Edmon Begoli

May 2014

Abstract

Autism Spectrum Disorder (ASD) is a complex developmental disability affecting as many as 1 in every 88 children. While there is no known cure for ASD, there are known behavioral and developmental interventions, based on demonstrated efficacy, that have become the predominant treatments for improving social, adaptive, and behavioral functions in children.

Applied Behavioral Analysis (ABA)-based early childhood interventions are evidence based, efficacious therapies for autism that are widely recognized as effective approaches to remediation of the symptoms of ASD. They are, however, labor intensive and consequently often inaccessible at the recommended levels.

Recent advancements in socially assistive robotics and applications of virtual intelligent agents have shown that children with ASD accept intelligent agents as effective and often preferred substitutes for human therapists. This research is nascent and highly experimental with no unifying, interdisciplinary, and integral approach to development of intelligent agents based therapies, especially not in the area of behavioral interventions.

Motivated by the absence of the unifying framework, we developed a conceptual procedural-reasoning agent architecture (PRA-ABA) that, we propose, could serve as a foundation for ABA-based assistive technologies involving virtual, mixed or embodied agents, including robots. This architecture and related research presented in this dissertation encompass two main areas: (a) knowledge representation and computational model of the behavioral aspects of ABA as applicable to autism intervention practices, and (b) abstract architecture for multi-modal, agent-mediated implementation of these practices.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Prevalence of Autism

Autism Spectrum Disorder (ASD) is a complex developmental disability characterized by impairments in social interaction and communication and by restricted, repetitive, and stereotyped patterns of behavior (American Psychiatric Association, 2000). It is a prevalent and challenging condition affecting 1 in 88 children (Baio, 2012) and 1 in 50 boys (Blumberg et al., 2013).

1.2 Therapies, Treatments, and Interventions

While there is no known cure for ASD, there are a number of interventions aimed at remediation of the symptoms of the disorder. These interventions for individuals affected by autism range from pharmacological therapies, diet modifications, vitamin therapy, occupational therapy, speech and language therapy, to behavioral and developmental approaches (Volkmar et al., 2005). Behavioral and developmental interventions, based on demonstrated efficacy (Foxx, 2008), have become the predominant treatments for improving social, adaptive, and behavioral functions in children.

1.3 Applied Behavioral Analysis and Discrete Trials

The focus of the research of this dissertation is on the group of behavioral treatment interventions based on the principles of Applied Behavior Analysis (ABA) and derived from the work of Lovaas et al. (1981).

ABA is a generic behavioral intervention not specific to autism, although frequently applied in the field of special education. It is a discipline of behavioral science concerned with the application of principles of behaviorism in practical settings such as schools, clinics, workplace, society, etc., with the aim of addressing socially significant behavioral issues such as behavioral problems, learning and habits. Cooper et al. (2007) define ABA as:

> a scientific approach for discovering environmental variables that reliably influence socially significant behavior and for developing a technology of behavior change that takes practical advantage of these discoveries.

Baer et al. (1968) defined the seven dimensions that describe the essential characteristics of ABA:

- Applied - ABA deals with problems of demonstrated social importance
- Behavioral - interventions deal with measurable, observable behavior
- Analytic - ABA requires objective demonstration that its procedures are causing the behavioral effect
- Technological - techniques making up the particular interventions need to be describable at the level of details at which anyone with appropriate training and resources could replicate the procedure and produce the same results just by reading the description of the intervention
- Conceptual Systematic - ABA interventions originate from well-established scientific and theoretical foundations of behavioral science, so its methods and procedures must be based on these well-established principles

- Effective - objective of the applied interventions is to produce strong, observable, and socially important effects. Although measure and analysis of the behavior is important, the end goal is to change existing or introduce new behaviors. If this is not happening, then intervention is not effective and hence not working

- General - ABA interventions need to be general and persist over time. They are designed to be effective in new environments and to continue to have effect even after the original treatments have with been withdrawn

According to (Foxx, 2008, p.821), ABA incorporates all of the factors identified by National Research Council (US). Committee on Educational Interventions for Children with Autism (2001) as characteristic of effective interventions in educational and treatment programs for children who have autism.

Intensive ABA, according to the American Academy of Pediatrics (Myers et al., 2007), was found to be the most effective of all behavioral and developmental approaches compared. Three comparative studies (Cohen et al., 2006; Eikeseth et al., 2002; Howard et al., 2005) found that intensive ABA is the most efficacious at 25-40 hours of individualized hours of treatment with a therapist. With prevalence rates stated earlier (1 in 88 children) and with high therapy hourly costs, this effective therapy, at the recommended levels, is largely inaccessible to many of the patients in need.

1.4 ABA Based Instructional Methods

1.4.1 Discrete Trials Training

Discrete Trials Training (DTT) is a form of an individualized and environmentally restricted ABA intervention with the aim of teaching new skills, discriminations, and forms of behaviors. It is an evidence-based intervention that has shown systematic effectiveness in education and behavioral interventions with children with autism (Smith et al., 2007). The idea behind DTT is to accomplish large or long-term behavioral goals by breaking them down into small, achievable learning units that are to be learned in the format of a discrete trial. A particular trial is to be performed multiple times until the skill is mastered. DTT is, therefore, structured as a series of repeated, single teaching units (Lovaas et al., 1981) called *trials*, with each trial consisting of three components: discriminative stimulus (S^D), the subject's response (R), and the consequence (S^R):

$$S^D \Rightarrow S^R$$

In a DTT session, there is a pause between each trial before the presentation of the next discriminative stimulus. Smith (2001) describes the DTT in the context of special education as having the following structure:

Cue (discriminative stimulus, also called Antecedent) is a presentation of a brief and clear instructions or a question such as "What color is this?" *Prompt* is a supplemental teaching aid aimed at assisting students in responding correctly to the cue. It may be the holding of a child's hand, co-vocalization, etc. Response is student's correct or incorrect response to the instructor's cue. *Consequence* is an instructor's action following the correct or incorrect response. Correct responses receive positive reinforcements. Incorrect responses receive a clear signal that response was incorrect, followed by instructor's demonstration of what is correct response (correction). Inter-trial interval duration is predetermined amount of time between trials in teaching situations.

Discrete trials are authored by a certified DTT therapist as scripts that repeatedly can be used in a controlled setting. A simple discrete trial, as exemplified by Cosgrave (2013), might look like one of these:

Full Gestural Trials

The teacher places one red and one blue card on the table then says "point to red." The teacher then immediately points to the red card (full gestural prompt). Jane responds by pointing to the red card.

The teacher would say, "That's right! Great job!". There would be a very short pause before a new discrete trial would begin.

Partial Gestural Trials

The teacher places one red and one blue card on the table, then says “point to red”. The teacher then immediately gestures halfway toward the red card (partial gestural prompt).

Jane responds by pointing to the red card. The teacher would say “You’re right! That’s Brilliant!” There would be a very short pause before a new discrete trial would begin.

Independent Trials

The teacher places one red and one blue card on the table, then says, “point to red” Teacher gives no prompt (independent). Jane responds by pointing to the red card.

The teacher would say, “That’s right! Well done!” There would be a very short pause before a new discrete trial would begin.

1.4.2 Pivotal Response Training

Pivotal Response Training (PRT) (Koegel and Kern Koegel, 2006) is another behavioral intervention based on ABA principles. PRT is considered a naturalistic behavioral intervention because it is intended to be integrated in a natural learning setting and implemented as a component of a non-scripted, regular teaching process with naturally occurring consequences. Its intent is to promote generalization, spontaneity, and reduce prompt dependency by conducting the reinforcement learning in a natural setting.

The PRT development stems from a number of studies that identified important, “pivotal” behaviors — those essential to a broad spectrum of a child’s development areas and non-targeted behaviors. Because of the complexity and diversity of the environment that it requires, PRT is beyond the scope of this dissertation but the outcome of our research and the framework we desire to put in place could lead to implementation of the agents capable of conducting the PRT.

1.5 Role for Intelligent Agents in Special Education and Autism Therapies

Evidence suggests (Barakova et al., 2009) the role that computers and related interactive technologies can play in early childhood interventions for autism. In the paper "An Approach to the Design of Socially Acceptable Robots for Children with Autism Spectrum Disorders", Welch et al. (2010) defines the need for more accessible and cost effective technology-based autism therapies:

> an important direction for research on ASD is the identification and development of technological tools that can make application of effective intensive treatment more readily accessible and cost effective.

Furthermore, Welch concludes that:

> there is increasing consensus in the autism community that development of assistive tools that exploit advanced technology will make application of intensive intervention for children with ASD more efficacious. (p.391)

In *Autism and Learning*, Murray (Powell and Jordan, 2012) outlines the following reasons why computers suit individuals with autism:

- Contained, very clear-cut boundary conditions
- Naturally monotropic* thus context-free
- Restricted stimuli in all sensory modalities
- Rule-governed and predictable, thus controllable (despite annoying mistakes)
- Safe error-making
- Highly perfectible medium

*monotropic is a technical term introduced by Murray meaning here "inducing attention-tunnel"

- Possibilities of non-verbal or verbal expression
- Interacts co-tropically[†] with the individual, i.e., it joins the individual's attention tunnel, starts where the child is

1.5.1 Use of Intelligent Agents

Interactive, graphical user interface (GUI) based technologies have existed for decades (Ashton, 2001) (see Figure 1.1), and they have been applied in standard school settings, including recent inclusion of tablet-based interactive software (Venkatesh et al., 2013), ever since the first introduction of the applications.

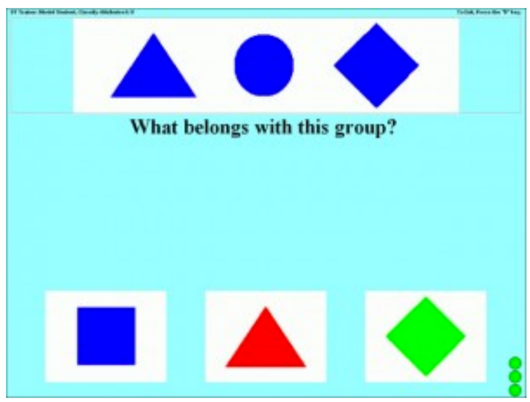

Figure 1.1: DTT a GUI-based Discrete Trial Training software (AES, 2014)

Recently, research has shifted to the use of instructional agents over the entire Milgram's Mixed Reality spectrum (Milgram and Kishino, 1994; Holz et al., 2009), with emphasis on two primary areas: use of virtual/animated pedagogical agents and use of embodied agents (Figure 1.2) for interactive play and instruction.

1.6 Motivation for Research

The use of intelligent agents, including robots, for the Autism Spectrum Disorder-focused interventions is an active area of research that brings together researchers from the fields of

[†] another technical term introduced by Murray meaning "participating in an attention tunnel"

Figure 1.2: Embodied Agent Kaspar (Wainer et al., 2010) in Autism Research. Image courtesy of Science Daily.

psychology, special education, neuroscience, computer science, and electrical engineering. These research efforts focus on different areas of the applications of robotics and virtual reality to autism therapies. We recognize the opportunity and the need to develop a unifying approach that will integrate the robotic and agent technologies and the instructional practice into a single framework. We chose the behavioral instruction, and specifically DTT because of their prescriptive, restricted and almost algorithmic nature. We also believe that the unifying framework should be independent of a technology or a specific implementation, and that it should serve as a conceptual model for any intelligent-agent like implementation.

The hypothesis of this research is therefore based on the following observations:

- the deterministic, scripted, and technologic nature (Baer et al., 1968) of ABA-based therapies is well suited for computational representation
- intelligent agents (including robots) can play an important role in the education of the children with ASD; and
- a control architecture for the instructional agent (embodied or virtual) can be abstracted away from the implementational and physical specifics

1.7 Thesis

As we have discussed in the previous sections, ABA-based early childhood interventions for ASD are effective, evidence-based approaches that, if applied early and intensively, yield significant long term improvements in an individual's ability to overcome typical social and communicative impairments associated with autism (Eldevik et al., 2009). We also have reviewed the promise and interest in the research community for use of intelligent agents (virtual, embodied, mixed) in early autism interventions as socially acceptable and comforting instructional agents.

This is a nascent field of research, and there is still an absence of a foundational computational framework for implementation of ABA-based, intelligent agent-mediated interventions that would serve as a basis for development of instructional agents. With this observation in mind, we embarked on developing a foundation for such a framework. We hypothesized about the plausibility of such a conceptual, computational model for ABA-based interventions because of ABA's foundations in principles of behavioral science, its "technologic", deterministic nature (Cooper et al., 2007, p.5), and its prescriptive, algorithmic structure (Alberto et al., 2009, p.239). To advance this goal and to confirm the hypothesis, we developed a conceptual model and a prototype of a framework that would formalize the fundamental ABA instructional concepts (Cooper et al., 2007, Ch.2), and we translated it into an executable, agent-based procedural architecture. We call this architecture a (PRA-ABA) Procedural Reasoning Architecture for Applied Behavior Analysis-based instructions. We emphasize the procedural nature of the architecture because of ABA's repetitive, prescriptive, and procedural nature. We emphasize reasoning because of the need for the intelligent agent, unlike the GUI, to semi-autonomously reason about the progress of the instruction, the student's behaviors, and his or her preferences.

1.8 The Scope of Research and Representation

ABA is a broad framework based on the principles of behaviorism. The focus of this research is on the identification of the essential principles that govern direct, interactive aspects of ABA-based learning between the student and instructor that can be translated effectively into interactions and reasoning tasks for the intelligent agent.

Outside of the scope are elements of the ABA such as single and multiple subject design, manual measurements, functional behavioral analysis (FBA), behavioral hypotheses, and ethical considerations that are usually conducted by teams of specialists and in a non-controlled environment prior or post instructional sessions.

The diagram 1.3 depicts which ABA general topics are and are not within the scope of developing the representational and reasoning formalism of this research. In-scope areas are shaded in blue.

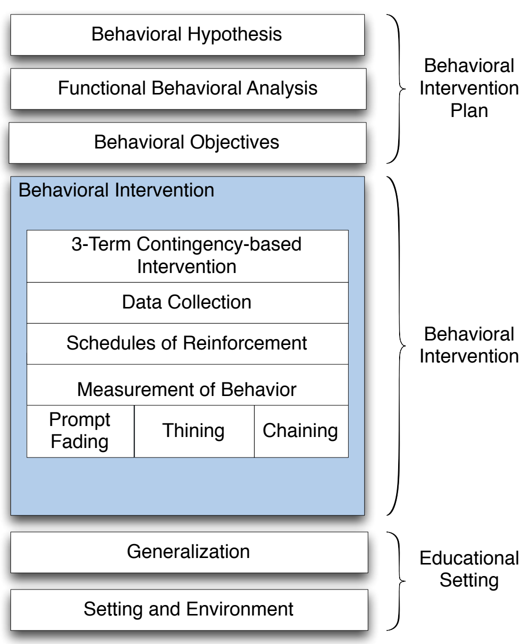

Figure 1.3: Scope of Representation

We expect that this scope will expand in future work as ABA-based instructional agents become more sophisticated and independent in their capability to conduct the behavioral interventions. For now, we are limiting the scope for the sake of practicality and limitations of the scope of the doctoral dissertation.

1.9 Approach to Research

Guided by the assumptions that behavioral instruction can be computationally represented, and that this computational representation can be translated into an architectural framework, we based our research approach around three main activities:

1. research and development of the computational foundation for ABA (an ABA ontology (Soares et al., 1997)),
2. design and development of the agent-oriented instructional control architecture, and
3. formal and practical evaluation of the resulting architecture.

1.9.1 Domain Study, Knowledge Acquisition, and Concept Analysis

In knowledge representation terms, knowledge acquisition is a process of acquisition of knowledge from human experts, books, electronic data, documents, sensors, etc. For this research, knowledge acquisition encompassed knowledge acquisition from human experts, analysis of literature, study of ABA procedures, ABA certification tutorials, and manuals. As part of this step, we sought to discover logical and ontological foundations underlying ABA as applied in early autism interventions, and to formalize the concepts and principles of ABA as an ABA ontology. The ABA ontology is discussed in Chapter 2.

1.9.2 Design and Development of the Agent-oriented Instructional Architecture

Operating under the assumption that there can be an abstract control component that would offer a sufficient and complete medium for agent-driven control of the behavioral instruction, we worked to develop an abstract, software-oriented framework that would be capable of autonomous or semi-autonomous control of the behavioral instruction. This architecture needed to support the agent's "understanding" of the key elements of the behavioral instruction and the capability to assemble them into a coherent instructional program. This area of research is covered in Chapter 4.

1.9.3 Validation and Evaluation

To validate the working hypothesis of this research we wanted to ensure that:

1. we have covered the right aspects of the domain,
2. our (formal) coverage of the domain is complete, and
3. the resulting architecture, based on the representation of the domain, can be built and that the solution works.

To arrive to these points, we:

1. evaluated the resulting theoretical and conceptual elements of the architecture with domain experts and
2. built and tested working prototypes in multiple modalities.

Validation and evaluation steps for the architecture are described in more details in chapter 5.

Chapter 2

Background and Related Work

The research presented in this dissertation encompasses two fields: behavioral instruction as applied in special education and intelligent-agents oriented applications, both in the context of autism interventions. For this reason we review the fundamentals of behavioral instruction, the background on intelligent agents, and the state-of-the-art in the robotics applications in autism research.

2.1 The Theory of Applied Behavioral Analysis

In Chapter 1 we briefly reviewed the main concepts and the meaning of the ABA as a branch of psychology that provides strategies for correcting or improving certain behaviors in individuals, especially those with special needs. In this chapter, we will survey the behavioral theories that ABA is based on, its principles and intervention techniques, and how are these intervention techniques systematically applied to achieve the desired learning outcomes.

2.1.1 Roots of ABA in Behavioral Science

Behavior is defined as an observable or measurable action exhibited by an individual. Its principles and manifestations are the subject of behavioral science, a field of study focused on the examination of causes and principles of the behavior of all species, including humans. ABA is specifically influenced by the works of behavioral psychologists Watson (1914), Skinner (1953), and Lovaas (1987).

2.1.2 Behavioral Learning

Behavioral learning is the process of learning of new behaviors or the modification of existing ones as a result of the interaction between the organism and the environment, and the nature of the interaction itself.

Under the educational classification (Lord et al., 1989), behaviors fall under three categories, namely social skills, academic skills, and challenging behaviors. Examples of social skills are greetings, raising one's hand, and shaking someone else's hand. Completing a writing task and doing oral mathematical computation are examples of academic skills. Challenging behaviors may include grabbing toys, hitting another student, screaming, or self-injury. The absence of development of academic and social skills, and presence of persistent challenging behaviors during the childhood significantly impedes the child's overall progress (Rao et al., 2008), and potential for future normal development.

Respondent Conditioning

Also known as *classical conditioning*, *respondent conditioning* (Skinner, 1938) is a basic form of behavioral learning. It involves pairing of a *neutral stimulus* with another stimulus that may elicit a spontaneous response. The *neutral stimulus* is known as the *conditioned stimulus*; the second stimulus is called *unconditioned stimulus*. The ultimate goal in respondent conditioning is for the participant to eventually transition from showing unconditioned or natural response to the unconditioned stimulus to showing conditioned response to the neutral stimulus (Figure 2.1).

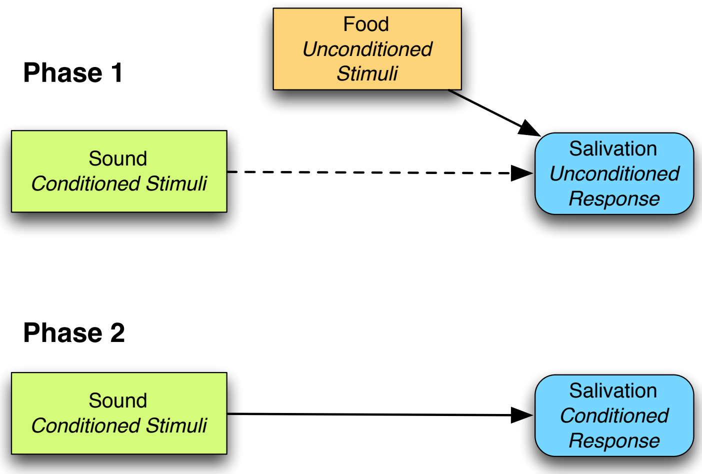

Figure 2.1: Respondent Conditioning

Respondent conditioning occasionally occurs in the academic and family setting. A child may associate sitting on a high chair by the table with eating his or her meal. The food placed on the table serves as the unconditioned stimulus while the high chair becomes the neutral stimulus. Stimulation such as excitement may be shown by the child upon seeing the food. Without associating the chair to the serving of food, no natural reflex may be shown to the chair *per se*. As part of successful conditioning, the child should eventually be showing some excitement upon seeing the high chair since it has been associated with the food. This happens as both stimuli are presented to the child.

To increase the occurrences of a positive operant behavior, rewards such as verbal praise or tokens are given. On the other hand, to decrease negative operant behavior, some forms of punishment are applied. A good example is intentional ignoring.

Operant Conditioning

Operant conditioning (Skinner, 1938) is a form of behavioral learning based on the system of rewards and punishments (consequences). Consequences are used to help introduce new behaviours or modify existing ones (Figure 2.2). With *operant conditioning* applied in an educational setting, a student learns to associate certain positive behaviors with rewards and disruptive behaviors with negative consequences.

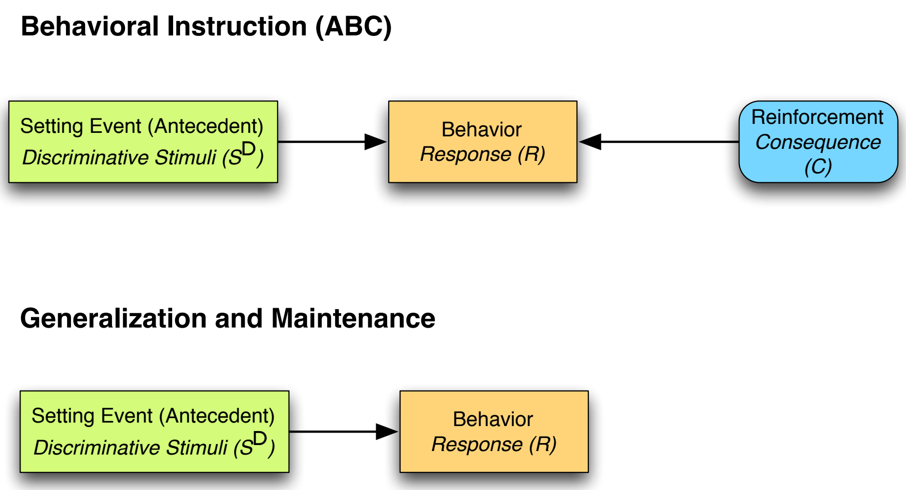

Figure 2.2: Operand Conditioning

Operant conditioning and the related three-term contingency framework are the fundamental concepts of ABA.

2.1.3 Three-Term Contingency

Three-term contingency is a framework for expression of events or situations before, during, and after a certain action of a learner who is the subject of intervention. Three-term contingency has three components: *Antecedent (A), Behavior (B)*, and *Consequence (C)*, which are known as "ABC". Telling someone to sit down is a form of *Antecedent*. When the person takes his or her seat and sits down, this action is an example of *Behavior*. Note that the behavior — sitting down — is prompted by the instruction "sit down" (antecedent). Giving verbal praise or a token like a piece of candy is an example of *Consequence*.

Consequences that follow a behavior promote the student's learning to exhibit desired behaviors or to reduce occurrence of the undesired ones (e.g. hitting, yelling, self-injury).

Three-term contingency is a core function of behavior modification that relates the behavior to its antecedent (*setting event* for the behavior) and the consequences that follow it. It is a fundamental mechanism for alteration of behavior, including learning of new behaviors.

2.1.4 Reinforcement

Reinforcement is a description of the relationship between the behavior and a consequence that immediately follows it. A relationship is *reinforcing* if the consequence increases the probability of the future occurrence of the behavior. Reinforcement is a crucial method in behavioral learning; it is used to promote recurrence of the desired behavior. Following are the principles that must guide its application in order for reinforcement to be effective.

1. Reinforcement must be contingent on the displayed behavior. Reinforcement should never be applied arbitrarily. Instead, it must be purposeful and always connected to a desired behavior. Without a display of positive behavior, no reinforcement is warranted.

2. Reinforcement must be applied immediately. Immediacy is critical in the efficacy of any form of reinforcement. Once a behavior is manifested, it should be reinforced within the first few moments. Otherwise, the student will not be able to establish the connection between the behavior and the reinforcer.

3. Reinforcement must be appropriate to the behavior. A reinforcement must be suitable to the task or behavior. A minor task requires a simple reinforcer while a more demanding task deserves a lot more. A half cookie may suffice for identifying a color but not for completing seatwork for 5 or 10 minutes.

4. Reinforcement must be specific and clear. A behavior that is reinforced using verbal affirmation must be clearly identified using specific words. Generic expressions such

as “Good job” or “Nice” are not clear enough because they do not indicate the actual behaviors. The student must know exactly what he or she has done and how well. Better alternatives include: “You finished your writing in 5 minutes.” or “You spelled all the words correctly.”

Reinforcement may be positive or negative. A positive reinforcement refers to any event or outcome that is given following a certain behavior. A verbal praise after the completion of a math homework is an example of positive reinforcement. If this reinforcement is given, it is likely that succeeding similar tasks will be completed.

A negative reinforcement often requires the removal of a pleasant event or outcome following a certain behavior. Removing time restriction on playing with certain toys is an example of a negative reinforcement.

2.1.5 Punishment

Punishment is an application of consequent stimulus (S^D) that:

- decreases the probability of the future occurrence of the behavior,
- is issued for the undesired or inappropriate behavior, and
- is issued immediately following the undesired or inappropriate behavior.

Punishment can be positive or negative. An example of a positive punishment is giving additional chores such as washing dishes or asking child to take out the trash. Limiting TV time or removing computer privileges are examples of negative punishment.

Behavioral practitioners prefer using reinforcement rather than punishment since a student may not understand why he or she is being given a negative consequence for a certain behavior. At times, students with special needs cannot distinguish what is favorable from unfavorable behavior.

2.1.6 Extinction of Behavior

Extinction is a procedure in which reinforcement for a problem behavior is discontinued in order to decrease or eliminate the occurrence of that behavior.

2.1.7 Schedules of Reinforcement

Schedules of reinforcement play an important role in the gradual development of the reinforcement-independent mastery of behaviors. Functionally, they are used to define parameters and boundaries when working with and delivering reinforcers. A good schedule of reinforcement provides both the teacher and the student the rules which govern the learning environment.

These defined rules are dynamic, changing the schedules of the reinforcement as the needs of the student change. Depending on the student's learning performance and mastery of the skill, reinforcement might be given after every correct response or for every three responses or after a certain amount of time has passed.

The two categories of reinforcement schedules are continuous schedules and intermittent schedules (Figure 2.3).

A continuous schedule of reinforcement happens when a reinforcement is given immediately following the identified or targeted behavior while an intermittent schedule of reinforcement occurs when the reinforcement is provided following every other correct behavior is executed.

One would use a continuous schedule when introducing a new behavior and an intermittent schedule when reviewing and maintaining previously learned behaviors. Both continuous and intermittent schedules provide the student with reinforcement for correctly demonstrated behavior.

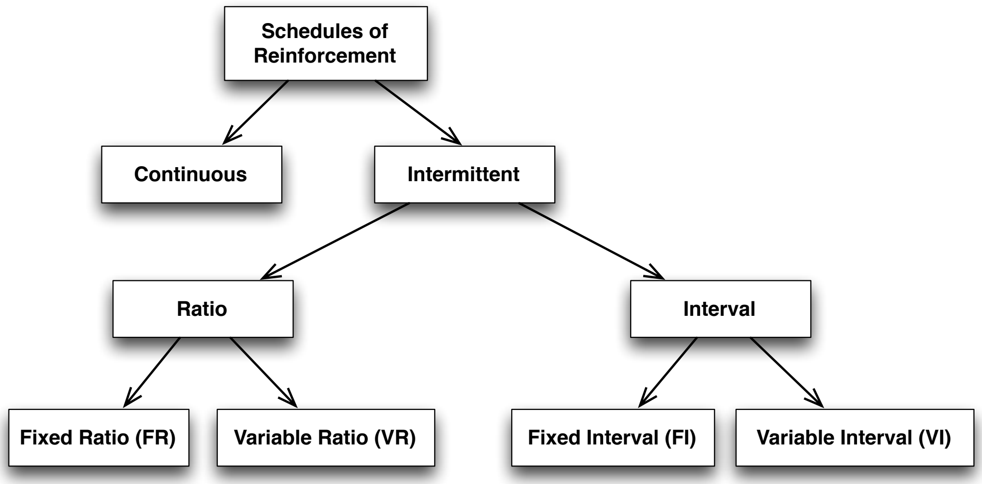

Figure 2.3: Schedules of Reinforcement

Continuous Schedules of Reinforcement

Continuous (CR) schedules of reinforcement are utilized to teach that every single time Behavior A occurs Reinforcement B will follow. Continuous schedules are used when teaching constants such as a child's name, letters of the alphabet, numbers, etc.

Intermittent Schedules of Reinforcement

In contrast to the continuous schedule of reinforcement offering no variance, an intermittent schedule of reinforcement (See Table 2.1) has four basic types:

1. *A fixed-ratio schedule (FR)* is a schedule of reinforcement in which the reinforcement occurs after a predetermined number of correct responses is given. When using a fixed-ratio schedule in discrete trial training, the value of trials must always be defined. When the value is defined as two (FR2), the student receives reinforcement every second correct response. When the value is defined as one (FR1), it is technically the same as a continuous reinforcement schedule.

2. *A variable-ratio schedule (VR)* is a schedule of reinforcement in which the reinforcement must average out as a specific number. When using variable-ratio schedules in discrete trial training, the value can be any number, but must always be defined. For example, if a student has a total of 10 correct responses and was provided 5 reinforcements throughout the trial, the reinforcement was delivered for every 2nd correct response on average.

3. A *fixed-interval (FI)* schedule is a schedule where the reinforcement becomes available only after target behavior occurs before the set time interval has ended. When used in discrete trial training, the schedule is represented as (FI) plus the selected measure of time. For example, if the student's target behavior is to stay in his or her designated area for 2 minutes, the schedule is represented as FI2. At the end of two minutes, if the student remains in his or her designated area, the reinforcement is given.

4. A *variable-interval (VI)* schedule is a schedule of reinforcement in which a certain period of time must pass prior to student's reinforcement. The availability of reinforcement must average out to a specific interval of time. For example, a VI 4 schedule would indicate that reinforcements are available on average every 4 minutes. As with the fixed-interval schedule, the student must be observed performing the desired behavior before reinforcement is received.

Thinner and Thicker Schedules

The terms *Thinner Schedule of Reinforcement* or *Thicker Schedule of Reinforcement* may be used to define adjustments made to the student's currently implemented schedule of reinforcement.

For example, a FR5 schedule (reinforcement delivered after every 5th correct response), then a "thinner" schedule would mean increasing the amount of correct responses needed to gain reinforcement. The "thinner" schedule would look like FR7 (where reinforcement

Table 2.1: Schedules of Reinforcement

Type of Reinforcement	Definition
Continuous (CR)	Reinforcement is provided after each correct response.
Intermittent	Reinforcement is provided for some, but not all, correct responses.
Ratio reinforcement schedule	Reinforcement is provided after a specific number of correct responses. Two types of ratio reinforcement schedules may be used: fixed and variable.
1. Fixed Ratio Schedule (FR)	Reinforcement is delivered after a specified number ,n, of correct responses. Symbol: FRn
2. Variable Ratio Schedule (VR)	A student is reinforced every n-th correct response on average. Symbol: VRn - VR5 reinforce, on average, every fifth behavior. For example, if the average reinforcement is set as 3, instructor will reinforce either second, third of fourth correct response and the counter will reset.
Interval reinforcement schedules	Learners are reinforced after a period of time.
1. Fixed interval schedules (FI)	A learner is reinforced following a specified t amount of time (in minutes) of correct behavior. (e.g., staying seated) Symbol: FIt Example: FI5 where 5 is 5 minutes.
2. Variable interval schedules (VI)	Reinforcement is provided after an average amount of time t where t is the number of minutes. Symbol: VIt Example: VI3 means that instructor might provide reinforcement on an average every 3 minutes.

would be delivered after every 7th correct response) thus “thinning” the amount of reinforcement the student is given for correct response or behavior.

As an example of a “thicker” schedule, the currently implemented schedule of FR5 (reinforcement delivered after every 5th correct response) would decrease to FR3 (reinforcement delivered after every 3rd correct response) thus “thickening” the reinforcement schedule.

Combining Schedules of Reinforcement

In an educational or classroom setting, a combination of schedules is often required to maximize a student's success of new skills while maintaining previously attained skills. Combinations of reinforcement schedules allow teachers to use verbal praise on a continuous (FR1) schedule of reinforcement for previously acquired skills while the new or introduced skill is on a FR1 with tangible reinforcement). One would write *FR1 praise*, *FR2 token* to indicate differences in the discrete trial scripting notes.

2.1.8 Thinning of Reinforcers

Thinning of reinforcement refers to the introduction of spacing into a reinforcement schedule, and it is a technique employed to gradually remove the reinforcers. Ideally, the instructor's objective is to remove the need to reinforce desired behaviors so that student can exhibit behaviors independently without needing a reward (e.g. writing his or her name without needing a reward). *Thinning* is a gradual process where such state of skill mastery is acquired over time by spacing out the reinforcers.

2.1.9 Prompting and Prompt Fading

Prompting is help or a cue provided by the instructor to assist the student in performing the correct behavior. For example, highlighting or pointing to the correct object to be selected by the student is an example of *prompting*. *Prompt fading* is the gradual removal of the prompt as the student acquires a certain level of mastery of the behavior. The ultimate goal is to remove the prompt altogether and avoid the student's dependency on the prompt.

2.1.10 Chaining

Depending on their needs and level of development, some students may be unable to handle certain activities that involve long, tedious, or more complicated procedures. In such cases, the teacher may employ *chaining*, a teaching technique that involves breaking down a difficult task into smaller units to make learning easier. An example of complex behavior may be joint attention. This learning objective might be acquired by breaking it down into maintenance of the gaze and focus of the conversation by chaining it together in step-by-step fashion.

Forward Chaining and Backward Chaining

Chaining is also helpful in behavior modification whereby the teacher can disrupt a chain of actions to stop a certain behavior from happening. In chaining, the student can be assisted in two ways — either forward or backward. *Forward chaining* is a strategy that helps the student complete tasks from beginning to the end of the process (Figure 2.4). The student is assisted with the first step of the sequence of tasks and then on to the next until the last one is mastered. Here, the student learns to connect the steps with the use of prompts and reinforcers. For example, if a student is to be taught how to play fair, this behavior could be broken down into smaller tasks such as learning to borrow or ask permission to use a toy that another child is playing with, sharing toys with other kids, and keeping the toys in their proper containers. The training provides a framework of sequential, succient steps for typical behaviors.

Here are the steps in administering *forward chaining*:

1. Determine what behavior is desired.
2. Break down the task to simpler steps.
3. Demonstrate the initial step and give reinforcer for the skill to be acquired.
4. Take note of the improvement (or lack of it) and determine how the student could be helped differently.
5. Prompt the student to the next step once the initial task is mastered.
6. Do the same for the rest of the tasks.

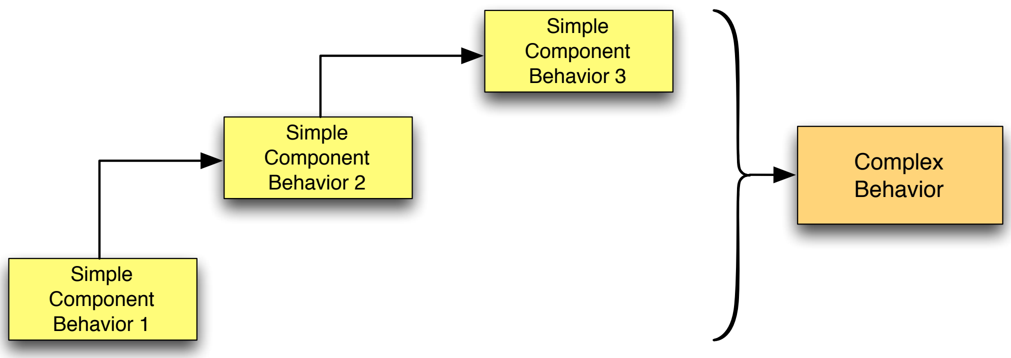

Figure 2.4: Forward Chaining

Backward chaining is the opposite - the student tries to master the steps by starting from the final task all the way to the initial step (Figure 2.5). For example, a child knows how to keep his or her toys, but grabs all the toys he or she wants every play time. In this scenario, backward chaining is most appropriate for teaching the child the first two steps. The teacher begins by prompting the child to say, “May I use this toy?” and waiting for the other child to respond favorably. After this is mastered, the child may then be taught how to share toys with prompts and sample demonstration from the teacher. From here, the child is now ready to be prompted to the last step of the target behavior. The entire chain must be rehearsed until mastered and the child could perform the behavior independently.

The steps in backward chaining include the following:

1. Establish the desired behavior.

2. Determine the individual steps in the sequence.

3. Identify the last step and then teach it using reinforcers.

4. Gather data on the progress of the student in developing a particular skill.

5. Teach the student the task preceding the last step once the final task is learned.

6. Introduce the previous step and then the one preceding it until all the skills are mastered backwards.

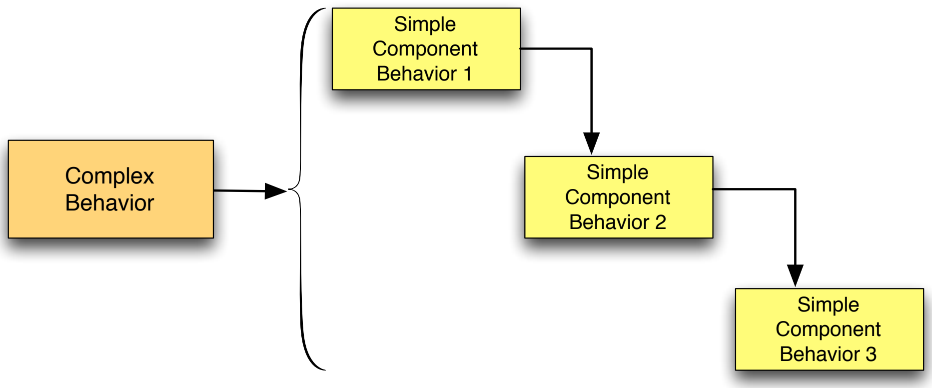

Figure 2.5: Backward Chaining

2.1.11 Analysis of a Behavior Change

Analysis of behavior change is the quantitative analysis of the relationship and manipulations of the independent variable (intervention condition) and its effects on the dependent variable (behavior). Mathematically, analysis of behavior change is a statistical method for tracking the frequency of behavior and analyzing the trend using linear regression (Seber and Lee, 2012). As part of the analysis of behavior change, data on skill acquisition and the change in behavior should be recorded and analyzed on a regular basis. The data collected through this process needs to be reviewed by the supervisor of the instruction and used to measure the student's progress. This data also serves as a basis for any adjustments to the design of the instruction.

2.1.12 Generalization

Generalization is a state of mastery of the skill that allows students to transfer behaviors learned in one circumstance to another. Baer et al. (1968) state this condition as:

> A behavioral change may be said to have generality if it proves durable over time, if it appears in a wide variety of the possible environments, or if it spreads to a wide variety of the related behaviors (p. 96).

There are three types of generalization recognized in ABA:

1. *stimulus generalization* - a behavior that previously occurred with some stimulus S_1^D now occurs in the presence of the similar, but different stimulus S_2^D;

2. *maintenance* - a learned, desired behavior occurs even when the ABA setting has been withdrawn; and

3. *response generalization* - a student will exhibit similar behaviors to the same stimulus class (similar stimuli).

2.1.13 Discrete Trial Training (DTT)

As we discussed in Chapter 1, the discrete trial consists of three distinct components, namely: antecedent, behavior, and consequence.

Antecedent

The *antecedent* is the first step in discrete trial training. The antecedent provides the instigating cue or instruction to prompt the student to perform a specific behavior in order to receive reinforcement. The teacher may say, "Give me red," or "Show me nice hands." In each example, the antecedent is a specific instruction given by the teacher for the student to demonstrate a specific task. It is possible for antecedents to be nonverbal cues or visual stimuli. A teacher might point to the cubicle where the student's backpack hangs which signals to the child it is time to leave. In any discrete trial, the antecedent should provide a clear and concise signal to the student to perform a specific behavior.

Behavior

The *response* or *behavior* of the student just after the antecedent is the second step in a discrete trial. The student may elicit the correct response, an incorrect response, or no response. Discrete trial training is based on the individual learner, therefore the behaviors received are broad. For example, a student may touch, point, repeat, look, or perform an action as a response to the antecedent. The criteria for a correct response should be established in advance and communicated to all team members working with the student. It is critical that team members are consistent and accept only the previously established behaviors as correct. For example, if the child is asked to "Give me red," acceptable items red in color are predetermined. Discrete trial training requires defined expectations in order to promote consistency and increase the skill mastery.

Consequence

The *consequence* is the third part and final part of the discrete trial. The teacher provides the consequence immediately after the student's response in order to reinforce the student's response. In every trial, a reinforcement is provided for both a correct or incorrect response. There are two types of feedback provided during a trial: Reinforcement and Corrective feedback. When the student delivers a correct response, reinforcement is the consequence delivered immediately after the correct response. There are multiple reinforcement strategies used in DTT including, but not limited to, verbal praise,food, preferred activities, preferred drink, etc. The reinforcement, however, must be based on the learner's individual preferences. When the student delivers an incorrect response, corrective feedback is provided in order to teach the learner that his or her response was not appropriate. Corrective feedback is comprised of verbal statements such as, "No," or "try again." In DTT, the teacher must always provide a consequence immediately following a student's response. Reinforcement is used to increase the amount of correct responses, and corrective feedback is used to decrease the amount of incorrect responses. The three components of discrete trial training are used to teach students the relationship between the surrounding environment and their own behaviors. For typical students, these relationships come naturally, but some students require consistent repetition to learn basic relationships.

Examples of Antecedent-Behavior-Consequence

By clearly pairing the discriminative stimulus (S^D) with the reinforcing consequence, the learner is taught to recognize that she has given a correct response. The learner may require some assistance in generating a correct response at the beginning of the trial. This assistance is provided by the teacher in the form of a prompt. Prompts are provided after the antecedent is given to help the student perform the correct response. Example: a team member says, "Point to the cow." However, the student has not had much experience with animal recognition, so the teacher prompts the student to respond correctly by guiding the student's finger to the photo of the cow, or the teacher may model the correct response by pointing to the photo of the cow so that the student can imitate the correct response. DTT prompt forms include gestures, physical guidance, verbal, proximity, visual, and others. Prompts are used to teach and promote correct responding, however, they must be systematically faded over time in order to promote independent responding. Prompts should only be used when necessary to avoid dependency on prompts. It is in the student's best interest to work to fade prompts from any given discrete trial lesson. Once the student has progressed and mastered a set of skills with discrete trial teaching, the student's educational team may choose to identify additional skills that may benefit from discrete trial training.

It is critical to promote the generalization of skills mastered with DTT. Once the skills are maintained and utilized, the student should be able to generalize these skills to other settings and demonstrate the same skills mastered in the DTT setting. Utilizing previously mastered skills in the student's natural environments is the ultimate goal of Discrete Trial Training.

DTT has three distinct benefits as outlined by the *Texas Guide for Effective Teaching* (Texas Statewide Leadership for Autism Training, 2012):

1. A skill is made into simpler and shorter tasks that a student can easily handle.

2. By using a reinforcement, the child's level of motivation soars.

3. Tasks needed for a skill are made clear and consistent.

DTT is a proven instructional method in teaching academic, social, and language skills to children with special needs such as the children with ASD. However, DTT cannot stand alone; in fact, it must be complemented with other intervention techniques so that skills are transferred from the teaching environment to normal, everyday situations that the student may be in.

Because of the teacher-centeredness of DTT, students may often rely on the antecedents or on the teacher as well as the anticipated reinforcements. Another disadvantage of this technique is that students tend to manifest a communication style that is passive since they simply respond to teacher-directed stimuli instead of initiating interaction without prompts.

2.2 Related Work in Computer Science

The focus of our research is the development of an intelligent agents-oriented, integrative framework that should encompass virtual, mixed, or embodied setting. In this section, we review the fundamental concepts of agents-oriented architectures and their applications in regular and special education. As a special case (Cordeschi, 2013) of agent-oriented application we review the recent and relevant robotics applications in autism therapies.

2.2.1 Intelligent Agents and Mixed Reality Applications

Intelligent agents are entities that autonomously or semi-autonomously operate in the environment, receive inputs, reason about the inputs, their state, the state of the world and of other actors' and issue actions. We review here three types of agent architectures of interest, namely: procedural reasoning architecture, architecture for animated pedagogical agents, and behavioral-instructional architecture.

2.2.2 Agent Architecture

An agent architecture, in addition to the typical input-output and processing components that characterize the software systems, needs also to support the capability of the agent to act in the environment and to reason about its own actions and the actions of the other participants in that environment. Russell and Norvig (2010) define the agent as the assembly of the architecture and the agent program. The architecture represents all the agent's components through which it interacts with the environment. The agent program is the 'code' that runs the entire process integrating the architecture with the reasoning elements.

$$\textit{agent} = \textit{architecture} + \textit{program}$$

Agent Function

The agent function is the mapping between the agent's percepts and its actions. Mathematically, an agent function is defined as a mapping from any given percept sequence (P^*) to an action of an agent (A). A percept sequence P^* is a history of everything that the agent has ever perceived.

$$f : P^* \rightarrow A$$

Percepts of the ABA Agent's function are environmental data representing the agent's and student's position in the environment, agent's own and student's actions (student's behavior), and other related environmental percepts, and it produces positional and instructional actions.

2.2.3 Belief Desire Intention (BDI)-based Agents

The *Believe, Desire and Intention (BDI)* architecture (Rao and Georgeff, 1991) is a well established agent architecture for *rational* agents. It is used to represent and model an agent's internal state and what it knows about itself and its environment, its goals, and its plan to achieve agent's desired state of the world.

2.2.4 The Procedural Reasoning System (PRS)

PRS (Georgeff and Lansky, 1987) is an agent design framework based on the Belief-Desire-Intention (BDI) model for intelligent agents. It was developed for the tasks requiring agents to follow prescribed procedures that can be dynamically adjusted based on the state of the world and the agent's beliefs (Figure 2.6).

The procedures for the PRS system are externally defined and deployed to the PRS system as knowledge areas. The PRS system is controlled by a PRS interpreter accepting the environmental events and running them against the knowledge areas. The PRS is a blueprint for an intelligent system that can adjust its beliefs and dynamically select procedures to follow.

A PRS consists of the following components:

1. Database - store for the relevant facts about the world.

2. Goals - conditions over an interval of time on internal and external state descriptions (desires) that the agents need to accomplish.

3. Knowledge Areas (KAs) - plans that define sequences of low-level actions toward achieving a goal in specific situations.

4. Intentions - a selection of KAs for current and eventual execution.

5. Interpreter or inference mechanism that runs the system.

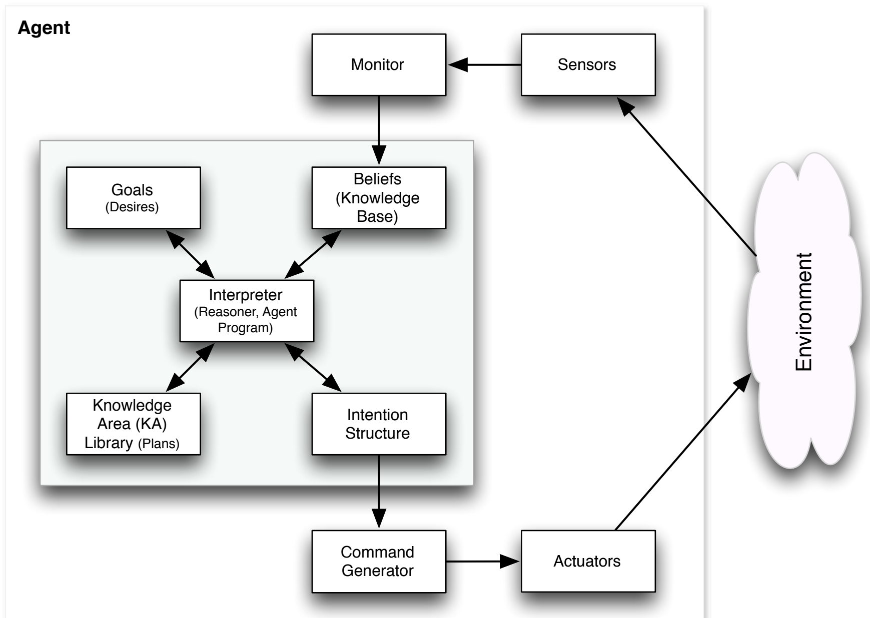

Figure 2.6: Procedural Reasoning System (PRS) Architecture

2.2.5 Animated Pedagogical Agents (APA)

APA (Johnson et al., 2000) are software agents that interactively support human learners with instructional prompts and guidance while the learner is engaged on a learning task (Figure 2.7). The degree of "intelligence" of the agents varies, ranging from a very simple assistive agents run by scripted, interactive scenarios often found in a popular online and desktop learning software, to more advanced types (Sklar and Richards, 2010) that are capable of observing and judging the learning progress.

The latter, which are of interest to our research, consists of an interactive component as well as internal teaching, knowledge domain, and adaptivity components that enable these agents not only to interact with a learner, but to reference its own knowledge base and to adapt to the learner's progress. Figure 2.7 depicts this architecture.

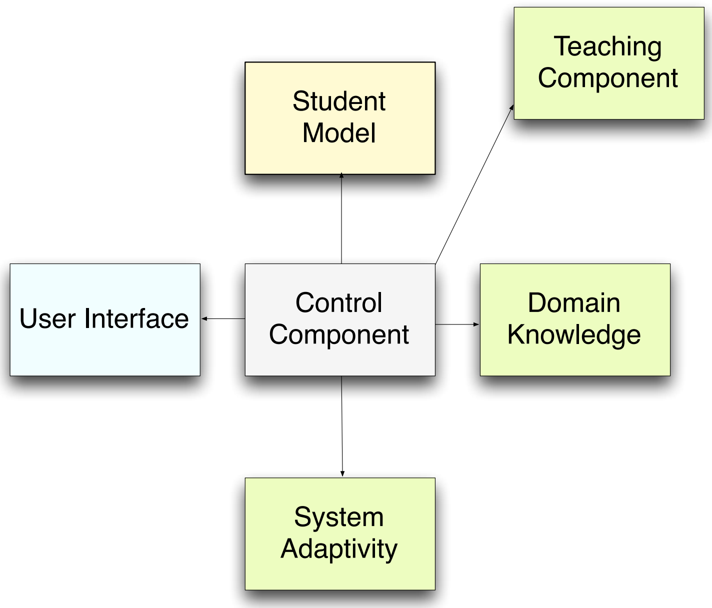

Figure 2.7: Architecture of Pedagogical Agents (APA) (Sklar and Richards, 2010)

2.2.6 Embodied Conversational Agents (ECA)

Another category of intelligent agents in which we are interested consists of standard Embodied Conversational Agents (ECA) as well as their more evolved counterparts called ECA with ASD (Milne et al., 2011). The purpose of that research is to discover the characteristics that make them unique in a real-world therapy (Serenko et al., 2007).

The standard ECA

The standard ECA, also called neurotypical ECA, is an intelligent agent endowed with artificial intelligence-like features. ECAs have existed for years and are, in essence, agents. According to some researchers (Wooldridge, 1997), an agent is an encapsulated computer system located in a particular environment and capable of flexible, autonomous action in that environment in order to reach its objectives. In many ECA systems, the module that processes behavior is often implemented as one monolithic unit or as multiple components with complex interrelations (Cassell, 2000). One of its main components deals with the representation of the knowledge structure of multimodal behavior. An interesting example of standard ECA is the Northwestern University Multimodal Autonomous Conversational Kiosk (NUMACK) (Werf, 2008), which, through speech, gestures, and facial expressions gives directions on campus. Users can interact with NUMACK with head movements and speech.

The ECA with ASD

An ECA with ASD is an ECA known to have human characteristics such as pleasure, agreeableness, and dominance that are computable (Vilhjálmsson et al., 2007). It contains a module that incorporates autistic features (personality, emotion, and mood) into its interaction with a human subject. The main feature that differentiates an ECA with ASD from the standard ECA is the existence of a conversation data set that contains, for a given topic, typical sentences uttered by a child with autistic syndrome. These conversations should display emotion and personality traits measured according to a given mathematical formula. They are the result of the careful work of an expert psychologist who extracts information from a child with ASD. This data acquisition procedure is very similar to that of a knowledge engineer collecting and eliciting knowledge from an expert and representing and storing it in a knowledge base in order to build a useful expert system.

Here the interactions between the ECA with ASD and the subject range from simple communications (Tepper et al., 2004) to more elaborate emotion problem-solving tasks that consist of identifying inconsistencies. A very good example is the Rachel system, an ECA developed at the University of Southern California, for interactions with children with autism (Mower et al., 2011). Such systems handle the syntactic and semantic structure of the content, the affective state of the ECA, and ascribe particular segments of text with appropriate nonverbal behaviors (Lee and Marsella, 2006). The purpose of the ECAs with ASD is to facilitate the communication between the parents, therapists, and the child with ASD, as well as to promote the communication skills between the child with ASD and the ECA with simulated ASD conditions.

2.2.7 Intelligent Agent-based Serious Games

An intelligent agent-based serious game is a therapeutic approach to the application of gaming and virtual reality (VR) technologies for the improvement of the social and communication skills of children with ASD (Bartoli et al., 2013). The most recent representative technology of this approach is the ECHOES (Bernardini et al., 2013), an agent-based technology built on the principles of the FAtiMA (Dias and Paiva, 2005) architecture.

Figure 2.8: "Andy" avatar from the ECHOES application. From Porayska-Pomsta et al. (2012).

The PRA-ABA shares many characteristics and objectives with ECHOES. Both are intelligent agent-based, focused on the improvement of the social behaviors, and capable of reasoning about the student's state. The main differences are the modality of the implementation (ECHOES is VR-only), and ECHOES' focus on specific social skills (joint attention). The focus of the PRA-ABA-related research is on the development of the multi-modal, reusable architecture for behavioral instructions.

2.2.8 Embodied Agents

Recent studies (Diehl et al., 2012) have shown that embodied agents, such as robots, present a possibly promising alternative as instructional agents or participants in autism therapies. Diehl's study organizes use of embodied agents into four broad categories:

- use of robots to observe the response of individuals with ASD to robots or robot-like behavior in comparison to human behavior
- use of robots to elicit behaviors
- use of robots to model, teach, and/or practice a skill
- use of robots to provide feedback on performance

All four uses are initial attempts to introduce robots into the therapeutic process. The most significant advances and the most related research to our approach is happening in the field of Socially Assistive Robotics (SAR).

2.2.9 Socially Assistive Robotics (SAR)

SAR (Feil-Seifer and Mataric, 2005) is a recently developed (2005 and onwards) and active field of research within computer science that lays at the intersection of social and assistive Robotics. Social robotics involves robots interacting with humans socially through speech, gestures, or other forms of human-recognizable expression (Breazeal, 2003; Shamsuddin et al., 2012). The field of *Assistive Robotics* (Breazeal, 2003) focuses on robots that aid people in general and those with special needs such as those in physical therapy and rehabilitation. SAR focuses on design and implementation of robots that assist humans, often therapeutically, through social means. One of the earliest applications of this field is in autism therapies, where SAR explores the design and development of the robots that assist and encourage children with autism to develop social skills.

According to Scassellati et al. (2012), the main role of SAR system in autism therapy is to promote the development of social skills in children. This role is to be fulfilled by designing robots to partake in relevant therapeutic interactions such as capturing, maintaining, and evoking joint attention, imitation, and mediating turn-taking. The authors emphasize that, within SAR, autonomous robots that can sense and respond to human behavior are the least developed. They suggest that significant research work will be needed in order to integrate control architectures for autonomous robots into practical autism therapies. According to these authors, one of the central questions of SAR is how can the field of SAR model the behavior of the learner and serve as the object of encouragement by the therapist. A robot in this role serves as the extension of the personality of the user.

Scassellati et al. (2012) (2012) define three roles for the robots in autism therapy:

- a leader-robot demonstrates social behavior and guides the interaction
- toy-robot responds to the child and mediates social behavior between the child and others
- proxy-robots can act as proxies between the children and others in the therapy, allowing children to express emotions or desires through a robot

In a typical ABA scenario, robots most likely would be in a leadership role, although other roles might be appropriate for learning social behavior.

Behavior-Based Behavior Intervention Architecture (B^3IA)

B^3IA (Feil-Seifer and Mataric, 2008) is a behavior-based robot control architecture purposefully designed for robots engaged in various autism interventions and behavior oriented therapies (Figure 2.9). The intent of the B^3IA is to provide a modular and extensible platform for robots that can sense and interpret subjects' actions, act autonomously within established scenarios, temporally process observed data to understand the interaction with the subject as a historically meaningful event, evaluate the interaction related to target quantity and quality of social behaviors, and adjust its own behavior based on the parameters specified by a human supervising the learning program.

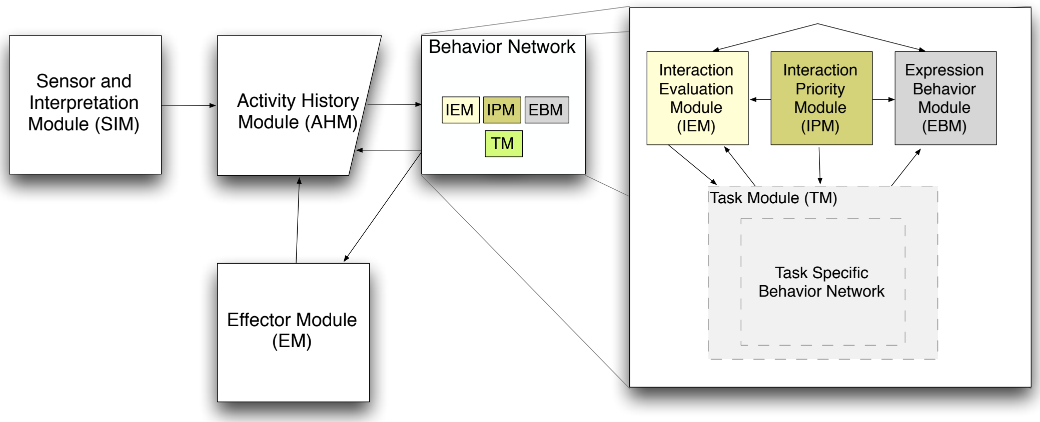

Figure 2.9: B^3IA Architecture of Behavioral Control. Adopted from Feil-Seifer and Mataric (2008).

B^3IA consists of the following modules:

The *Sensor and Interpreter Module (SIM)* controls the robot's observation of the behavior of humans and objects in the environment. The Activity History Module (AHM) collects and stores interpreted sensor data as well as interpretations of the user's and robot's behaviors (including robot's actions) in a time-indexed form. The purpose of this collection activity is to enable analysis similar to human annotation of video recordings.

The *Task Module (TM)* is a behavior network that makes most of the operational decisions related to the robot's behavior. In B^3IA, this module is where most of the task- or scenario-specific control occurs. The network consists of a combination of specific behaviors necessary for the robot to operate safely in a given scenario.

Interaction Evaluation Module (IEM) uses historical data from the AHM as input, as well as the technique proposed by Tardif et al. (1995) for quantitative evaluation of the quality of interaction for children with autism to evaluate how much interaction is currently occurring and how rich is that interaction.

The Interaction Priority Module (IPM) is designed to allow the human operator to set priorities for the intervention interactions. IPM is inspired by human-centric behavioral therapies whereby a therapist can modify his or her behavior based on the personal needs and on the relevance of the behavior to the student with ASD. Authors of B^3IA consider this feature to be important for future robot-assisted interventions in clinic or home settings because it might enable therapists and parents to develop personalized therapies.

The Expression Behavior Module (EBM) stores the robot's affect generating behaviors, such as expression of emotions, personality, and direction of the robot's attention through physical effectors.

The Effector Module (EM) controls the operation of the robot's hardware and is designed to support a variety of effectors. The B^3IA architecture is an elaborate architecture for behavior control of the robot's expression, and it represents the state of the art in Socially Assistive Robotics. B^3IA, however, emphasizes control of the behavior of the robot in the interactions with the student/subject, whereas our research is focused on the robot's ability to introduce or modify behaviors of the student.

2.2.10 Behavior Modeling with Robots

Another prominent application of embodied agents in behavioral interventions is integration of an interactive, humanoid robot into social skills interventions (Barakova and Lourens, 2013; Diehl et al., 2012). The purpose of this research (Shamsuddin et al., 2012) is to examine if the integration of the humanoid robot in social skills interventions has a positive impact on the learning of social routines. The robot is programmed to demonstrate, in a simplified manner, social communication behaviors, such as gestures and facial expressions, with the goal of teaching students with autism how to understand them and use them in social context.

2.2.11 The Need for a Unifying Framework

Behavioral instruction is a well established, highly beneficial, and well defined domain with already applied computational methods (functional behavioral analysis), and logic-like concepts (three-term contingency). However, more work is required to bring translate its principles and concepts into a form that is amenable to computation. Furthermore, there is a significant ongoing research related to intelligent agents and robotics-oriented applications in the area of early childhood development and special education. More work is needed also to formally bring these two areas together, and this is the focus of this dissertation. In the following chapters we describe the research we performed to:

- computationally formalize the behavioral instruction,
- design the architecture with behavioral-instruction specific controls and reasoning components, and
- produce a framework that is intended for use in multiple modalities.

Chapter 3

The Representation of ABA Concepts and Procedures

As mentioned in Chapter 1, an *agent* is an entity capable of acting in the environment. An *intelligent agent* is an agent that does what is appropriate in the given circumstances, is flexible to the changing environment, learns from the experience, and can maintain internal representation of the world. In the case of behavioral instruction, the instructional agent has to have and maintain the internal representation of the state of instruction, the lessons, the student, and to make decisions about its own actions based on the computations involving the internal representation and the rules of ABA.

Research in this dissertation encompasses two interdisciplinary areas: (i) knowledge representation and computational model of the behavioral aspects of ABA as applicable to autism intervention practices, and (ii) abstract architecture for multi-modal, agent-mediated implementation of these practices. In this section, we establish the fundamental concepts of knowledge representation and reasoning over dynamic domains and computational representations of behavioristic approaches, both general and specific to autism interventions.

3.1 Knowledge Representation

Knowledge Representation is a foundational discipline for any system intended for engaging in or exhibiting any form of ”intelligent“ behavior, i.e., capable of inferring and reasoning about the situations, conditions, and other (human or artificial) agents in a dynamic and interactive setting. Despite its fundamental role in artificial intelligence and related fields, a precise and comprehensive definition of *knowledge representation* remains to be established. In his PhD thesis, Smith (1982) stated the *Knowledge Representation Hypothesis* as:

> Any mechanically embodied intelligent process will be comprised of structural ingredients that
> a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and
> b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge (p. 2).

A knowledge representation scheme therefore has in some form to be understandable to a human reader, but it also has to serve as a foundation of intelligent, independent behavior of some non-human, intelligent agent capable of independent reasoning about the domain being represented.

Davis et al. (1993) offer an *intensional*[*] definition for Knowledge Representation that defines Knowledge Representation through the five different roles it serves.

According to Davis, Knowledge Representation is defined through five roles as:

1. Surrogate for the actual phenomena it represents. A Knowledge Representation system is used to enable an intelligent agent to determine outcomes by reasoning about the domain rather than taking action in it.

[*] intensional definition gives term its meaning by specifying all of its properties.

2. Set of ontological commitments. These commitments are formal representations and relationships that the agent should know about the world being represented.

3. Fragmentary theory of intelligent reasoning. This theory is expressed in terms of three components:

   (a) the representation's fundamental conception of intelligent reasoning;

   (b) the set of inferences the representation sanctions; and

   (c) the set of inferences it recommends.

4. Medium for pragmatically efficient computation. Knowledge Representation is the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information to facilitate making the recommended inferences.

5. Medium of human expression. Knowledge Representation is a language in which humans state facts, properties, relations, and other concepts about the world.

These five roles serve as the guiding principle for the representational aspects of the framework developed for this research.

3.2 ABA Ontology

Ontology (Gruber et al., 1993) is a computational medium for capturing the knowledge about the domain, the key concepts, rules, and relationships within the domain. We define a ABA ontology as a conceptual foundation for the reasoning and inference functions of the instructional architecture. It is a formalization of concepts, rules, and processes that govern the ABA-based instruction with the intention of providing the unambiguous reference for computational process such as tracking of the student's learning rate and overall progress, and appropriate schedules of reinforcements.

Through the ABA process ontology, we define the key concepts, namely: classes, relations, interactions, rules, metrics, and measures that the instructional agent uses to create the internal representation of the instructional setting and to conduct the instructional session.

In our design, the ABA ontology consists of the three parts - theoretical (formalizing the ABA concepts and instructional process), instructional (formalizing the instructional setting, progression and the participants and computational (formalizing the measures and metrics that govern the instructional process) (Figure 3.1). Theoretical and instructional aspects of the ABA process ontology relate to the descriptive and taxonomic aspects of the ABA - definition of the setting, suitable reinforcers, behavior, etc. Computational aspects of the ABA process ontology define the criteria and the computation for proper conduct of the ABA-based instructional process - three-term contingency (i.e. when to reward and when to punish), prompting, advancing of the learning objectives, etc.

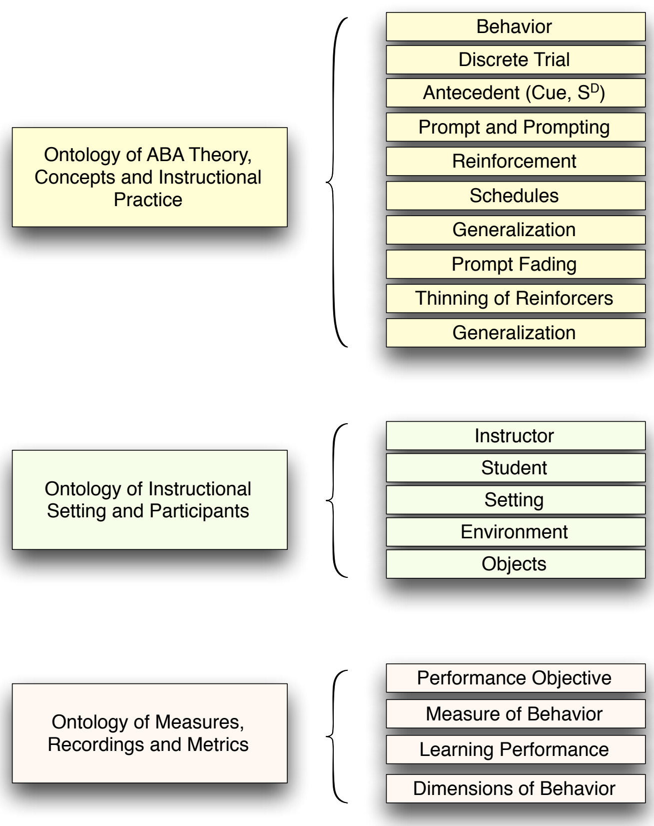

Figure 3.1: Ontology of ABA

3.2.1 Sources of Knowledge

The source of knowledge about the ABA process comes from three sources - engineered knowledge, externally provided, and internally derived knowledge.

Engineered Knowledge

Engineered knowledge is knowledge derived during the study of the literature, interviews with the special education teachers, and ABA certified instructors. This knowledge is also a result of the *formal concept analysis* (Wille, 2005), and the ontology engineering processes DOGMA (Jarrar and Meersman, 2009) and NeON (Suárez-Figueroa, 2010) that we applied to the domain materials.

Externally Provided

Although ABA is prescriptive and deterministic, a significant aspect of the ABA process is, and will remain, externally provided with respect to the agent's knowledge and representation of the instructional processes. The student's preferences, learning needs, and lesson plans are determined by the educational team consisting of the primary educators, psychologist, physical occupational and speech therapist, parents/guardians, and, sometimes, a physician. A functional relationship between the stimuli and the relationship as well as the function the behaviors might serve are determined outside of the direct instructional process (Haynes and O'Brien, 1990). The outcomes of all of these activities need to be represented and provided to the instructional agent to be used as reasoning criteria in the agent-mediated instructional process.

Internally Derived

Internally derived knowledge is the knowledge that the agent derives through the execution of the instruction, interaction, and the observations about the student, the learning process, and the environment. In our design, this knowledge is about the student's preferences, learning performance, position in the environment, and the environment itself.

Knowledge and Process Mapping

The National Professional Development Center (NPDC) on Autism Spectrum Disorders recommends nine steps for the implementation of Discrete Trials. Some steps are conducted by the multidisciplinary team consisting of educators, behavior specialists, parents, and other individuals involved in the educational and care taking process for the child with special needs. This team work is beyond the scope of the proposed ontology for the single-agent system. However, most of the knowledge and processes related to the steps proposed for the implementation of Discrete Trials are to some degree represented in the ontology. A Discrete Trial is represented as either engineered knowledge, externally provided, or internally derived. Table 3.1 outlines the mapping between the recommended steps and practices (Bogin et al., 2010) and how are these translated into the ABA Ontology.

Table 3.1: Ontology Mapping to NPDC Steps

NPDC Steps for Implementation	Ontology	Source of Knowledge
Step 1. Decide what to teach: Assessment and Summarizing Results	Antecedent, Behavior and Consequence Structure Criterion classes	External
Step 2. Breaking the Skill Down into Teachable Steps	Skill, Lesson, Session and Trial classes. Desirable and non-desirable behavior classes	External
Step 3. Setting-up the Data Collection System	Data structures for data collection on success measures, approximation, prompting needs.	Internal (externally modifiable)
Step 4. Designating Location(s)	Setting and environment classes	Either
Step 5. Gathering Materials	Reinforcers, instructional material classes	Either
Step 6. Delivering the Trials	Trial Procedure	Internal
Step 7. Massed Trial Teaching	Maintenance Trial, Prompting, Reinforcing, and Progression Rules	Internal
Step 8. Conducting Discrimination Training	Distractor class, Change of stimuli, Random rotation of stimuli and situations, Generalization	Internal, externally modifiable
Step 9. Review and Modify	Steps, Maintenance trials, Generalization rules	Internal, externally modifiable

3.2.2 Ontology of the ABA Theory

The ontology of the ABA theory covers the concepts, definitions, and rules from the general theory of Applied Behavior Analysis and its applications in the educational practice. As we discussed in Chapter 1 (Figure 1.3), we represent only concepts that are relevant to the reasoning and operational functions (instruction, rewarding, prompting, etc.) aspects of the instructional agent.

Taxonomy of Stimuli and Behavior

Stimuli and Behavior (events and actions emitted by either an agent or a student) are the top-most concepts in the taxonomy of representation of the environmental inputs. Stimulus is anything that one sees, hears, smells, tastes, or feels, and it is, therefore, a subsumptive class of all other environmental phenomena. Behavior is the range of one's actions in response to a particular situation or stimulus.

Human behavior is a complex and diverse phenomenon with the breadth and complexity of expression that is, representationally, beyond the scope of the research covered in this dissertation (see *Future Work*). Therefore, we focus our research and the ontology engineering effort on the ontology of the behavior in the context of instructional setting. We specifically focus on the functional aspects of the behavior - attributes and characteristics relevant to the learning of the skills (behaviors) in a typical early childhood and special education setting. Furthermore, from the instructional control and skill acquisition reasoning perspective, we are concerned about the recognition of the behavior and level of approximate correctness of the student's behavior and the expected behavior, rather than the entire taxonomy of behaviors.

However, we propose a representational framework in which any future behavior can be described, recognized, and measured according to the criteria we require of the reasoning agent. We call this abstract criterion a *degree of correctness* of the behavior, and it is a foundation for the entire reinforcement instructional process.

Different behaviors are qualified by different dimensions and attributes. The related recognition mechanism interprets these behaviors appropriately (see reference implementation and experiments). From a process ontology perspective and the operand value of the behavior, we need to understand if the behavior is:

1. Correct (C),
2. Approximately Correct (A),
3. Incorrect (I), or
4. No Response, Not Evident (N).

We will further define the criteria for this classification when we will talk about the correctness of the behavior later in this chapter.

Behavioral psychologist recognize verbal and non-verbal behavior as two parent categories of all possible behaviors (Burns, 1980). In our approach, we also introduce the supra-verbal behavior (see Figure 3.2) - the behavior that is both verbal and non-verbal. An example of supra-verbal behavior is a child asking for a toy and simultaneously pointing at the toy.

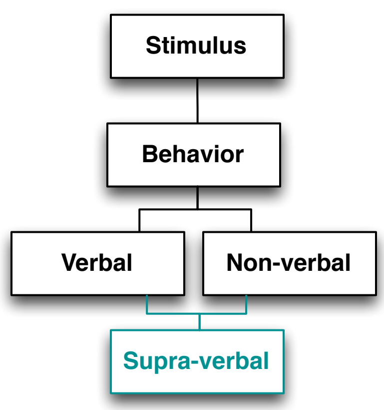

Figure 3.2: Taxonomy of Behavior

Three-term Contingency

Three term contingency is a formalization of the relationship between an environmental stimulus, behavior, and consequence.

Antecedent (A), also known as a Discriminative Stimulus S^D, is a setting event for the expected behavior.

A *setting event* is an action that serves or is supposed to trigger a behavior. This action can be any form of expression; it is usually voice, gestural, visual stimuli, or a combination thereof.

Behavior (B) - Behavior is what the student does in response to the setting event (Antecedent).

Consequence (C) is a stimulus that immediately follows the behavior. Depending on the appropriateness of the behavior, the consequence can be either rewarding (R) or a punishing (P).

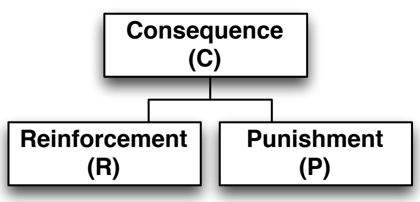

Figure 3.3: Hierarchy of Consequences

Formally, we define a three term contingency as a modal (Emerson, 1990) relationship between the discriminating stimulus A (antecedent), a response behavior B and a consequence C:

$$A \rightarrow \nabla B \rightarrow C$$

$A \rightarrow \nabla B$ is a contingency relationship meaning that antecedent A will always be followed with either expected behavior B or unexpected behavior (¬B). We use the contingency expression ∇B which is, for formula B, defined as $\nabla B \stackrel{def}{=} (\diamond B \wedge \diamond \neg B)$. We use a modal logic (Kripke, 1963) operator $\diamond$ for possibility. However, there is a consequence regardless of the occurrence or the appropriateness of the behavior. Desired or expected behavior will have a reinforcing consequence for expected, or punishment for unexpected or undesired behavior.

$$\text{Reinforcement: } (A \rightarrow \text{Expected(B)}) \rightarrow R$$

or punishing (P):

$$\text{Punishment: } (A \rightarrow (\text{ Unexpected(B)} \vee \text{Undesired(B)})) \rightarrow P$$

Reinforcement is a consequence that increases the likelihood that following the antecedent, the expectd behavior will occur.

Functionally, reinforcement is a relationship between the student's behavior and a consequence that follows that behavior. A relationship is reinforcing if it increases the probability that a given behavior will occur in the future given the antecedent stimuli.

Formally, R is a reinforcing relationship between Behavior B and a Consequence C, if the probability of occurrence of behavior B at some point of time t_j is consistently higher than the probability of occurrence of the same behavior B at some point of time t_i, if the behavior B was accompanied by the consequence C.

$$\text{Reinforces } (C, B) \iff \text{Issued}(C, B, t_i) \rightarrow P(B, t_j) > P(B, t_i)$$
$$\text{where } t_j > t_i.$$

Punishment is anything that decreases the likelihood that following the antecedent, the expected behavioral response, will occur.

Functionally, a relationship is punishing if it decreases the probability that a given behavior will occur in the future given the antecedent stimuli.

Formally, P is a punishing relationship between Behavior B and Consequence C, if the probability of occurrence of behavior B at some point of time t_j is consistently lower than the probability of occurrence of the same behavior B at some point of time t_i, if the behavior B was accompanied by the consequence C.

$$\text{Punishes}(C, B) \iff \text{Issued}(C, B, t_i) \to P(B, t_j) < P(B, t_i)$$
$$\text{where } t_j > t_i.$$

3.2.3 Ontology of Instructional Practice

This aspect of the ontology describes the procedural and structural elements of the ABA-based instructional process and the instructional process itself.

Instructional Process

An *instructional process* is a series of instructional steps aimed at reaching a particular learning goal. The *learning goal* is a mastery of some new behavior or extinction of some undesired behavior. If the behavior is complex (Figure 3.4), the learning goal is broken down into more than one step, each involving learning simple behaviors leading to a more complex behavior (chaining). The student has learned a behavior if he or she has reached the mastery criterion. *Mastery criterion* is defined as n trials with m percent success rate or higher (number of total successful trials). Typically, this mastery criterion is 80% success rate on past ten sessions of ten trials each.

Each step is learned through a number of sessions consisting of multiple, most commonly ten, discrete trials. The **Discrete trial** is the fundamental, atomic unit of instruction. It is a process based on the ABC structure, enhanced with additional instructional aid (prompt), consisting of the following components:

- cue,
- prompt,
- response, and

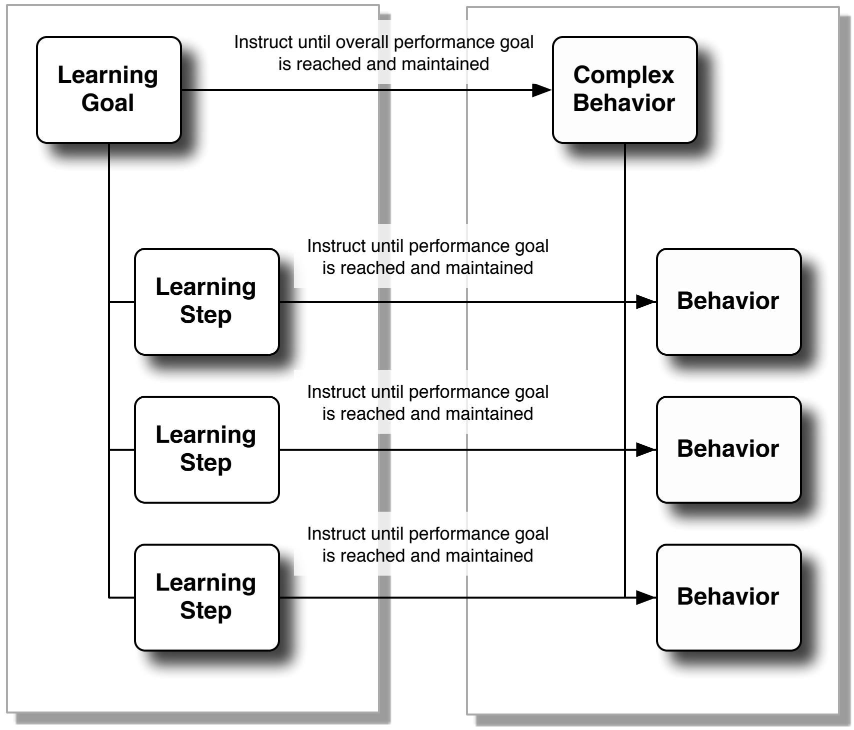

Figure 3.4: Structure of Complex Learning

- consequence.

A **Cue** is an instance of a stimulus, or a behavior, issued at the beginning of the trial. Cue is a synonym for discriminative stimulus (S^D).

A **Prompt** is an assistive teaching tool which purposely aides the student in producing the expected behavior.

We formalize this taxonomy of prompts into three hierarchies according to their genus, expression, and didactic intensity.

In terms of its genus, a prompt is either a stimulus or a behavior issued by the instructor.

Expressively, a prompt can be verbal, gestural, model, or physical. Prompts are also organized in a hierarchy of levels ordered by the degree of contact, intensiveness, and the help experienced by the student (didactic intensity). The prompt taxonomy and its hierarchical structure are discussed later in this chapter.

Temporal Attributes

A Discrete Trial is a time-bounded procedure for which we define the following time units:

Duration of the Trial (TD) is the maximal possible duration of the trial expressed in seconds. TD includes the maximum time given for the student to respond (including waiting time) and the time needed to present the prompt.

Prompt wait time (P_{wt}) is the time between the completion of the issuance of the ***Cue*** and the issuance of the prompt. P_{wt} is expressed in seconds. Typically, this is a value that is externally specified, generally between 0 and 5 seconds, by the designers of the learning process.

Consequence delay time (C_{dt}) is the time within which the consequence (reinforcement or a punishment) should be issued; typically,(C_{dt}) is between 0 and 3 seconds.

Intra-trial interval (ITI) is the waiting time between the execution of trials. (Holt and Shafer, 1973, p. 181) define the length of the ITI as a "temporal variable that may influence the number of trials to criterion, final performance reached, and stability of final performance." It is expressed in seconds, and its recommended default duration is 3-5 seconds.

Instructional Setting, Environment, and Participants

This section of the ABA ontology formalizes actors involved in the instructional setting (Figure 3.5), their properties, relationships, the setting and the environment of the instructional process. We limit the properties of the classes to the ones relevant to the instruction - i.e., we abstract away the properties of the student or the environment that are universally recognized and otherwise iconic in everyday life but of no relevance to the instruction.

*The **Student*** representation captures three aspects about the student in the instruction relevant to the instructional computation (Figure 3.6): (i) student's preferences including

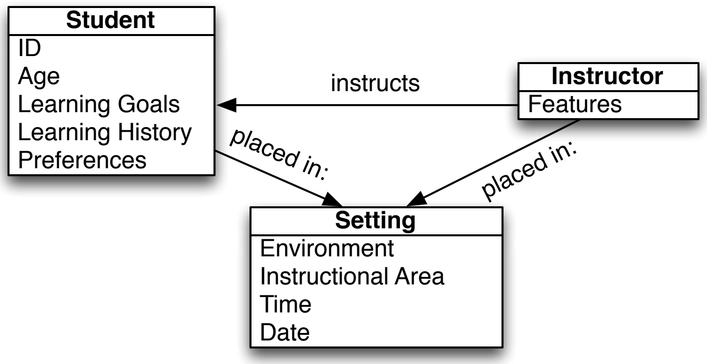

Figure 3.5: Instructional Setting

preferred reinforcers, cue, and stimuli, (ii) student's personalized learning plan which includes a list of target behaviors, learning steps, learning goals, and recommended stimuli, cues, prompts, and reinforcers, and (iii) learning record consisting of student's overall learning performance and the list of records for the learning steps, sessions and the trials. Student representation also defines the age and the identification (first name, last name, and student id) of the student.

*The **Instructor*** representation captures the features of the instructor relevant to the external appearance. These attributes are important because knowing the appearance of the instructor, correlating it with the learning performance of the student, and being able to alter it supports the *generalization* objectives of the learning process.

*The **Setting*** representation captures all the physical elements of interest within an instructional setting (Figure 3.7). These include (i) *setting* itself, which is defined as the composition of the instructional area and the environment, (ii) *instructional area* which is a specific area and the physical placement of all the objects relevant to the instruction (instructional objects, distractors, objects such as chairs), and (iii) *the environment* which is the broader setting in which the instructional setting is situated. All these elements are

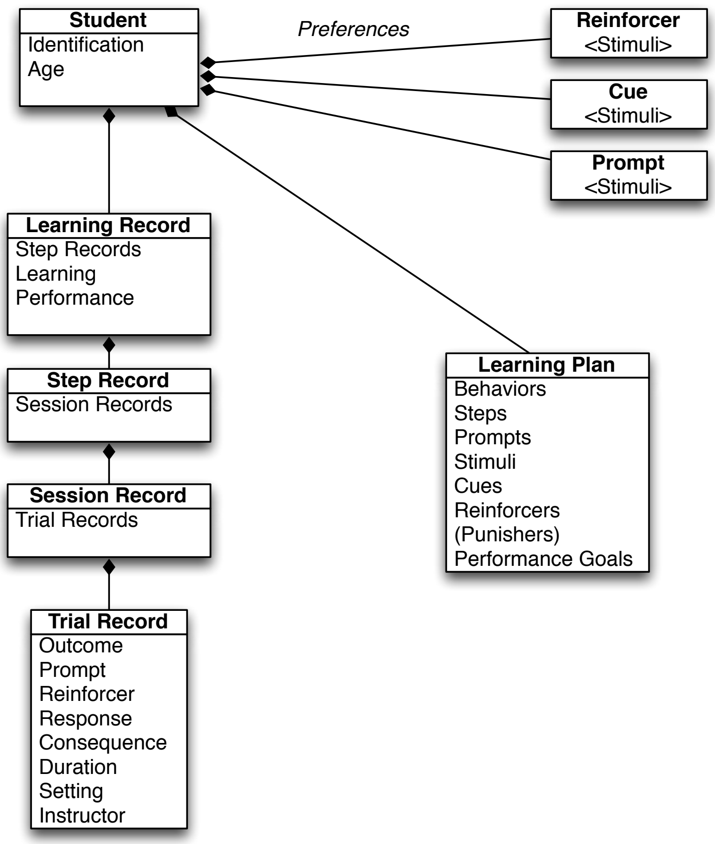

Figure 3.6: Student Representation

abstractly defined because their specific instantiation will depend on the realization of the instruction (virtual vs. embodied).

Prompting, Generalization and Schedules of Reinforcement

In the previous chapter, we described the theory of prompting and the associated best practices in the current instructional practice. We defined prompting hierarchy and different approaches to the application of prompts. Here we translate those educational definitions into rules and taxonomy of prompts as well as define the computational procedures for the application of prompting.

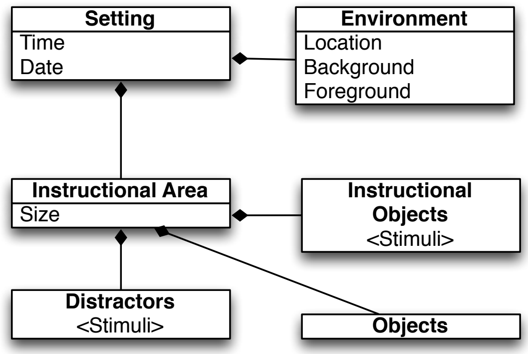

Figure 3.7: Setting Details

Expressional Taxonomy of Prompts

Following the classification proposed by the educational practice, we recognize the following taxonomy of prompts (Figure 3.8) based on their manifestation in the instructional setting:

- gestural - a movement of part of the body, a hand or the head, performed by the instructor indicating the correct response. The gestural prompt is a *non-verbal behavior*.

- verbal - a verbal expression performed by the instructor indicating the correct response. The verbal prompt is a *verbal behavior*.

- visual - a visual indicator helping student express a correct behavior. This is a *stimulus*.

- model - an expected behavior performed by an instructor. This can be a *verbal* or *non-verbal behavior*.

- physical - a full or partial physical assistance (i.e., hand-over-hand) by the instructor helping the student to perform a behavior. This is a *non-verbal behavior*.

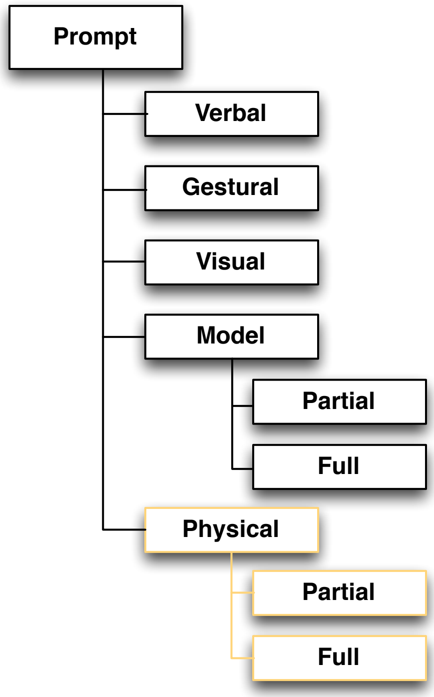

Figure 3.8: Taxonomy of Instructional Prompts

Prompting Hierarchy

A prompting hierarchy is an organization of prompts into groups based on the instructional intensity of the stimuli serving as the prompts. It is applied in the hierarchical prompting schemes such as *least-to-most* and *most-to-least*. The recommended hierarchy consists of at least three level of prompts:

1. Independent level, which requires no prompts.

2. Intermediate level, which consists of at least one level of prompts, but in practice is likely to consist of multiple levels.

3. Controlling level, which is an application of a full physical assistance.

The prompts are applied as an aid for the student hesitating to produce behaviors after the *cue* has been issued.

We formalize this procedure as:

Step 1: Agent issues a *Cue*

Step 2: Agent awaits the behavior

Step 2.a: Agent issues a *Prompt* from the appropriate hierarchy if the student does not respond within prompt waiting interval where *prompt waiting interval* ($P_w i$) is a time, measured in seconds, between the time that the *Cue* was issued and the time when the agent should issue a prompt.

Step 3: Agent issues a *Consequence*

Applications

We formalize two strategies for the application of prompting - *least-to-most* and *most-to-least*. We do not formalize *graduated guidance*[sic] and other similar applications because they are subjective and "common-sense"-based which makes them unlikely and difficult candidates for automation.

With *least-to-most* prompting (see Algorithm 3.2.3), the agent first presents the cue and expects the behavior without providing any prompts (i.e., presents the prompt from the independent level). If the student struggles to demonstrate the expected behavior across the entire session, the agent will, in the next session, present the prompt from the next level of the prompting hierarchy. The student's performance on the task serves as the criterion for the presentation of the next level of prompts or for their fading. If the student does not show any improvement, prompt levels continue to increase from the least to most assistive.

If the maximal level is exceeded, the agent will terminate the instructional program and indicate the need for lesson redesign.

Algorithm 3.2.1: prompt least to most(student,session)

for each trial in session
issue prompt at the current hierarchy level
if no increase in learning performance(LP)
  increment current hierarchy level
  if the prompt is already at the maximal level
    stop instruction and require re-design of the instruction
else
  decrement current hierarchy level
  if prompt hierarchy is already at the lowest (independent) level
    use independent level

With the *most-to-least* prompting approach (See Algorithm 3.2.3), the agent starts with the issuance of the prompts at the highest level (full physical prompt). If the student shows the improvement across the entire session, in the next session, the agent will, present the prompt from the next lower level of the prompting hierarchy. The student's performance on the task serves as the criterion for the presentation of the next level of prompts or for their fading. If the student does not show any improvement at the highest level, the agent will terminate the instructional program and indicate the need for lesson re-design.

Algorithm 3.2.2: prompt most to least(student,session, trial)

for each trial in session
issue prompt at the current hierarchy level
if learning performance(LP) increases
  decrement current hierarchy level
  use independent if the prompt hierarchy is already at the lowest (independent) level
else
  increment current hierarchy level
  if the prompt is already at the maximal level
    stop instruction and require re-design of the instruction

Prompt Dependence is the condition where the student cannot maintain the learning performance without the prompted assistance. Prompt dependence is detected by observing

the drop of learning performance (LP) when prompts are removed. We formalize this condition as:
Let PT_t be the set of prompted trials performed at time interval t and let $uPT_{(t+1)}$ be the set of unprompted trials of the same size and type performed at the time $t+1$.

If the student's learning performance for P is by n percent greater than the student's learning performance for uPT, then reintroduce the most recent prompt.

The Degree of Prompt Dependence (PD):

$$PD = LP(PT(t)) - LP(uPT(t+1))$$

Generalization is a student's desirable learning ability to transfer behaviors learned in one learning setting to other settings without a significant drop in performance. In order to evaluate generalization, the instructor needs to measure if the student can demonstrate the same or similar level of performance in a different setting (instructor, environment). For this purpose, we define the following measures:

Generalized Learning Performance (GLP) is the difference between the learning performance in the original environment ($E_{original}$) and the new environment (E_{new}). Negative learning performance indicates the percent of the loss of generalization.

$$GLP_E = LP(E_{new}) - LP(E_{original})$$

We can apply the same formula for the measurement of generalization with the new (I_{new}) vs. with the original instructor ($I_{original}$):

$$GLP_I = LP(I_{new}) - LP(I_{original}),$$

new ($S^D_{original}$) vs. original discriminative stimulus/Cue ($S^D_{original}$):

$$GLP_{S^D} = LP(S^D_{new}) - LP(S^D_{original}), \text{ or}$$

new ($R_{original}$) vs. original reinforcer ($R_{original}$):

$$GLP_R = LP(R_{new}) - LP(R_{original}).$$

The process of generalization entails alteration of the components of the instructional process, which includes practicing the trials in other settings, practicing with (i) different adults, (ii) different reinforcers, and/or (iii) different instructions/stimuli.

Generalization is formalized as a configurable procedure (See Algorithm 3.2.3) that, based on the student's learning procedure, might use different instances of the environment, instructor, reinforcers, or cues.

Algorithm 3.2.3: procedure for generalization()

$$
\textbf{if } learning\ performance\ equals\ target\ learning\ performance(LP)
\begin{cases}
perform\ maintenance\ trial \\
\textbf{if } maintenance\ trial\ is\ successful \\
\begin{cases}
record\ generalization\ step \\
alter\ some\ generalizing\ aspect\ of\ instruction \\
perform\ n\ sessions \\
\textbf{if } learning\ performance\ isbelow\ target\ learning\ performance(LP) \\
\begin{cases}
record\ generalization\ step\ failure \\
retract\ generalizing\ aspect\ of\ the\ instruction
\end{cases}
\end{cases}
\end{cases}
$$

Schedules of Reinforcement

Different schedules of reinforcement exist to support the thinning of reinforcers (Chowdhury and Benson, 2011) and independence. We present here a formalization of the hierarchy of reinforcement schedules along with a formula for the application of different schedules that we will use in the application of schedules in the reinforcement algorithms.

In ABA, reinforcement schedules are divided into Continuous and Intermittent schedules. Intermittent schedules can be applied on a fixed or variable ratio. In general, reinforcement is applied if (a) behavior meets the reinforcement criteria (correctness of behavior), and (b) reinforcement is on the appropriate schedule.

$$\text{Expected}(B) \approx \text{Observed}(B) \wedge \text{IsScheduled}(\text{Schedule(s)}, \text{Last}(\text{Reinforced(s)}), \text{Current(T)}) \rightarrow \text{Issue}(s, r)$$

where s is a student under instruction, T is a set of student's trials, Schedule is a function returning student's reinforcement schedule.

The *IsScheduled* function is polymorphic (Reynolds, 1974); its implementation depends on the type of a reinforcement schedule.

Continuous Reinforcement (CR**)** is a reinforcement schedule where each reinforceable behavior is reinforced. Hence, the *IsScheduled* function always returns boolean *true*.

Formally, we define CR as a formula:

```
if Correct(R) & Schedule(t) == CR then
        Reinforce(s,r)
```

where R is a student's response, $t$is a trial, s is a student, and r is a suitable reinforcer. *correct* and *reinforce* are abstract functions. *correct* returns a boolean value depending on the correctness of the student's response R, and *reinforce* function (procedure) applies the reinforcer r to the student s.

Fixed Ratio (FR_n**)** is a reinforcement schedule where every n-th behavior is reinforced. Formally, we define FR_n as :

```
if correct(R) & schedule(trial) == FR then:
       if index(R) modulo n == 0:
            reinforce(s,r)
```

where n is the ratio of reinforcement, and *Index(R)* is the function that returns index i of the current correct response.

Variable Ratio (VR_n**)** is the schedule of reinforcement where every n-th correct response is reinforced.

Formally, we define VR_n as:

```
i = random(0,n*2)
if correct(R) & schedule(trial) == VR:
```

```
if index(R) modulo i == 0:
    reinforce(s,r)
```

Ratio n is a randomly selected value within n distance from the current trial. This value is re-selected for each session.

Fixed Interval (FI_t) is a reinforcement schedule where behavior is reinforced every t-th minute.

Formally, we define FI_t as:

```
if correct(R) & schedule(trial) == VR:
    if (t_c - t_l) ≥ t:
        reinforce(s,r)
        t_l = t_c
```

where t_c is the time of the current trial and the t_l is the time of the last reinforcement.

Variable Interval ($VI_{\mu t}$) is the schedule of reinforcement where every n-th correct response is reinforced, on average, every t-th minute.

Formally, define $VI_{\mu t}$ as:

```
t = random(0,μt*2)
if correct(R) & schedule(trial) == VR:
    if average(t_c - t_l) ≥ t:
        reinforce(s,r)
        t_l = t_c
```

where t_l is the time since the last reinforcement, t_c is the current time, and μt is the average interval of reinforcement.

3.2.4 Ontology of Computation

ABA and Discrete Trial Training procedure is a data and computation dependent process. The progression and the outcome of Discrete Trial Training and related ABA procedures largely depend on the data collection and computations performed on this data. Instructional goals are defined in terms of the student's mastery and performance in trials and across sessions; data is collected during the trial on the correctness of the student's responses, cues, and prompts used; lesson steps are advanced when the student accomplishes a certain level of mastery which is defined in terms of the student's learning performance (percent of successful trials); the prompt levels are changed or the reinforcements are thinned out based on the student's performance on the trials. All these are computable measures derived from the collected data.

Metrics and Measures

Performance Goal and *Learning Performance* are two metrics that drive the entire ABA-based instructional process.

Performance Goal (PG) (Willett, 1988) is a desired learning performance of the student. In practice, it is expressed as a specific learning performance such as 80 % success rate over the past ten sessions.

Learning Performance (LP) is the ratio of successful trials over the interval of recording. Interval of recording can be: (i) a count of n last trials, or (ii) a count of n last trials over the time interval t.

Count-based *Learning Performance (LP_n)* defines the learning performance as ratio of successful trials T_s over the total number of trials T_n in the last n consecutive trials:

$$LP_n = \frac{T_s}{T_n}$$

Time interval-based *Learning Performance (LP_{nt})* defines the learning performance as the ratio of successful trials T_s over the total number of trials T_n within a time interval $[t_s, t_e]$ where t_s is a start time and t_e is an end time for interval recording.

$$LP_{nt} = \frac{T_s}{T_n}$$

Learning Rate (LR) is the change in the learning performance between the two periods of observation:

$$LR = \Delta LP$$

ΔLP is a difference between current learning performance LP_c and some learning performance measured at some reference trial r that occurred before the current trial:

$$\Delta LP = LP_c - LP_r$$

Positive learning rate is called a *Learning Progression*. Negative learning rate is called a *Learning Regression*.

Tracking and Measurement Data Structures

The progression of steps and trials, the use of prompts and prompt fading rules, the application of reinforcement schedules, and the thinning of the reinforcements are all determined by the statistical calculations over data collected before and during the instructional process. Data structures presented in this section are defined with the purpose of capturing all the data relevant to the instructional process, and for supporting the instructional agent in reasoning about the student's learning progress.

Trial record is used to capture all the relevant data about a single trial. It is a data structure that tracks the antecedent, cue, prompt (type and level), and reinforcers used in a trial, the duration of the trial, the outcome of the trial including the degree of correctness of the behavioral and temporal measures of the trial. **Session Record** is used to capture all the relevant data about the instructional session (n number of trails performed in one "sitting"). The *Session Record* data structure tracks the following aspects of the session: the session's index in the learning plan, the number of trials in the session, the number of successful trials in the session, the current trial, the setting of the session, the time and date of the session, and its duration.

Learning Objective is defined by the name/id of the behavior, the abstract behavior descriptor, and the performance goal (PG).

Measure of the Correctness of Behavior (|B|)

Earlier we classified the expression of behavior in terms of its operand value as:

- Correct Behavior (C),
- Approximately Correct Behavior (A), and
- Incorrect Behavior (I).

Although *No Behavior (N)*, depending on the learning objective, can be treated as either correct, approximate, or incorrect behavior, in some instances it might be useful to record it as an auxiliary measure (e.g., child exhibiting no behavior might signal another issue beyond the learning objective).

Degree of correctness |*B*| of behavior is defined as a difference between the expected behavior and observed behavior, where the measure of the difference is specific to the attributes of that behavior class.

This measure is a real number between 0 and 1 expressing the degree to which the attributes of the observed behavior match the attributes of the expected behavior. The exact method for establishing this measurement depends on the sub-classification and dimension of the behavior.

We are also interested in an *approximately appropriate behavior*. This approximation is determined by behavior a similarity function S, which is defined contextually for each class of behaviors.

In a general form, the behavior similarity function accepts the actual behavior description (B_a) and expected behavior description (B_e) and returns a scalar degree of similarity measure.

We define the behavior similarity function S as a mapping between the pair of behaviors of the same class $\mathbb{B}^{\mathbb{C}}$ and the interval [0,1]:

$$S : \{\mathbb{B}_{\nVdash}, \mathbb{B}_{\neq}\} \rightarrow [0, 1]$$

Behavior description is defined abstractly as a set of all behaviors $\mathbb{B}^{\mathbb{C}}$ that share the same attributes. Each sub-class of behavior will have its own attributes and associated comparison function that compares the similarity of the two behaviors that are members of the same behavior class.

Abstractly, any member b of the sub-class B^k is describable by the same set of identifying attributes A_k = $a_1, a_2, a_3, ...$.

For example, any verbal behavior is describable by two attributes: lexical correctness and duration. Therefore, we might, define the class of verbal behaviors B^v as a set of elements such that each member of the set has the same set of attributes A_v = *duration, lexical correctness*.

B^V = {v | v has *{duration, lexical correctness}* }

More complex behavior might have multiple spatial, temporal and other attributes.

3.3 Representation of States

The agent's behavior in the instructional process is partially determined by reasoning over global and local states with respect to the student and the instruction.

Global instructional states are representing the state of the student's overall learning progress and the state of the learning process itself (step in a learning goal, etc.). Local states are the states related to the instructional session - state of the student's attention, the state of the instructional session respective to the cue-prompt-response-consequence chain, and the state of the agent itself.

3.3.1 Tracking Global and Local States

The agent's behavior of the agent in the instructional process is controlled by reasoning over two levels of states — global state sets and local state sets respective to the instructional session.

Global instructional states are representing the state of the student's learning progress and the state of the learning process. Local states are related to the instructional session. They are the state of the instructor, the state of student's attention, and the state of instructional session respective to the ABC. We have chosen the finite state machine approach for the design of the instructional reasoning engine because of the ABA's significant reliance and dependence on the situation, context, and state of the student's learning of student.

Instructor's States

Instructor states help track instructor's current activity. They are useful for both internal tracking (self-awareness) of the instructor's state in the instructional process and for the visibility and awareness into the process by any external observers.

Instructor's states are (See Figure 3.9):

- Initial - instructor has not engaged the student, but it is about to engage.
- Gaining Student's Attention - instructor has started the instruction and is attempting to engage the student.
- Instructing - instructor is actively engaged in the instructional session.
- Seeking Student - the student has left the instructional setting, so the instructor is temporarily disengaged from the instructional session.
- Done Instructing - Instructor is disengaged from the instructional session.

Instructor states are formally represented as NFA by a 5-tuple, $(Q, \Sigma, \Delta_{instructor}, q_0, F)$, consisting of:

- a finite set of states Q ={q_0 (Start),q_1(Gaining Attention),q_2(Seeking Student), q_3(Instructing),q_4(Done)}
- a finite set of input symbols Σ={A (Student Inattentive), B (Student Absent),C (Student Attentive), D (Lesson Complete), E(Student Gone)}

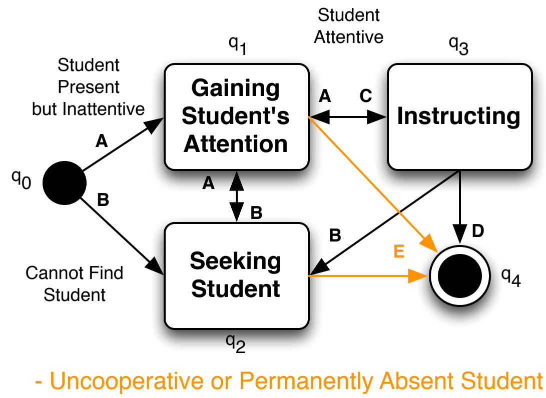

Figure 3.9: States of Instructor

- a transition relation $\Delta_{instructor} : Q \times \Sigma \rightarrow P(Q)$
- an initial state $q_0 \in Q$
- a set of states F distinguished as accepting (or final) states F $\subseteq$ Q. F=$\{q_4\}$

Transition relation $\Delta_{instructor}$ is defined with the following transition table:

Table 3.2: Transition Relations for Instructor State

	A	B	C	D	E
q_0	q_1	q_2	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	q_2	q_3	$\varnothing$	q_4
q_2	q_1	$\varnothing$	$\varnothing$	$\varnothing$	q_4
q_3	q_1	q_2	$\varnothing$	q_4	q_4
q_4	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

The State of Instruction

The state of instruction (See Figure 3.10) refers to a state of the overall lesson for the student. In the ABA instruction, the instructor may be working with the student on a new behavior through a series of steps and towards a desired performance goal. It is also possible that the instructor is running maintenance trials to establish if the student has mastered the behavior. Otherwise, the instructor may be helping the student to generalize by changing the aspects of the instruction. All these global states are important so that the instructor-agent can save them and resume the overall progress of the instruction.

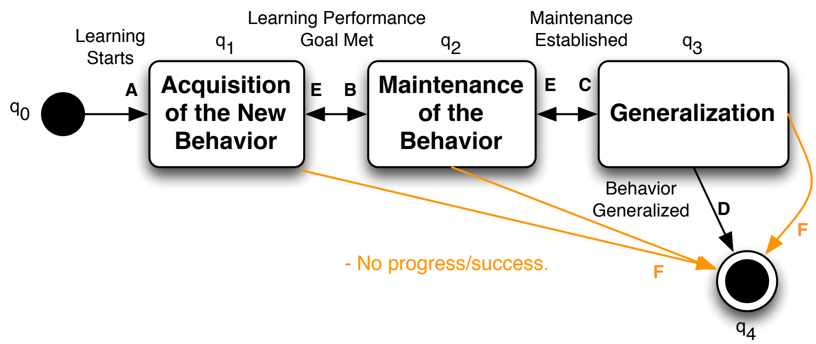

Figure 3.10: States of Overall Instruction

Instructor states are represented as NFA by a 5-tuple, $(Q, \Sigma, \Delta_{instruction}, q_0, F)$, consisting of:

- a finite set of states Q ={q_0 (Start),q_1(Acquisition of the New Behavior),q_2(Maintenance of the Behavior),q_3(Generalization),q_4(Done)}
- a finite set of input symbols Σ={A (Learning Starts), B(Learning Performance Goal Met), C (Maintenance Established),D (Behavior Generalized), E (Learning Performance Drop), F (No Progress)}
- a transition relation $\Delta_{instruction} : Q \times \Sigma \to P(Q)$
- an initial state $q_0 \in Q$
- a set of states F distinguished as accepting (or final) states F $\subseteq$ Q. F={q_4}

.

Transition relation $\Delta_{instruction}$ is defined with transition table:

Table 3.3: Transition Relations for the Global State of Instruction

	A	B	C	D	E	F
q_0	q_1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	q_4
q_1	$\varnothing$	q_2	$\varnothing$	$\varnothing$	$\varnothing$	q_4
q_2	$\varnothing$	$\varnothing$	q_3	$\varnothing$	q_1	q_4
q_3	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	q_2	q_4
q_4	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

The State of Trial

This state machine (See Figure 3.11) is central to the realization of the instructional session based on the three-term contingency (antecedent, behavior and consequence). Its purpose is tracking the state of operand conditioning in the learning session - if the setting event has been issued, if the agent is expecting a behavior, if the prompt has been issued, or if the consequence has been issued.

Trial states are represented as NFA by a 5-tuple, $(Q, \Sigma, \Delta_{trial}, q_0, F)$, consisting of:

- a finite set of states Q ={q_0 (Start),q_1(Issued Cue), q_2(Awaiting Response),q_3(Issued Prompt),q_4(Issued Consequence),q_5 (End of Trial)}
- a finite set of input symbols Σ={A (Cue), B (Prompt Wait Time Expired),C (Hesitation), D(Consequence),E(Student's Absence)}
- a transition relation $\Delta_{trial} : Q \times \Sigma \to P(Q)$
- an initial state $q_0 \in Q$
- a set of states F distinguished as accepting (or final) states F ⊆ Q. F={q_5}

Transition relation Δ_{trial} for the trial is defined with:

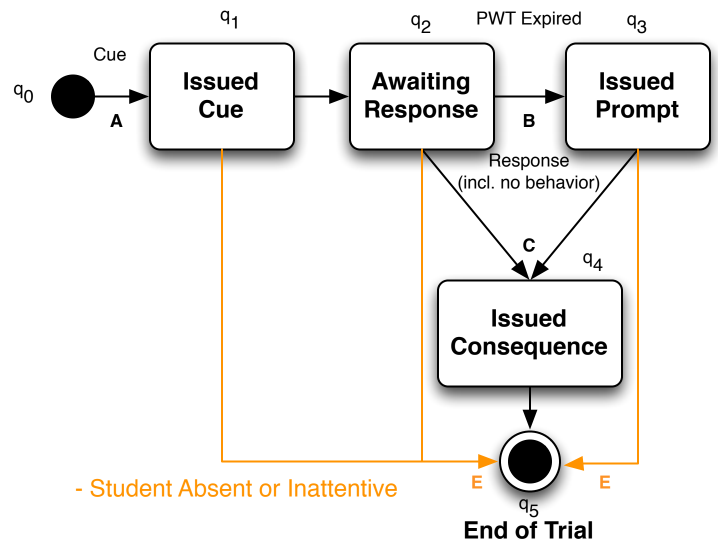

Figure 3.11: States of Discrete Trial

Table 3.4: Transition Relations for the Discrete Trial

	A	B	C	D	E
q_0	q_1	$\varnothing$	$\varnothing$	$\varnothing$	q_5
q_1	q_2	$\varnothing$	$\varnothing$	$\varnothing$	q_5
q_2	$\varnothing$	q_3	q_4	$\varnothing$	q_5
q_3	$\varnothing$	$\varnothing$	q_4	$\varnothing$	q_5
q_4	$\varnothing$	$\varnothing$	q_4	$\varnothing$	$\varnothing$

Agent's Representations of Student's State

The representation of the student is the instructional agent's internal representation of the student's state in the learning process and the student's state of attendance. We use these two state representations to track two different aspects of learning and required agent responses:

1. specific learning state - a representation of how well the student is progressing and if there is a need for the agent to change the instructional and reinforcement approach.

2. physical situation - a representation of whether the student is physically attentive and accessible by the instructional agent. The student might be completely inattentive or physically depart from the learning environment (e.g., walk away) and we want to track this situation and enable the agent to appropriately react (e.g., call out to a student or inform others).

Student State During Instruction

The state of learning (See Figure 3.12) is used to track the student's progress in the learning process and to appropriately advance and adjust the agent's instructional and corrective actions. During the learning process, the student is assumed to start in the initial ("blank slate") state when being introduced to new skills. From that state, the agent might advance to a progressing state, regressing state, or to a mastery state which is a final state for the learning task. No progress in the student's learning performance indicates the need for re-designing the lessons.

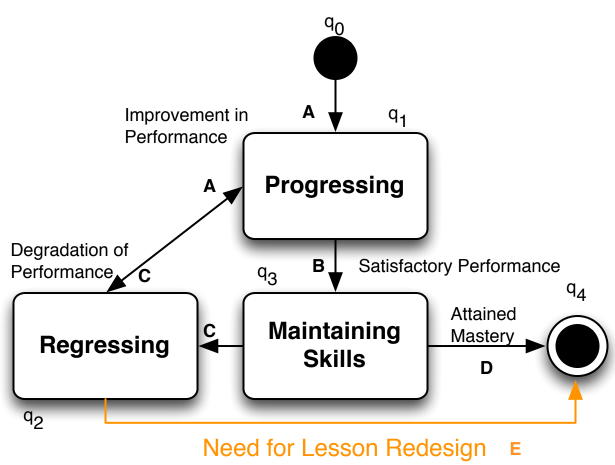

Figure 3.12: States of Student in Lesson

The student's learning states are represented as NFA by a 5-tuple, $(Q, \Sigma, \Delta_{learning}, q_0, F)$, consisting of:

- a finite set of states Q ={q_0 (Start),q_1(Progressing),q_2(Regressing),q_3(Maintaining),q_4(Mastery)}

- a finite set of input symbols Σ={A (Progress), B (Regression),C (Learning Performance Plateau), D (Skill Mastery), E (No Progress)}
- a transition relation $\Delta_{learning} : Q \times \Sigma \rightarrow P(Q)$
- an initial state $q_0 \in Q$
- a set of states F distinguished as accepting (or final) states F $\subseteq$ Q. F={q_4}

Transition relation $\Delta_{learning}$ for the student's learning is defined with:

Table 3.5: Transition Relations for Student's Learning States

	A	B	C	D	E
q_0	q_1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	q_3	q_2	$\varnothing$	$\varnothing$
q_2	q_1	$\varnothing$	$\varnothing$	$\varnothing$	q_4
q_3	$\varnothing$	$\varnothing$	q_2	q_4	$\varnothing$
q_4	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Student State During Instructional Session

The state of attendance (See Figure 3.13) represents the student's state of attentiveness during the trial. This state machine is constructed to track and account for the student's attentiveness to the learning process - physical presence, behavioral, and cognitive attentiveness (bound by observational ability) to the lesson. The instructional agent relies on this machine to know when to intervene and call out to the student once he or she is determined to be inattentive or the student leaves the lesson.

The student's states in a trial are represented as NFA by a 5-tuple, $(Q, \Sigma, \Delta_{attention}, q_0, F)$, consisting of:

- a finite set of states Q ={q_0 (Start),q_1(Attentive),q_2(Inattentive), q_3(Absent),q_4(Responding),q_5 Done)}
- a finite set of input symbols Σ={A (Attentiveness), B (Inattentiveness),C (Absence),D(Instruction),E(Responded), F(Permanent Absence)}

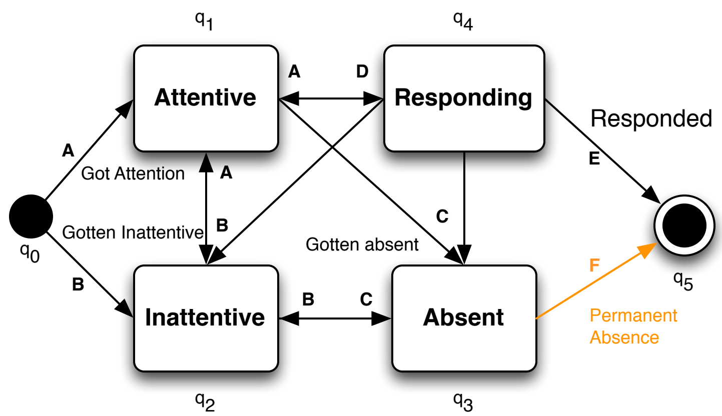

Figure 3.13: States of Student in Trial

- a transition relation $\Delta_{learning} : Q \times \Sigma \rightarrow P(Q)$
- an initial state $q_0 \in Q$
- a set of states F distinguished as accepting (or final) states F $\subseteq$ Q. F=$\{q_5\}$

Transition relation $\Delta_{attention}$ for the student's learning is defined with:

Table 3.6: Transition Relations for Student's States of Attention

	A	B	C	D	E	F
q_0	q_1	q_2	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	q_2	q_3	q_4	$\varnothing$	$\varnothing$
q_2	q_1	$\varnothing$	q_3	$\varnothing$	$\varnothing$	$\varnothing$
q_3	$\varnothing$	q_2	$\varnothing$	$\varnothing$	$\varnothing$	q_5
q_4	q_1	q_2	q_3	$\varnothing$	q_5	$\varnothing$

3.4 Instructional Procedures

As part of the ABA ontology we also define the instructional procedure as a set of abstract instructional scripts. The abstract scripts are high level algorithms defined independently of any specific technology and implementation concerns. The instructional script is a series of instructional steps consisting of the interactions with the student with the purpose of eliciting and rewarding the student's learning of new behaviors or the modification of the existing ones.

We use *discrete trial* (See Figure 3.14) as a foundation for the implementation of the agent's instructional capabilities.

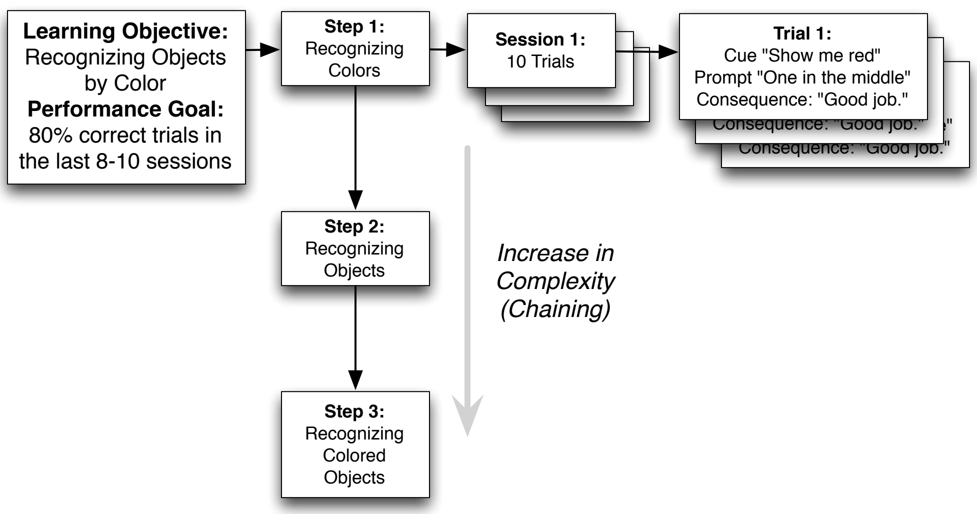

Figure 3.14: Discrete Trial Training Process

3.4.1 Global Instructional Procedures (GIP)

Global procedures govern the execution of the entire lesson plan with the students and the global learning instructional methods such as step progression, prompt fading, generalization, application of appropriate schedules of reinforcement, and the thinning of reinforcers.

Conduct Lesson

Conduct Lesson (See Algorithm 3.4.1) is a top-most procedure that governs the executions of the sessions.

Algorithm 3.4.1: conduct_lesson($lesson, student$)

for each $step$ **in** $lesson$
do {
  for each $session$ **in** $step$
  do {
    $conduct\ session$
    if $learning\ performance\ improves\ and\ if\ change\ criteria\ applies$
    then {
      $reduce\ prompt\ level$
      $change\ reinforcement\ schedule(thinning)$
      if $change\ cannot\ be\ made$
      then {$change\ instructional\ setting$
    else $increase\ prompt\ level$
    $compress\ reinforcement\ schedule$
    if $change\ cannot\ be\ made$
    {$stop\ instruction$

3.4.2 Session Instructional Procedures (STP)

Session Procedure (*conduct session*)

Session procedure (see Algorithm 3.4.2) consists of an execution of all trials in the session.

Algorithm 3.4.2: conduct_session($session, student$)

for each $trial$ **in** $session$
do {
  $set\ preferred\ stimuli$
  $set\ preferred\ prompt$
  $get\ student's\ attention$
  $conduct\ trial$

The session might terminate early if the student leaves the learning environment permanently, or if his or her conditions do not permit the continuation.

Trial Procedure (*conduct trial*)

Trial procedure (see Algorithm 3.4.2) closely resembles the procedure of the educational Discrete Trial. It consists of sub-procedures for the issuance of the cue, prompt, and consequence. Prompts are issued on the expiration of a pre-defined *Prompt Wait Time (PWT)*.

Algorithm 3.4.3: perform_trial($cue, prompt, schedule, reward, correction, student$)

$get\ cue\ from\ the\ trial\ specification$
$issue\ cue$
$wait\ for\ response$
if $no\ response\ within$ *Prompt Wait Time (PWT)*
$$\textbf{then}\ \begin{cases} get\ prompt\ from\ trial\ specification\\ issue\ prompt\ to\ student\\ record\ prompting \end{cases}$$
if $behavior\ is\ expected$
$$\textbf{then}\ \begin{cases} issue\ reward\ \ to\ student\\ record\ success \end{cases}$$
$$\textbf{else}\ \begin{cases} issue\ correction\ \ to\ student\\ record\ failure \end{cases}$$

3.4.3 Global Control Procedures (GCP)

Global control procedures govern the agent's interaction with the environment, observations about the placements of the objects, expressions of behavior, calling and observing the student's attendance and behavior. The three main control procedures are:

- Presence determination procedure establishes if the student is physically present within an instructional area. Input to this procedure is the environmental observation of the student in the environment and the output is the update to the state of the student.

- Attention determination procedure establishes if the student is attentive to the instructional process. Input to this procedure is the observation of the student's activity and the output is the update to the state of the student.

- Progress determination procedure establishes if the student is making progress. Input to this procedure is the data collected in the lesson and session record and the output is the learning performance measure.

Chapter 4

The Instruction Control Architecture

The ABA Ontology formalizes the conceptual, computational, and representational aspects of the ABA; the instruction control architecture formalizes the executive, reasoning, and operational elements of the overall agent system. In the words of Bass et al. (2003):

> The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them (p. 3).

The architecture of the architectural agent is also a connecting and coordinating component between the computation happening within the agent and the environment where the instruction is situated.

4.1 Design and Development of the Architecture

Hofmeister et al. (2007) define the formal process for the development of the software architectures. This process consists of the following four phases: (i) architectural analysis, (ii) architectural synthesis (design), (iii) architecture evaluation, and (iv) architecture evolution. The purpose of this process is to rigorously examine the requirements and functions that the system has to support and to foster the creation of the blueprint for an efficient and complete solution of the identified problem. The same four-phase process serves as a procedural foundation for design and development of the Procedural-Reasoning Architecture for Applied Behavior Analysis-based instruction (PRA-ABA). The outcomes of that process are presented in the following sections.

4.2 Architectural Analysis

Architectural analysis is the process understanding of the context, requirements, and the operating environment for the future system. The main outcome of this phase is the collection of essential requirements called *Architecturally Significant Requirements (ASRs)* (De Boer and Van Vliet, 2009) that the future architecture has to support.

As part of the process of architectural analysis and development of ASRs, we examined the theory of the Applied Behavior Analysis, its ontological formalization, the state of teaching practice, and the state-of-the-art in the applications of intelligent agents in the special education domain. We used certification material from the Behavior Analyst Certification Board (2012), and we reviewed and prioritized the requirements in collaboration with Doctoral-level Board Certified Behavior Analyst (BCBA-D) (The BCBA Board of Directors, 2013). We synthesized the resulting requirements for the system and its reasoning component into a set of competency questions (Grüninger and Fox, 1995). We used the competency questions as both requirements for the system and, later, as the input and quality metrics for the evaluation (Chapter 5).

4.2.1 Architecturally Significant Requirements

We synthesized our analysis into the following requirements for the features and functionality of the instructional agent-based system.
The Agent-based instructional system should be capable to:

- execute the execution of the main components of the behavioral instruction (three-term contingency, discrete trial, session, and lesson) in an appropriate order,

- infer the student's learning progress in an ABA-based instruction, i.e., the agent should "know" if the student is progressing, regressing, or struggling with particular aspects of the instruction,
- apply the appropriate behavioral instructional methods: issuance of cues, application of prompts, prompt fading, issuance of the consequences, promotion, and recognition of the generalization,
- infer student's state (presence, attentiveness) and learning performance within the instructional session.

In addition, the architecture should support the overall system with the abilities to:

- control the execution of the instruction in the mixed modality (virtual, embodied, etc.),
- pause and resume the instruction,
- modify the execution based on the learning progress of the student, and
- alter the stimuli and behaviors used in the instruction (cues, prompts, consequences).

4.2.2 Agent-specific Characteristics of the Architecture

In their book *Computational Intelligence*, Poole and Mackworth (Poole and Mackworth, 2010, p.14) define three aspects of computation that constitute the intelligent agent's reasoning framework:

- the computation that is part of the design of the agent,
- the computation that the agent does before it observes the world and needs to act, and
- the computation that is done by the agent while it is acting and interacting with the world.

In PRA-ABA, these aspects are reflected as:

- a set of rules and scripts based on the guiding principles and rules of ABA as represented in the ABA ontology that are part of the agent design,
- a computation that agent performs to select the appropriate instructional policy given the learner's needs and progress in the learning process, and
- a reasoning and selection of the appropriate sequence of behavioral instructions while conducting the discrete trial.

4.2.3 The Dimensions of Implementation Complexity

Another analytic device adopted from Poole and Mackworth (2010) was a nine-dimension taxonomy for the categorization of intelligent agents. We used this taxonomy to further define the characteristics of the emerging architecture.

1. The *Modularity* dimension qualifies the architecture of the system, and how it can be decomposed into interacting modules that can be understood separately - flat, modular, hierarchical. The design of the PRA-ABA is modular, consisting of the modules relating to the instructional controls, execution, reasoning, and knowledge persistence.

2. The *Representation Scheme* specifies how the world is described. Agents typically reason in terms of states, features, relational descriptions, or in terms of individuals and relations. The PRA-ABA representational emphasis is on the states, actions, and percepts that alter these.

3. The *Planning Horizon* dimension examines how far the agent "looks" into the future when deciding how to act. Within the planning horizon dimension, an agent can be a non-planning agent, finite, one with an indefinite, or one with an infinite planning horizon. PRA-ABA design is for a finite planning agent that works on predetermined lesson plans (scripts) with finite terminating conditions.

4. The *Uncertainty* dimension examines how uncertain the agent is about its environment and its own action. This dimension is therefore sub-divided into two sub-dimensions:

   - The *Sensing Uncertainty* examines if the agent can determine the state of the world from the observations as fully observable or partially observable. For the scope of the research and practicality of the implementation, the design assumes full observability. However, we recommend that future research should explore partial observability.
   - The *Effect Uncertainty* dimension examines if, given the state of the world and the agent's action, the agent can accurately predict the state of the world resulting from its actions. Taxonomy specifies that the effects of the agent's actions can be *deterministic* or *stochastic*. In the PRA-ABA design, the effects of the agent's actions are generally deterministic, although our design accounts for stochastic events (e.g. randomness of the student's attention and responses).

5. The *Preference* dimension is about the agent's preference and to which degree its actions are driven by some desirable outcomes. The preference dimension examines whether an agent has: (i) goals, which can be achievement or maintenance goals, or (ii) complex preferences, which can be ordinal or cardinal. The PRA-ABA system design focuses primarily on learning achievement goals and considers both kinds of complex preferences, with greater focus on the ordinal preferences.

6. The *Number of Agents* examines whether the agent system design is for a single or multi-agent system. The PRA-ABA is a single-agent system interacting only with one human subject.

7. The *Learning* dimension examines whether the knowledge of the world is given or whether it is learned. In the PRA-ABA design, knowledge is both learned and given.

8. The *Computational Limits* dimension examines whether an agent has *perfect rationality* - the agent reasons without taking into account any constraints imposed by

limitations of computational resources, or *bounded rationality* - the agent reasons by taking into consideration computational limitations. The PRA-ABA design assumes perfect rationality given the imposed restrictions of the problem domain (see *The Dimensions of ABA* in Chapter 1).

9. The *Interaction of Dimensions* dimension examines the degree of interaction and the impact between the eight other dimensions of implementation. This dimension also examines the complexity of these relationships. The dimensions in PRA-ABA are inter-related although the relationship is relatively straightforward. Table 4.1 outlines the analysis of the PRA-ABA dimensions according to Pool and Mackworth's taxonomy.

Table 4.1: The Dimensions of Complexity

Pool and Mackworth Dimension	PRA-ABA Characteristic
Modularity	Modular
Representation Scheme	States and Actions
Planning Horizon	Finite Planning
Uncertainty	
Sensing	Full observability for sensing uncertainty
Action Effect	Mixed deterministic and stochastic on action effect uncertainty
Preference:	
Goals	Learning achievement goals
Complex Preferences	Both ordinal and cardinal preferences
Number of Agents	Single
Learning	Knowledge is initially provided, but agent learns about student's preferences and learning performance
Computational limits	Perfect rationality
Interaction of Dimensions	Generally interactive

4.2.4 Agent Inputs

An intelligent agent is an interactive entity that operates in its environment by receiving the external inputs (observations), examining these inputs against the internal inputs (prior knowledge, history, states, procedures, etc.), and issuing actions back onto the environment. The PRA-ABA instructional agent operates on the following inputs:

- **prior knowledge** consisting of the knowledge about the instructional process, student preferences, setting, and the agent itself (appearance);
- **history** of agent's interaction with the instructional environment, consisting of:
  - **observations** about the current environment, and
  - **previous actions, recordings, and data**;
- **goals** represented as a lesson plan consisting of a sequence of learning steps and the student's target performance criteria; and
- **abilities** - the agent's ability to operate in the environment, interact with the student, emit behaviors, and reason about the instructional process.

Table 4.2 outlines the analysis of the PRA-ABA characteristics according to Russell and Norvig (2010) *PAGE* (Percepts, Actions, Goals, Environment) taxonomy for classifying intelligent agents. *PAGE* is another well-established scheme for analysis of the properties and classification of the intelligent agents:

Table 4.2: The PAGE Characteristics of PRA-ABA

Percepts	Actions	Goals	Environment
Behaviors	Stimuli	Learning New Behavior	Instructional Environment
Movements	Reinforcers	Target Learning Performance (LP)	Student
Objects	Prompts		Objects
Other stimuli	Objects Placements		
	Movements (for embodied agent)		

In summary, the PRA-ABA agent has to satisfy two main functions: (i) execution of the instruction in the environment and with a student, and (ii) inference of the student's progress, preferences, and dynamic adjustments to the behavioral aspects of the instructional process.

To realize these two functions, we proposed an approach where the PRA-ABA architecture is structured as a union of the three main groups of components dedicated to: (i) control of the execution of the instruction and (ii) reasoning about student's progress and preferences.

4.3 Architectural Synthesis

The Architectural synthesis is the process of translating the findings of the architectural analysis into a design for the future system. The architectural design is driven by the ASRs identified during the architectural analysis and its main objective is to fully realize all of these requirements. Furthermore, the architectural synthesis establishes the core technical principles that should drive all of specific technical implementations of the system. In the case of PRA-ABA, the three main architectural principles that were identified during the synthesis process were: (i) the PRA-ABA is a procedural architecture, (ii) its design relies on the abstraction hypothesis, and (iii) it requires elementary (procedural) reasoning.

4.3.1 Abstraction Hypothesis

Architectural abstraction (Jennings, 2001) is the fundamental assumption of the PRA-ABA's design - i.e., the design of the overall system is based on the assumption that an instructional control system, including procedural, ontologic, and reasoning elements can be designed while abstracting away the locomotoric, spatial, and environmental specifics. Other agent-based initiatives such as Virtual Human (VH) (Reidsma et al., 2011), BML (Vilhjálmsson et al., 2007), and MPML (Prendinger et al., 2004), have demonstrated the viability of the abstraction of machine-generated and interpreted the multi-modal behavior and the implementation of the abstract control system that orchestrates them.

Adhering to this assumption, we identified following essential, high-level abstractions (Shaw et al., 1995) for the proposed control architecture:

- Percepts - These are the representations for all external stimuli serving as inputs to the control architecture. Percepts were abstracted as external inputs to the system consisting of the student's posture and behavior descriptors; descriptors of the student's and agent's position and their presence in the instructional setting; and the information about the setting itself.

- Agent program - This is a "driver" for all sequential control, reasoning, and knowledge retrieval or update routines. The agent program is responsible for the execution of the instructional steps in the appropriate order or for the termination of the instruction. The agent program is also responsible for appropriately coordinating the actions of the other components of the architecture.

- Reasoning components - These are the elements of the overall architecture that can infer the student's state of progress in the instruction, and dynamically recommend the appropriate adjustments to the execution of the instruction (e.g., prompting, prompt fading, thinning of the reinforcers).

- Actions - These are the high level behavioral instructions issued to the physical peripherals of the agent platform. These are abstract descriptions of the agent's behavior and issuance of the stimuli. The actions are issued as consequences of the instruction progress, the state of the instruction, and the state of the student in the instruction and the environment.

4.4 The Reasoning Components of the Architecture - Beliefs, Desires, and Intentions (BDI) Model

We have chosen the *Belief-Desire-Intention (BDI)* (Rao and Georgeff, 1991) agent model as a high-level reasoning framework for the formalization of procedurally-oriented reasoning aspects of the PRA-ABA instruction. The structure of the BDI model (Figure 4.1) logically separates the agent's reasoning characteristics into three major components: (i) agent's beliefs about the state of the world (instruction), (ii) agent's desires or the objectives of the instruction, and (iii) intentions which are, in the context of ABA-specific instructional scripts and actions. A BDI model is a basic model for representing the reasoning capabilities of the rational-agent. In this rudimentary form, it has a limited support for the representation of planning, online search, and for other advanced agent capabilities (Busoniu et al., 2008). However, for the PRA-ABA architecture, BDI offers the appropriate and complete conceptual formalization given the prescriptive nature of the ABA-based instruction.

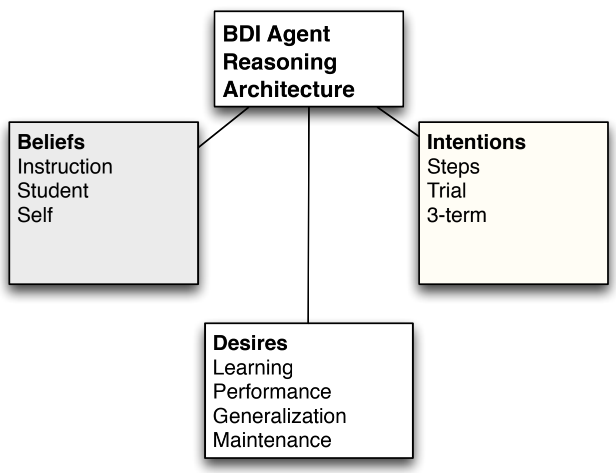

Figure 4.1: BDI Architecture

4.4.1 Beliefs

The *beliefs* are the agent's internal representations of its understanding of the world. In the PRA-ABA architecture, beliefs are representations of the states that were conceptually defined in the ABA ontology.

PRA-ABA defines three groups of beliefs: beliefs about the state of the instruction, beliefs about the state of a student in the instruction, and the agent's beliefs about itself.

Each one of these beliefs have a global (instruction-level) and immediate (trial-level) context.

The states of the agent's beliefs are managed by the instructional reasoner component which is reviewed later in the Chapter 4.9.

Global Beliefs about the Student

These beliefs represent the agent's understanding of the student's learning progress and his or her state of learning at a global level of instruction:

- Progressing (P) - A student is progressing in the current learning step;
- Regressing (R) - A student is regressing in the current learning step;
- Maintaining (M) - A student is maintaining knowledge;
- Generalizing (G) -A student is in the process of generalizing a learned skill; and
- Finished (F) - A student has finished the learning process.

Immediate Beliefs about the Student's State of Attention

The immediate beliefs represent the agent's understanding of the student's state in the immediate instructional session. They reflect the student's mental and physical presence and participation in the process:

- Inattentive (I) - the student is not attentive;
- Absent (ABS) - the student is absent;
- Attentive (AS) - the student is attentive; and
- Finished (FS)- the student is finished with a session.

Immediate Beliefs about the Student's State at the Trial Level

The immediate beliefs at a trial level represent the agent's understanding of the student's state of responding in the discrete trial.

- Responding (R) - the student is about to respond;
- Hesitating (H) - the student is hesitating to respond;
- Correct (C) - the student responded correctly;
- Incorrect (I) - the student responded incorrectly;
- Done (DT) - the student is done with the trial.

Beliefs about Self at the Global Level

The beliefs about self represent the agent's understanding of its own state in the instructional process. These beliefs play an essential role in the agent's ability to sequence the instruction, maintain the student's attention, and to restart the instruction, if interrupted. The agent's global beliefs reflect the agent's understanding of its own state in the overall process:

- Gaining Student's Attention (GA) - the agent is trying to gain the student's attention in order to commence the learning process;
- Seeking Student (SS) - the student is absent, and the agent is trying to find the student;
- Instructing (AI) - the agent is engaged instructing; and
- Finished (FI) - the agent is finished with the process.

Beliefs about Self in the Instruction

These beliefs represent the agent's understanding of its own state related to the conduct of the behavioral instruction in the trial:

- Issued Cue (IC) - the agent has issued a cue;
- Awaiting Response (AR) - the agent is awaiting the student's response;
- Issued Prompt (IP) - the student has responded correctly (C);
- Issued Consequence (IC) - the agent has issued a consequence; and
- Finished (FT) - the agent is done with the trial.

4.4.2 Desires

Desires are instructional objectives that the agent wants to accomplish. They are expressed as learning performance levels that the agent is helping the student to accomplish:

- target learning performance (TLP) is a level of learning performance (LP) that the student should reach on the task;
- maintenance of the skill (SM) is defined as an acceptable variance of the student's learning performance within some predetermined control limits (MacGregor and Kourti, 1995);
- overall learning objective (LO), and
- generalization (G) of the student's learning performance across different learning settings.

Desires are encoded within a reasoner and stored in the knowledge base as either control limits of a control chart for SM and G or as a target measure for TLP and LO bounded by halting conditions. The halting condition is defined as the maximal number of trials to be performed if the performance is not met, and its purpose is to prevent the "infinite loop" problem.

4.4.3 Intentions

Intentions are specific steps that the instructional agent is going to take in order to accomplish or maintain certain goals. Intentions are encoded as a set of actions to be performed in order to accomplish the goal. The *agent program*, which we discuss in the next section, is responsible for the selection, sequencing, and dynamic re-ordering of the intentions. Following are the intentions defined within PRA-ABA architecture:

- Teach Learning Step (TLS) - the agent will teach n sessions until the student's target learning performance is met (TLP);
- Draw Attention (DA) - the agent will execute attention drawing actions until the agent has the student's attention. If the initial action does not produce the expected result (student's attention), alternate different attention drawing actions until the agent has the student's attention. Terminate the instructional procedure if the agent cannot get student's attention within n attempts. (n is a configurable parameter);
- Teach Session (TS) - the agent will teach n discrete trials t; and
- Teach Trial - the agent will execute the trial script t (consisting of a typical cue/prompt/response/consequence structure).

Intention scripts, actions, and related desires are defined and stored within the knowledge base as executable, parametrized scripts.

4.5 Percepts

A *percept* is a general concept representing any external input received by the agent system. In PRA-ABA, the percepts are abstract. They are represented in terms of concepts defined in the ABA ontology (behaviors and stimuli). This approach is taken in support of the abstraction hypothesis, and in order to make the essence of the architecture and its functions portable across different modalities. In different implementations, modality-specific interpreters translate the low-level percepts (signals, sensor readings, etc.) into percepts understood by the architecture. For example, in a virtual context, a percept interpreter implemented using Kinect SDK (Lai et al., 2012), might interpret Kinect-observed face and eye tracking input indicating the continuous gaze as attention (Oikonomidis et al., 2011); pointing motion at the object would be interpreted as the behavior (Chang et al., 2011).

4.5.1 Instructional Percepts

Instructional percepts are any percepts that describe the states and the instruction process:

- the student's responses (behaviors) which are the percepts that represent the student's vocal, verbal, gestural, physical, and other behaviors;
- the student's attentiveness which are the abstract indicators of the fact that the student is maintaining the attention to the instructional process. They are the translations of the common physical indicators of attention: maintenance of the eye gaze, presence, attentive posture, etc.; and
- objects and stimuli, which are the percepts describing the placement of the objects or stimuli used in the behavioral instruction as cues, prompts, distractors, or reinforcers.

4.5.2 Environmental Percepts

These are the percepts that describe the physical properties of the instructional environment such as the features of the environment and the locations and placement of the non-instructional objects and stimuli.

4.6 Actions

Actions are high-level, abstract instructional or instruction-related commands issued by the agent controller. They are translated into modality specific steps such as sounds, movements, or other simulated or actuator-based commands. Actions can take parameters and return the results. (Some actions do not return values; they run as procedures.)

4.6.1 Environmental Actions

Environmental actions are actions that in some way change the state of the instructional environment. The two environmental actions defined by PRA-ABA are:

- *Place*, the action that the agent uses to place the object into the environment. Its purpose is to place the objects with instructional purpose (object of instruction or a distractor) within the instructional setting, and is defined by the *location* variable. The location is a three-dimensional coordinate for the placement of the object. The syntax for the action is Place(object, location){};

- *Move*, the action that the agent uses to move the object within the environment. The syntax for the action is Move(object, old_location, new_location){}; the *polymorphic* version of this action is *Move(self, location){}* for the agents own movement.

4.6.2 Non-instructional Actions

Non-instructional actions are actions that are not directly related to the behavioral intervention. Their purpose is to support the preparation and effective execution of the instruction. The non-instructional actions are:

- *Gain Attention* of the student (GA). This action is issued to the student. It involves issuing a stimuli or a behavior that is known to help gain the student's attention (sound, voice, gesture). The syntax for the action is *GainAttention(student,prompt){has attention,no attention}*. The action returns the outcome of its execution. The outcome indicator is used by the agent's program controller to decide on the next suitable action (i.e. to proceed with the instruction, or to try again to gain attention).

- *Look for Student* (LS). This action is issued when the student moves out of the immediate instructional setting, and he/she cannot be found (i.e., student wanders off). This is an abstract action whose implementation depends on the modality of the agent system implementation. Its syntax is *LookForStudent(student):{found,not found}*. The action returns the indicator for the outcome of the action. This indicator is examined by the program control to determine the next action and the state of instruction.

- *Exit Instruction* (EI). This is the action that the agent performs at the end of the instruction. It includes any close-out, data recording and reporting tasks. The syntax for the action is *ExitInstruction():{success,failure}*. The action returns an indicator of whether the overall instruction was a success or a failure.

4.6.3 Instructional Actions

Instructional actions are actions that directly relate to the actions otherwise performed by the human instructor in the ABA/DTT-based process. They are implementations of the three-term contingency steps:

- *Issue Cue* (IC). This action issues a cue (Antecedent (A), Discriminative Stimuli (S^D)) to the student. This action is procedural and does not return any value. The action takes *cue* and a student identifier as parameters. The syntax for the action is *IssueCue(cue,student):{}*.

- *Issue Prompt* (IP). This action issues a prompt contingent on no response within time t. The action takes as parameters prompt and a time delay value; it does not return any value. The syntax is *IssuePrompt(prompt,t):{}*.

- *Issue Consequence* (IC). This action issues a consequence for the student's response. It takes a consequence as parameter and does not return any value. The consequence is abstractly defined - it can be a reinforcing or a correcting consequence. The syntax for the action is *IssueConsequence(consequence):{}*.

4.6.4 Virtual Actions

These actions are only available in a virtual setting. Virtual modality allows for the flexibility and ease of changing of the environment, the setting, the layout, and even the appearance of the instructor conducting the instruction. This flexibility of the virtual modality is particularly suitable for promotion of the generalization in learning. The student can be introduced easily and gradually into the new instructional setting. If the generalization does not happen, changes can easily be rolled back. Actions available in this modality are:

- *Alternate Setting* (ALS). This action enables changing of some or all of the features of the instructional setting - the environment, objects and the appearance of the instructor. It is applied for students with maximal learning performance (LP). The syntax for the action is *AlternateSetting(old setting, new setting):{}*.

- *Alternate Environment* (ALE). This action allows for the change in the instructional environment (time of the day, surrounding objects). The syntax for the action is *AlternateEnvironment(old environment, new environment){}*.

- *Alternate Instructor* (ALI). With this action, the agent can alter its appearance (sex, age, race,voice,features,clothing) which, under the right circumstances, helps the generalization of learning. The syntax for the actions is *AlternateInstructor(old instructor, new instructor){}*.

- *Alternate Stimuli* (ALS). Stimuli are used by the instructor as a teaching cues, prompts, or distractors. Alternating stimuli allows student to generalize behaviors to different stimuli. The syntax for the action is *AlternateStimuli([role], old stimuli, new stimuli){}*. The *role* parameter, which is optional, indicates if the stimulus is in a role of a cue, prompt, or something else.

4.7 Instructional Reasoning Components

Instructional reasoning components of the PRA-ABA manage the dynamic aspects of the behavioral instruction:

- applications and changes to the use of stimuli based on the preference, ability, and learning performance of the student, and
- sequencing of the generalizing and independence-promoting actions (prompt fading, chaining, thinning of the reinforcements) based on student's learning plan, and his or her abilities.

The component driving this process is an instructional statistical reasoner operating over a hierarchy of state machines (representing beliefs), instructional rules and the recordings about the instructional session.

4.7.1 Instructional Reasoner (IR)

The responsibility of the *IR* is to track the learning progress of the student, to infer his or her learning abilities and preferences, and to issue appropriate adjustments to the key aspects of the instruction (the layout, stimuli, the schedules of the reinforcement). To accomplish this, the IR maintains the internal representation of the learning state and it uses statistical process control structures (Bakker et al., 2008) to monitor the state of student's learning. IR adjusts the progress and the elements of the instruction based on the associated statistical process control rules (Gülbay and Kahraman, 2006). Table 4.3 outlines the mapping between the Nelson (1992) rules and the actions inferred by the instructional reasoner.

These rules are established individually for each student and they cover changes in the learning progress (progress, regress), the continuing lack of progress (plateau), and the monitoring of the generalization.

The *Instructional Reasoner* impacts the progress of the instruction by issuing changes to the states and by issuing actions, where and when appropriate. Internally, IR executes

Table 4.3: Nelson (1992) Rules for Inference of States and Actions

Rule	Description	PRA-ABA Interpretation
1	One observation is three or more standard deviations above or below the mean.	There are out of control behaviors.
2	Nine or more observations in a row are on the same side of the mean.	There is a prolonged bias as either under-performance or over-performance.
3	Six or more observations, in a row, are continually increasing or decreasing.	There is a trend of learning progression or a learning regression.
4	Fourteen or more observations in a row alternate in direction, increasing then decreasing.	This oscillation in performance indicates instability in learning. Changes to the instruction are needed.
5	Two or three, out of three observations in a row, are more than two standard deviations from the mean in the same direction.	There is a mild tendency for behaviors to be somewhat out of control.
6	Four or five, out of five observations, in a row are more than one standard deviation from the mean in the same direction.	There is a strong tendency for behaviors to be slightly out of control.
7	Fifteen observations in a row are all within one standard deviation of the mean on either side of the mean.	Learning is at the plateau. Depending on the objective, this is indicator of, maintenance or lack of no-progress.
8	Eight observations, in a row, are all outside of one standard deviation from the mean, and they are in both directions from the mean.	This learning pattern shows lack of maintenance or stability of the performance.

reasoning functions that take as an input events from the percept interpreter and the internal state of belief(s) and return appropriate actions or changes to the states.

4.7.2 Belief State Transition and Command Functions

Belief state transition and command functions are mappings between the current state of instruction and percepts, and the next state of the instruction or the command actions, respectively.

The agent's states are changed by a belief state transition function called *remember*. It is a function from a set of all of the agent's belief states S and percepts P onto a set of states S:

$$\textit{remember}: S \times P \to S$$

The mapping criterion for the state transition is defined by the rules in the knowledge base. These rules are dynamically parametrized by control rules inferred by the IR.

A belief state transition function for discrete time t is a function from the set of all agent's belief states S and the set of possible percepts P onto a new state. State s_{t+1} is a resulting state at time $t+1$ of the function *remember*, resulting from an application of the function at some time t to the percept p_t and the state s_t (p and s being current as of time t):

$$s_{t+1} = \textit{remember}(s_t, p_t).$$

An $action$ function is a function of the agent controller that returns an action matching the current state and a percept. It is a mapping between all of the agent's belief states S and the set of percepts P, and the set of all of the agent's actions A:

$$do : S \times P \to A$$

The mapping criteron for the command function is defined by the rules sourced from the knowledge base. These rules are dynamically parametrized by control rules inferred by the IR.

Operant Machine - A Hierarchical State Machine (HSM) for Behavior Instruction

Given the complexity and number of states in the PRA-ABA, IR operates on the global representation of beliefs called Hierarchical State Machine (HSM). HSMs (Alur et al., 1999; Keating, 2011) are convenience devices that are structurally and logically equivalent to multi-state representations, but are represented in a form that is externally easier to comprehend. In PRA-ABA, IR conceptually reasons over an HSM called *operant machine*

(Figure 4.2). The Operand machine consists of states called superstates. Each superstate is a logical grouping of the machine states collectively representing the overall state of instruction.

The IR operates on five superstates: (i) initial (IN), (ii) instructing (I), (iii) adjusting instruction (A), (iv) interrupted (IR), and (v) finished (F).

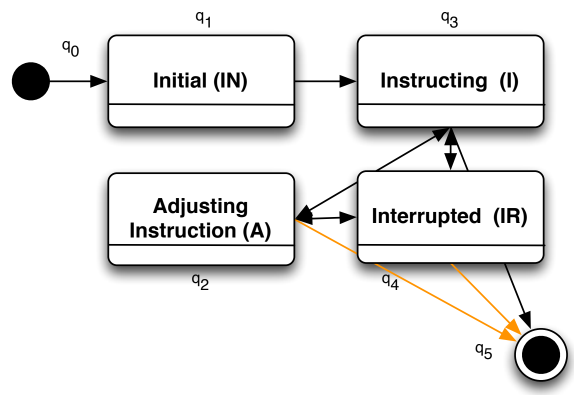

Figure 4.2: Operant Machine - Hierarchical State Machine (HSM)

The *Initial* state (IN) is a bootstrapping state of the agent. It consists of the state of instruction, the state of agent, and the state of the student in instruction from the knowledge base (KB). Once these states are loaded from the KB and the student is attentive, the agent program transitions to the *Instructing* state. In the instructing state, the agent is conducting a session. Upon the completion of the session, the agent will assess, and, if appropriate, adjust the layout of the instruction or some other aspects of the instruction (prompting, schedules of reinforcement, etc.) This superstate is called an *Adjusting Instruction* state. If all of the instructional objectives are accomplished, or if the instruction needs to be re-designed, the agent will transition to a *Finished* state. If at any point, the student or instruction requires a pause or it cannot proceed, the HSM will transition to an *Interrupted* state. From an interrupted state, the agent can resume its previous state, or, if instruction cannot proceed, it will transition to a *Finished* state while recording the permanently interrupted condition to a knowledge base.

4.8 Knowledge Base

In the BDI context, *Knowledge Base (KB)* serves as a repository for the rules that are used by the state transition and command functions. These rules are simple expressions involving the invocation of the action with a simple condition. Conditions are supplied by the IR. In PRA-ABA, *Knowledge Base* plays a bigger role than in a typical BDI/PRS knowledge base. The details of the structure of the KB and the expanded role of the KB in PRA-ABA are discussed in more detail in the next section.

4.9 Control Components of the Architecture

Control components of the PRA-ABA architecture are the executive and coordinating components that support the execution and operation of the agent as a system. Their function (Figure 4.3) is to (i) execute all the functions of the agent in a coordinated, efficient, and appropriate manner and (ii) coordinate the interaction of the agent within an instructional environment.

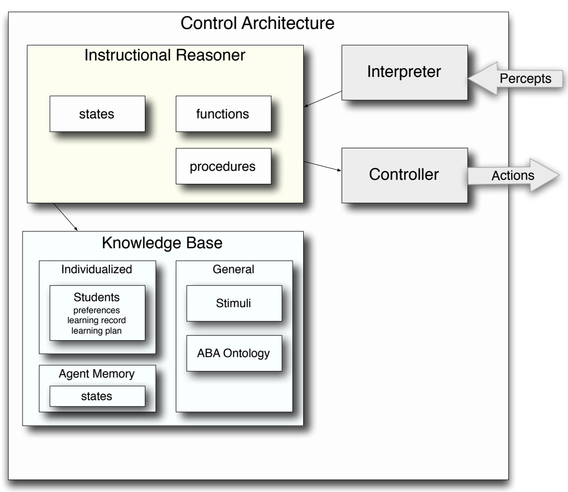

Figure 4.3: Components of the Control Architecture

4.9.1 Agent Program

The *agent program* is the “driver” component of the architecture. It is a set of procedures that coordinate the functioning of the agent as an instructional entity. The interpreter interacts with the student and the instructional environment via the *percept interpreter*. The *percept interpreter* interprets the environmental stimuli and translates them into percepts understandable by the program (behaviors, observations about the presence, and attentiveness of the student). The agent program (Figure 4.4) evaluates these percepts against the state of instruction (current procedure), the global and current state of a student, the rules of the ABA (*Knowledge Areas* and *Goals* stored in a *Database*), and issues the appropriate actions (behavior, stimuli) according to the inferred next steps (*Intentions*), and the personal preferences of the student.

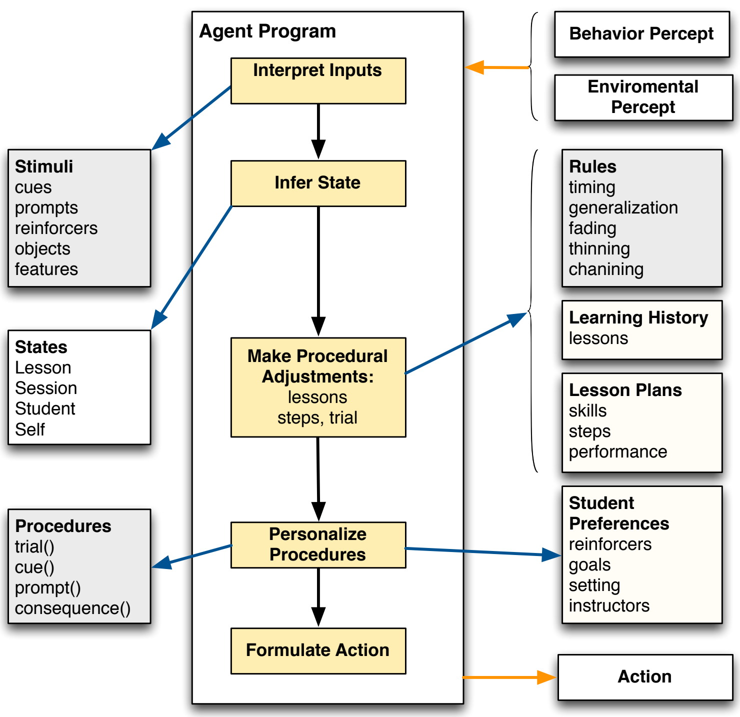

Figure 4.4: ABA Agent Program

4.9.2 Percept Interpreter and Action Generator (PI-AG)

The *Percept Interpreter* and *Action Generator* are bridge (Johnson et al., 1995) components that enable PRA-ABA architecture to transcend the modalities of the instruction. We realize the *abstraction hypothesis* through these two components as they translate abstract notions of behaviors, stimuli, cues, situated states, etc. from and to modality-specific actions and interpretations.

4.9.3 (Percept) Interpreter

The *Interpreter* translates environmental inputs such as positions, movements, expressions, gestures, postures, and actions into events that are understood in terms of representations defined in the ABA Ontology. The *Interpreter* is at any time producing synchronous and asynchronous streams of events. The *Asynchronous* percepts are representations of the student's position and attention to the instruction. The *Synchronous* percepts are representations of student's actions in response to agent's instructional actions. In both cases, the interpreter acts non-deterministically; the interpreter can issue any event at any time as they are representations of the student's actions which are, for the agent, non-deterministic. These events (Figure 4.5) are inputs for the transitions of the state machines that the agent maintains about the state of the instruction and the state of the student in the instruction.

4.9.4 Action Generator

The Action Generator translates the instructional actions such as cues, prompts, and consequences into situational events that are appopriate for the modality of the implementation (embodied, virtual, or mixed). *The Action Generator* also generates locomotoric and other forms of expression that are dependent on the modality of the implementation (Figure 4.6).

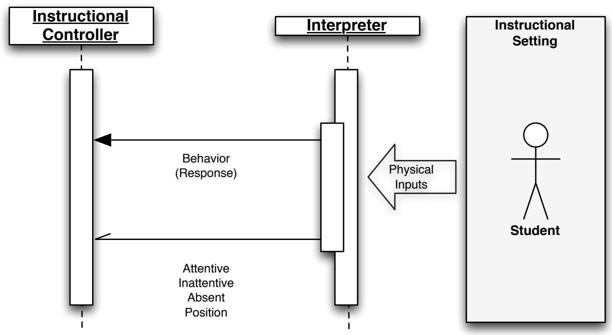

Figure 4.5: Interpretation of Instructional Events

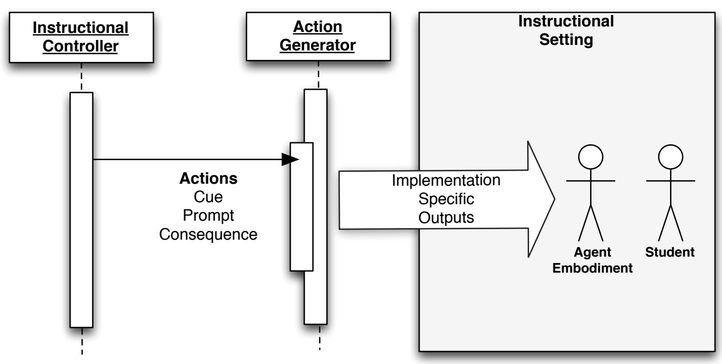

Figure 4.6: Translation of Instructional Actions

4.9.5 Knowledge Base

The Knowledge Base (KB) serves as the store for the general knowledge and facts about the ABA process, individualized lesson plans and preferences, and the memory of the state of instruction. It is also a repository for the rules for the "layout" of the instruction (how the instruction proceeds).

The Knowledge base of the instructional agent consists of:

- General Knowledge (KB-G) - a repository for ABA Ontology-related facts, rules and representations of stimuli and behaviors appropriate for the context (virtual, embodied). This segment of the knowledge base stores facts and rules related to facets of ABA such as schedules or reinforcement or generalization.

- Individualized Knowledge (KB-I) - a repository for student profiles, preferences (stimuli, reinforcers, and prompts), and their learning history. This component of the knowledge base is inspired by a concept from the actual special education practice called Individualized Education Program (IEP) (Smith, 1990). This base is accessed at the beginning of the instruction when the student is first identified, and it is updated as the instruction progresses with statistics about student's learning performance and preferences for reinforces, cues, and prompts.

- Memory (KB-M) - the memory component is used to store the states of the instruction. It is used for bootstrapping purposes and for the resumption of the incomplete or interrupted instructional session. KB-M is updated as frequently as there are any changes to any of the states.

4.10 PRS - Putting It All Together

As discussed in Chapter 2, PRS is the blueprint for the intelligent agent-based implementation of the BDI model in the settings that require dynamically adjusted and sequenced execution of the predefined procedures. In the case of behavioral instruction, pre-defined procedures are instructional procedures (trial, session, generalization, chaining) that are sequenced and adjusted based on student's learning abilities and preferences. A PRS based system consists of a database, goals, knowledge areas, intentions, and an interpreter. Each knowledge area consists of a unit of procedural knowledge. Based on the interpretation of the situation, these units of procedural knowledge are selected for execution as intentions. We use the PRS architecture as an implementation model for the conceptual BDI architecture that we presented in the previous sections. The components of the PRA-ABA mapped to PRS model are:

- *Database* for the instructional agent's beliefs about the states and characteristics of the student, instruction, instructional setting, and the instructor itself,
- *Goals* to be accomplished by the system over the duration of the instructional session and the learning process as a whole,
- *Knowledge Areas (KAs)*, are sequences of actions to be performed in the specific situations,
- *Intentions* which are KAs selected for execution, and
- *Interpreter* serving as an inference mechanism that manages the system.

The table 4.4 outlines the mapping of the PRA-ABA's concepts to the PRS concepts.

Table 4.4: PRS to PRA-ABA Concept Mapping

PRS Concept	PRA-ABA Concept
Database	Knowledge Base (KB)
Goals	Desires
Knowledge Areas (KAs)	Instructional Scripts
Intentions	Intentions
Interpreter	Percept Interpreter, Instructional Reasoner

4.11 Evaluation

This step involves an evaluation of how well the architectural design satisfies the ASRs that were identified during the architectural analysis phase. The architecture evaluation step can be typically performed during the design process, right after the design was completed, or after the architecture was implemented and deployed.

The evaluation for PRA-ABA was performed during the design process (concurrent evaluation), after the design was completed (post-design), and through the POC (post-construction). The results of this evaluation are presented in the next chapter.

4.11.1 Concurrent Evaluation

This is the evaluation that was performed during the design process. Evaluation activities included assurance that all the procedures and conceptual aspects of the ABA ontology were covered by the design of the architecture. Special attention was given to the design of the state machines introduced as part of the ABA ontology and assurance that states and transitions were fully and accurately represented.

4.11.2 Post-design Evaluation

The post-design evaluation focused on the consistency, integrity, and comprehensiveness of the design of the architecture as a whole. Some of the evaluative questions considered were: (i) does the architecture cohesively support the implementation of all ASRs in a agent-oriented way, (ii) does the architecture completely and unambiguously cover all of the instructional scenarios, (iii) are all exceptional scenarios covered, and (iv) are there any conditions that might cause the agent to fail to reach a calculable outcome.

4.11.3 Post-construction Evaluation

Post-construction evaluation was performed by implementing three functional proof-of-concepts: one examining the agent in a role of a verbal behavior instructor, one performing the instruction with a simulated student, and one, interactive, implemented in a virtual setting. These implementations are discussed at the greater level of detail in the next chapter.

4.12 Evolution

Given the research nature of this entire effort, and its early stage of implementation, the process of the architecture evolution has not yet happened. However, the process of technical evolution applies to any architecture, so it is likely that the architecture will significantly evolve over time. A natural direction for the evolution of the architecture is in the areas related to various modalities of implementation, and in the areas related to the implementation of the variants of the ABA methods.

Chapter 5

Implementation and Experimental Evaluation

In the previous two chapters, we described the theoretical properties and proposed capabilities of the PRA-ABA. In this chapter, we describe how we evaluated the proposed architecture and its feasibility as a blueprint for an agent-based instructional system. We evaluated the PRA-ABA by applying both qualitative and quantitative evaluation methods, namely scenario and simulation-based approaches. The examination consisted of an evaluation of the completeness of the overall design and its features against the Architecturally Significant Requirements (ASR), the functioning of the architecture in two different modalities, and the reasoning characteristics of the architecture.

5.1 Assessment of the Software Architecture

There are four broad categories of software architecture evaluation methods, namely: experience-based, simulation-based, mathematical modeling-based, and scenario-based. For the purpose of PRA-ABA evaluation, we applied simulation and scenario-based methods given that experience-based was not practical and that mathematical modeling was feasible only for the small deterministic aspects of the reasoning components.

The purpose of the software architecture is to implement some externally visible functions through the interaction of its constituting components. The purpose of the software architecture evaluation is to examine how well the architecture fulfills this role. The function that the architecture implements is most commonly defined directly or indirectly by the stakeholders (Chung et al., 1999). In the case of PRA-ABA, the stakeholders are ABA-experts and educators that act on behalf of the students. The input was collected from the actual user base as well as from the ABA literature describing the desired characteristics of the instruction.

5.1.1 Completeness of the Implementation

The evaluation of the completeness of the implementation encompassed the evaluation of:

- coverage of the implementation of all the essential requirements of the future system,
- the prioritization and trade-offs in the design of implementing components, and
- efficiency of the implementation (non-redundancy).

The foundation for this assessment was provided by the requirements for the PRA-ABA and its reasoning component synthesized during the architectural synthesis process into a set of competency questions (Grüninger and Fox, 1995). Competency questions are a special kind of requirements that are used as analytic devices to help define the scope of the representation as well as the requirements for the reasoning power of the intelligent system (including agents). Competency questions for PRA-ABA were defined with the help of the domain experts: special education teachers, academics, and certified behavioral specialists. We also used ABA training and certification materials (Shook et al., 2004; Field, S, 2013) as a theoretical foundation.

The high-level system competencies for the PRA-ABA architecture were defined as:

- capability of executing the main components of the behavioral instruction (three-term contingency, discrete trial, session, and lesson) and in an appropriate order,

- capability of inferring the student's learning progress in an ABA-based instruction. That is, the agent should "know" if the student is progressing, regressing, or struggling with particular aspects of the instruction,

- application of the appropriate behavioral instructional methods: issuance of cues, application of prompts, prompt fading, issuance of the consequences, promotion, and recognition of the generalization, and

- capability of inferring the student's state (presence, attentiveness) and learning performance within the instructional session.

In addition, the architecture should support the overall system with the abilities to:

- control the execution of the instruction in a mixed modality (virtual, embodied, etc.),

- pause and resume the instruction,

- modify the execution based on the learning progress of the student, and

- alter the stimuli and behaviors used in the instruction (cues, prompts, consequences).

These high-level competencies were translated into the following specific system competencies:

1. responding to appropriate/expected, approximately expected, unexpected behavior, and no behavior (Behavior recognition),

2. issuance of the appropriate consequence (Consequences),

3. advancing learning objectives from the simplest to more complex (Chaining),

4. issuance of the prompts (Prompting),

5. fading of the prompts (Fading),

6. reinforcing at the predefined schedules (Schedules of reinforcement),

7. fading of the prompts at appropriate rate (Prompt fading),
8. detection of the student's dependency on the prompts in order to produce expected behavior (Prompt dependency),
9. detection of the student's progress or regression (Progress Measurement),
10. "thinning" of the reinforcements (Thinning),
11. generalization,
12. detection of student's attention and absence (Observation of the Student's State),
13. progressing through the instructional process (Lesson advancement),
14. data collection throughout the process, and
15. orderly execution of the instructions including pausing, resumption, and termination of the instructional process.

5.1.2 Evaluation Criteria

The evaluation of the PRA-ABA architecture was performed against the following criteria:

1. *feature coverage* - a measure of how many of the competencies of the future system are covered by the features and functional components of the architecture.
2. *extensibility* - an evaluation of each component extensibility within the architecture.
3. *multimodality* - an assessment of how well the components of the architecture can be implemented in a multimodal setting.

Table 5.1: Mapping of Competencies to Features of the PRA-ABA

Competency	PI	AG	IR	AP	HSM	KB
1. Behavior Recognition	✓			✓		✓
2. Consequences			✓	✓	✓	✓
3. Chaining			✓		✓	✓
4. Prompting		✓	✓	✓	✓	✓
5. Fading		✓	✓		✓	✓
6. Reinforcement Schedules					✓	✓
7. Prompt Fading			✓		✓	✓
8. Prompt Dependency Detection			✓			✓
9. Progress Measurement			✓		✓	✓
10. Thinning			✓		✓	✓
11. Generalization		✓	✓		✓	✓
12. Observation of the Student's State	✓		✓		✓	✓
13. Lesson advancement			✓	✓	✓	✓
14. Data collection				✓		✓
15. Resumable Execution			✓	✓	✓	✓

5.1.3 Findings

The resulting architecture was evaluated by multiple panels representing three groups of stakeholders: special education teachers, certified behavioral therapists, and researchers in the fields of artificial intelligence in education and intelligent agents.

As part of the evaluation, we first ranked competencies for the PRA-ABA by the priority score assigned to them by the panel members, and then attempted to map these prioritized system competencies to the components of the system that implements them. Table 5.1 lists the implementation mapping for the prioritized competency questions.

During the walkthrough with behavior specialists, we discovered that certain aspects of the behavioral instruction cannot be easily implemented in a mixed modality. Functions and features that have limited implementation and that are considered important for the behavioral instruction are:

1. token economy (Ayllón and Azrin, 1968), which is frequently used in the context of operant conditioning, was not implemented.

2. physical/edible reinforcers were not covered nor implemented in a virtual setting (they could be supported with special devices).

3. physical (full) prompts were not implemented.

Token economy could be implemented in the future by issuing specialized control actions to devices that exchange tokens. The same applies for physical or edible reinforcers. The physical prompts are difficult to implement in any of the modalities. In the virtual setting, the agent will not be able to physically assist a student because of the virtual nature of the implementation. In the physical setting the physical contact between the child with a special needs and a machine might require special safety precautions and other accommodations.

5.2 Verbal Behavior Instruction Simulator

The purpose of this simulator was to evaluate the complete implementation of all of the features of the PRA-ABA architecture by implementing them in their simplest form. The verbal-behavior instruction simulator implemented all the essential elements of the behavioral instruction (cue, prompt, consequence, chaining, schedules of reinforcement, etc.) while not implementing any of the more challenging situational or spatial aspects of the instruction. This simulator was also the first opportunity to functionally evaluate one of the fundamental and novel aspects of the proposed approach to the automation of the behavioral instruction, a behavior recognition function and behavior similarity measure.

Implementation

The prototype framework was implemented as a simple Python-based simulator that allows the user to interact with an agent via the command line interface. We used this simulator to simulate the instruction of both verbal and non-verbal behavior. The central components of the architecture were implemented in a Python-based, SPADE distributed agent framework (Gregori et al., 2006). We chose this framework because of its loosely-coupled, simple but expressive, and flexible architecture. SPADE allowed for a great degree of freedom and customization in the implementation of the PRA-ABA components. We also favored SPADE because of its support for Finite State Machine (FSM) model, extensible knowledge base architecture and because of its support of behavior-specific template handlers.

The *Percept Interpreter (PI) and Action Generator (AG)* of the verbal-behavior instructional prototype were implemented using text-to-speech and OSX Dictation & Speech voice recognition (Dixon, 2013, p.39) features which have been available since the 10.8 version of that operating system. We used this technology to issue verbal behaviors - cues, prompts and consequences, as well as to capture the user's voice input as a verbal behavior. We translated the verbal behavior into text-strings and applied the normalized *Levenshtein* distance (Yujian and Bo, 2007) to measure the degree of similarity between the expected and received verbal behaviors.

The specific instructional scenario included execution of few verbal behavioral sessions focused on the *intraverbals* (Frost and Bondy, 2009) of the well known and vocally non-complex concepts (sun, moon, sky, sea).

Verbal behavior was measured by the implementation-specific *behavior similarity function S*. S is implemented as an averaged difference between expected and actual character (textual) descriptor of the *intraverbal*, and a scalar describing its expected and actual duration. To measure textual similarity we used the normalized *levenshtein* distance algorithm* (Yujian and Bo, 2007) to measure the similarity between the character string representation of the expected verbal behavior and of the actual behavior. Formally, S is defined as:

Let c_e be the array of expected characters in the *intraverbal* of length l_e and let c_a be the array of actual *intraverbal* characters of length l_a.

We define the length difference function $L : C_a, C_e \to R$ as

$$L = levenshtein(c_a, c_e)$$

*a dynamic programming algorithm that calculates the *levenshtein* metric expressed as the number of single-characters edits required to change one word into another.

Let d_e be the scalar duration of expected *intraverbal* and let d_a be the duration of the actual *intraverbal*.

We define the function $D : R \to R$ as the difference between two non-negative real numbers representing time intervals for d_e and d_a, normalized on a [0,1] interval:

$$D = 1 - \frac{abs(d_a - d_e)}{max(d_e, d_a)} \text{ for } (d_a - d_e) \neq 0$$

for $max(d_a, d_e) = 0, D = 1$

Consequently, $S : R \to R$, the *intraverbal*-based behavior similarity function, is defined as the arithmetic mean of the functions L and D:

$$S = \frac{L+D}{2}$$

5.2.1 Evaluation

The verbal behavior simulator was evaluated against the list of common everyday intraverbals, common animal, and generalizing intraverbals (Hilsen, 2011). The simulator was evaluated for the accuracy of three-term contingency execution, prompting, and chaining. Prompt action was implemented as a timed task that issues a verbal prompt if there is no verbal behavior within a specified period of time. If the produced verbal behavior was approximately correct, the agent would issue a verbal praise and a corrective intraverbal. If the verbal behavior was correct, the agent would issue a verbal praise. If the verbal behavior was incorrect, the agent would issue a corrective consequence and a corrective intraverbal.

5.2.2 Findings

Overall, the verbal simulator performed correctly on all functional tasks. The simulator had a high error rate on onomatopeic animal intraverbal tasks (Ingvarsson and Le, 2011). It was established that the error rate was related to the speech-to-text auto-correction features. In summary we found that:

- the *behavior similarity measure* is a suitable and practical way to compare the behaviors;

- the speech recognition technology obfuscated or corrected some of the important features of the *intraverbals*. Future implementation would require a different technology to help capture the subtle pronunciation differences ('moo' vs. 'moon').

5.3 Evaluation of the Reasoning Components

This aspect of the evaluation was developed with the expert guidance from special education teachers with extensive experience in conducting ABA-based instruction in the classroom. They helped us construct the list of typical non-verbal behaviors including all the typical variations and patterns of learning and unexpected behaviors.

We translated these behaviors into simulation scripts through an interactive collection instrument. We translated classes of behaviors into codes, assigned them a familiar labels, and created the user-friendly, interactive menus (See 5.1). The purpose of this exercise was to:

- simulate, in a significantly simplified fashion, the actual behavioral interaction between the instructional agent and the student;

- evaluate the capability of architecture to handle different and unexpected behaviors; and

- record the scenarios for the future scripted evaluations of the architecture.

The textual menu offered a choice of actions where the script author for the simulated behavior could choose from a selection of on-task behaviors, approximate behaviors, no or off task behaviors.

```
Show me the green circle>
  1. Point to green circle
  2. Motion towards green circle
  3. Point to red circle
  4. Lose attention
  5. Wander off
  6. Hesitate over green circle
  7. Do nothing
choose your action:
```

Figure 5.1: Interactive Collection Instrument

5.3.1 Evaluation

The choices collected from the interactive collection instrument were recorded as coded input scripts. These inputs were used in place of inputs that would be otherwise generated by the *Percept Interpreter* component of the PRA-ABA. They had to be manually edited to create session scenarios for evaluation. Simulated evaluation sessions covered five scenarios:

1. behavioral instruction with an average learner†,
2. behavioral instruction with a "slow" learner,
3. behavioral instruction with a "fast" learner,
4. behavioral instruction with a learner who regresses, and
5. behavioral instruction with a sporadic learner.

Each session consisted of 10×10 trials.

Session Set 1 - Average Learner (AL)

The session set 1 featured a simulated student that was making an expected progress (as specified in instrumented simulation). The agent started the instructional session with no prompts and with fixed schedules of reinforcement (every response was reinforced). The student was modeled as not making any progress without prompts, so the agent had to

†the progress or speed of learning is judged by a Learning Rate (LR).

apply the initial level of prompting (1) following the least-to-most prompting strategy. With prompts and initially saturated reinforcements the student started a learning progress (detected by Rules 2 and 3). As student progressed, the prompts were scaled backed down to level 0 and then the reinforcements were thinned out at the Fixed Ratio, FR3.

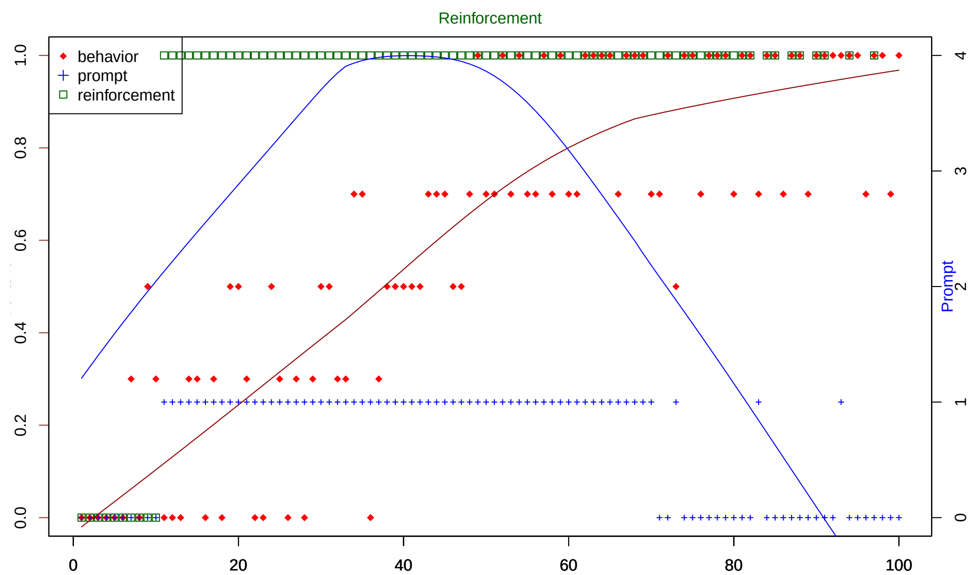

Figure 5.2: Session Set 1 (AL) Instruction Results

Figure 5.2 shows a simulated learning process of an average learner. The Y-axis shows the simulated correct behaviors measured by degree of correctness on the rational scale 0-1. The scale on the right shows the application of reinforcement by the agent (scale 0-1) and the level of prompts applied (0-4 where 0 is independent and 4 is simulated full prompt).

Session Set 2 - Slow Learner (SL)

For session 2 (Figure 5.3), the simulated student was making a slower progress than a student in session set 1 (average learner). Applying the rule 7 (see SPC rules in Chapter 3), the Agent detected the plateau/no-progress in learning, so it started increasing prompt levels from 0 to 3 quickly (from session to session). Higher level of prompts helped the learner accelerate the learning progress. As the student's performance improved, the agent reduced the prompt levels. This experiment confirmed the agent's ability to adjust the prompt levels based on the needs of the learner recognized by SPC rules.

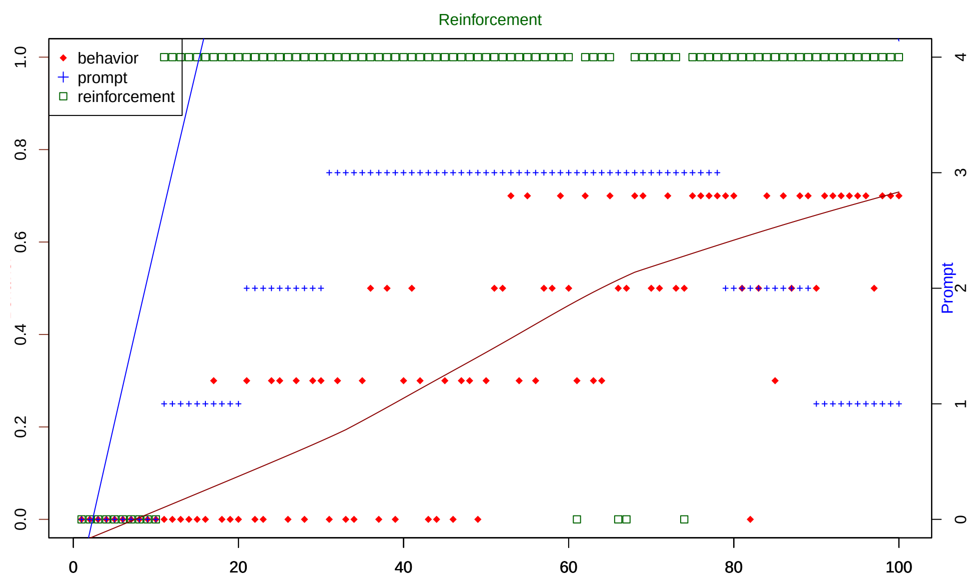

Figure 5.3: Session Set 2 (SL) Instruction Results

Session Set 3 - Fast Learner (FL)

In session 3 (Figure 5.4), the simulated student had a learning rate that helped master the target behavior within four sessions. Applying rules 2, 3, and 7; the agent detected progress and a plateau in performance, so it reduced the prompt level at session 4, thinned out the reinforcers at fixed ratios 3 and 4, and performed the maintenance trial by session 6. The agent stopped the learning process before the 10th session because of the student's mastery of the material (Figure 5.4).

Session Set 4 - Regression in Learning (RL)

This session examined the agent's ability to detect the regression in learning. The script was designed to simulate a student that starts as a fast learner, but that, as the prompts are thinned out, starts exhibiting regression in learning. The agent relied on rules 2 and 3 to detect this condition. As the prompts were introduced, the student's learning performance

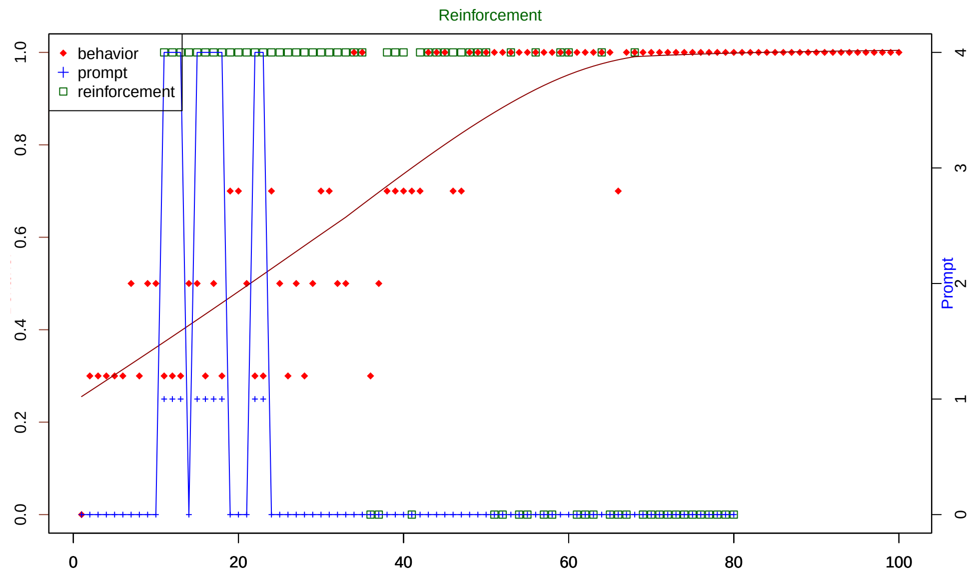

Figure 5.4: Session Set 3 (FL) Instruction Results

increased. This situation also simulated, by design, a prompt dependency. Figure 5.5 shows the final version of the progression of this simulated scenario.

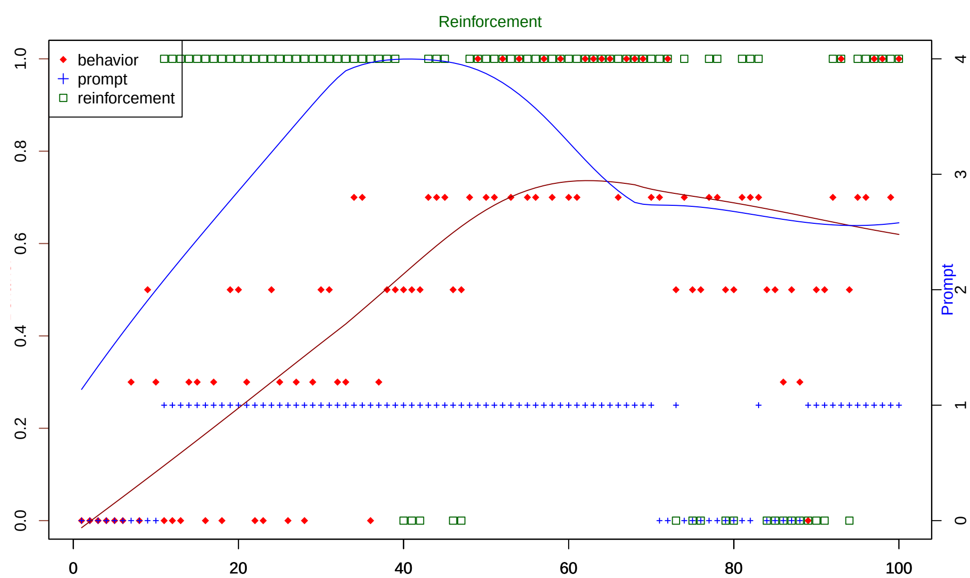

Figure 5.5: Session Set 4 (RL) Instruction Results

Session Set 5 - Sporadic Learner (SPL)

The last experiment in this group of evaluations featured a simulated student that was not sustaining the learning rate. The student was designed to start progressing and then, around the middle of the scenario, start exhibiting the oscillation. The Agent was supposed to detect this situation through SPC rule 4 as a lack of consistent learning. Once the rule was activated, the agent attempted to alternate the stimuli. This alteration did not result in a significant improvement, so the agent signaled this situation as a condition for re-design of the lesson and terminated the session (Figure 5.6).

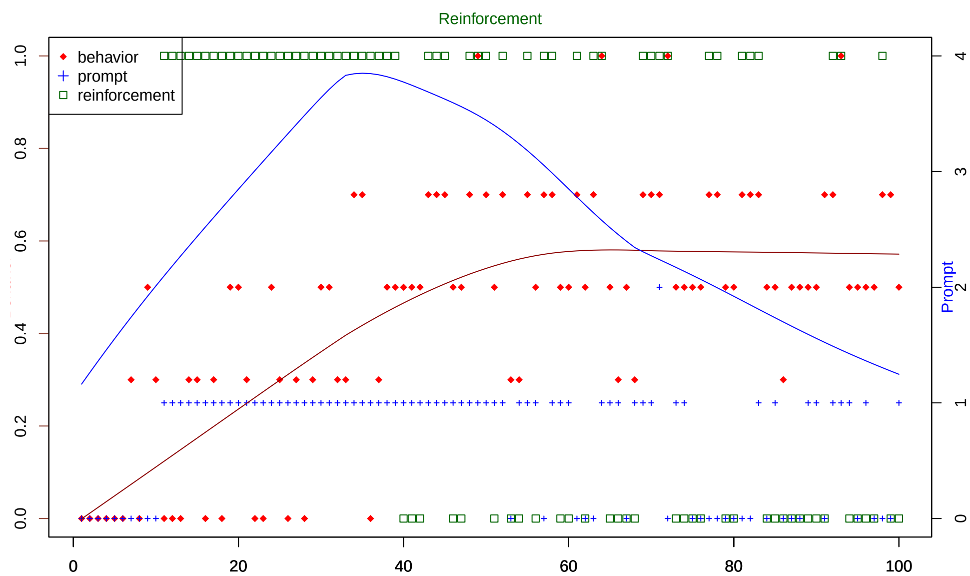

Figure 5.6: Session Set 5 (SPL) Instruction Results

5.3.2 Findings

The simulations have demonstrated satisfactory results on four out of five scenarios. The Agent successfully recognized average, slow, fast learning performance, and the regression in learning. Initially, the agent did not recognize the sporadic learner. Upon closer examination of the test results, it was determined that agent's SPC rule was not activating because measurements were taken at the 10-session sample. The rule had to be adjusted to activate on a smaller sample. In general, we observe that the PRA-ABA exhibited (i) sensitivity to the duration of the session, and (ii) the need for a broader set of SPC rules.

5.4 Evaluation by Construction (Proof-of-Concept)

The purpose of this phase of the evaluation was to test the architecture by actually constructing the proof-of-concept based on the blueprint proposed by the PRA-ABA. The intent was to develop a solution as complete as possible using some existing, sufficiently complete virtual reality technology. The intent of this phase of the evaluation was also to exercise all of the essential elements of the architecture, examine the viability of its construction, and, if the construction was deemed successful, demonstrate the execution of the framework in the actual setting.

5.4.1 Evaluation

The evaluation consisted of the feasibility analysis (Hwang et al., 2006), *feature cover* assessment, and testing of the virtual solution against the common instructional scenarios.

Proof-of-Concept Scenarios

The implementation of the proof-of-concept featured two scenarios. The first scenario exercised general characteristics of the virtual instruction situated in the classroom with the agent, appearing as a female teacher, presenting colored objects of the same shape, and asking student to recognize them. The second scenario examined the capabilities of the architecture to chain the instructional objectives, and to alter the appearance of the instructor, the environment, and the stimuli.

Recognition of Colors

In this scenario, the agent, embodied as a female teacher, instructed a student in a controlled, classroom setting to recognize colors. Instructed colors were red and green with black and yellow used as distractors.

The scenario started with the teacher presenting the objects of different colors and asking the student to point to an object of the right shape. If the student hesitated, the teacher (an agent) offered a visual, verbal, gestural, and model prompts. If the student answered correctly, the teacher issued a rewarding verbal and gestural consequence. If the student did not answer correctly, the teacher issued a correcting consequence in form of a verbal and gestural correction. Each of the consequences used in the instruction were intended to be modifiable by the instructional agent, or by the designer of the instruction. The instruction consisted of ten sessions of ten trials. The target learning performance (LP) was 90%. The schedule of reinforcement was set at a fixed ratio (FR3).

Chaining, Generalization, and Maintenance

In this scenario, the student was instructed to first recognize colors, and then to recognize specific objects of a specific color. This scenario was used to examine the implementation of the learning of complex behaviors through chaining (recognition of colors, recognition of objects, recognition of colored objects). The setting of the instruction was changing as the student progressed through the instruction and demonstrated learning progress (or maintenance).

The scenario was begun with the agent-instructor, appearing as a female instructor, asking a student to identify the objects of the appropriate color. The first trial was assumed to be a maintenance trial. Once the student achieved consistent learning performance, the new behavior was introduced. For the new behavior, the student was asked to identify an object. Once the student achieved a consistent learning performance on this task, the complex new behavior is introduced. The student was asked to identify the specific object (cup) of the specific color (green). Once the student demonstrated satisfactory learning performance and the maintenance of the skill (*LR 90*), the agent program changed the environment of the instruction and the sessions re-started.

Generalizing changes were applied in a pre-defined order: first, the environment was changed; next, the appearance of the instructor was changed from female to male. The instruction was to conclude if the student demonstrated a satisfactory learning performance and maintenance of the newly acquired skill. Otherwise, the agent would re-adjust the prompt level in order to support the student's improvement of the learning performance.

5.4.2 Implementation

For the virtual reality version of PRA-ABA, we used the Unity 3D framework (Blackman, 2013). Unity 3D was chosen because of its successful use in other commercial, gaming, and research-oriented virtual reality (Gratch et al., 2013) applications. Another reason for using Unity 3D was its good support for the development of artificial intelligence-oriented applications (Finkelstein et al., 2009). Related to specific PRA-ABA needs, the following features were of particular interest:

- broad availability of free and commercial animation assets,
- Unity's implementation of the concept of hierarchical state machines (*Multilayer Hierarchical State Machine (MHSM)*) (with close mapping to PRA-ABA's state machine concepts),
- dynamic generation of character animations and expressions through its encompassing *Mecanim* API.

In addition, Unity 3D supports custom scripting in C# and JavaScript. We chose C# for the implementation of the custom code for PRA-ABA because of the greater availability of AI-related code and learning materials in that language.

Table 5.2: Mapping of PRA-ABA Components to Unity Components

PRA-ABA Component	Animations	Unity Engine	MHSM	Custom
Percept Interpreter	✓	✓		✓
Action Generator (Actions)	✓	-		✓
Instructional Reasoner	-	-	-	✓
Agent Program		✓		
HSM			✓	✓
Knowledge Base		-		✓

General Approach

One of the main objectives of the proof-of-concept (POC) was to demonstrate the feasibility, dynamicity, and flexibility of the PRA-ABA when implemented in a virtual modality. To this end, we explored options for the dynamic execution of the instruction, changes to the appearance of the learning setting and the instructor, and ability to parametrize the instruction.

To meet these goals we:

1. implemented the instruction in two different virtual settings (classroom and playground, see figure 5.7);

2. used a collection of pre-built assets (animated scripts for verbal and gestural behaviors, stimuli), and custom scripts for PRA-ABA actions;

3. used Unity finite state machine model (MHSM), otherwise used for AI programming of *non-player characters* (NPCs), for the implementation of HSM; and

4. custom implemented a knowledge base (KB) and an agent program using the built-in C# engine.

The mapping of the PRA-ABA components to Unity 3D architecture is outlined in Table 5.2.

The scene shown in Figure 5.7 shows the default proof-of-concept instructional setting with the agent presented as a female instructor and in a typical classroom setting.

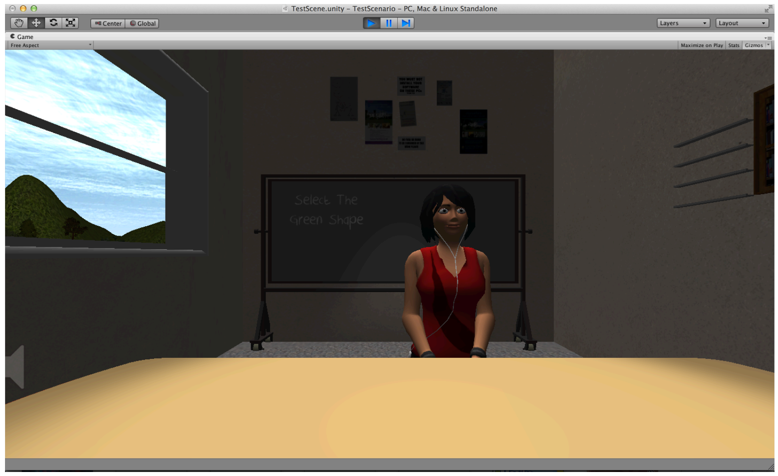

Figure 5.7: Virtual Instruction - Initial Classroom Scene

As part of the implementation of the behavioral instruction, we implemented all typical actions and components of the discrete trial. *Cues* and *prompts* were implemented as Unity 3D animations. A complete, non-physical hierarchy of prompts was implemented using this approach; we implemented verbal prompts, gestural prompts, and model prompts. Figure 5.8 shows the agent issuing a gestural prompt accompanied with a verbal prompt.

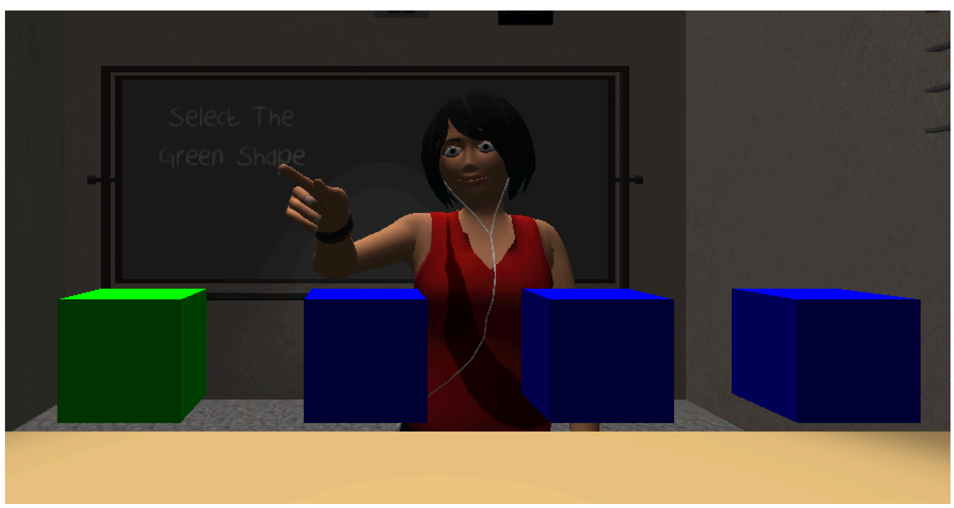

Figure 5.8: Instructor Prompts (Gesture) the Target Object

The prompts were activated using Unity 3D timer features that triggered the prompting animation if the behavior was not produced within the specified *prompt wait time* (P_{wt}).

Reinforcers were implemented in the same fashion as prompts, using select Unity 3D animation assets. Their execution was controlled by the custom-built routines of the PRA-ABA agent program. Figure 5.9 shows the agent reinforcing student's response with a verbal and gestural praise. These two actions play the role of a reward (R).

Figure 5.9: Instructor issues a Consequence (Reinforcement)

The same mechanism was used for the corrective prompts as well. Figure 5.10 shows the instructor issuing the corrective consequence for the incorrect behavior (incorrect response). Corrections, like rewards, were rule-driven and parametrized (i.e. the animation representing a corrective action was replaceable, in a programmatic fashion, by another). In the situation presented in Figure 5.10, we used combination of off-the-shelf *Mixamo* animations (Mixamo Inc., 2013): pointing at the right object, providing a verbal direction, and a head shake.

One of the significant strengths of this platform was its support for the promotion of generalized learning. The POC implementation demonstrated support for the generalization by: (i) alteration of the environment, (ii) alteration of the setting including instructional objects and stimuli, and (iii) appearance of the instructor. These generalizing characteristics

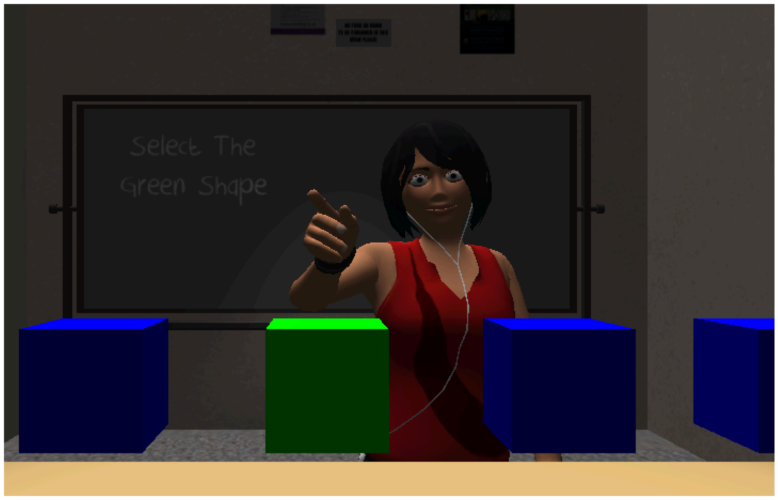

Figure 5.10: Instructor issues a Consequence (Correction)

of the instruction were implemented as configurable properties, adjustable by the agent program and instructional reasoner. Figure 5.11 shows the alternations in the instructional environment that have happened as the student had successfully progressed through the sessions and chained behaviors.

Figure 5.11: Generalizing Instructional Environment

5.4.3 Findings

The main intent of the proof-of-concept implementation was to examine the validity of the PRA-ABA framework by implementing it in a practical or near-practical setting. During this process, we primarily sought to discover any egregious obstacles or severe issues that would uncover some flaw in the PRA-ABA architecture. There were no such discoveries. Although this finding supports the general philosophy of the PRA-ABA architecture, the POC was limited in the implementation, and was not evaluated in a formal experimental setting with human subjects. Hence the findings outlined below are only preliminary:

- The Unity and C# *Timers* played the essential role in the implementation of the discrete trial, namely the control of the trial duration, the prompting, and the issuance of the consequences. Despite the successful final implementation, we experienced challenges working with Unity 3D in this respect. The framework has a limited support for complex, concurrent, and coordinated timers, so a custom framework had to be implemented to overcome these limitations. We expect that the implementation of the time-dependent routines in systems with more limited support for *timers* would be significantly more challenging.

- Our original design of a *Knowledge Base* for action-producing rules had to be modified. Our original design represented actions as singular, atomic objects. Rewards or correcting consequences in the Unity implementation had to be represented as complex as they consisted of the multiple animations. Hence, we had to modify the representation scheme of the areas of the knowledge base that maps the activation rules for behaviors to actions (animations) .

- Although we did not perform a complete usability testing, we observed that virtual modality will likely be an excellent environment for the promotion of generalization in learning.

5.5 Summary

After reviewing the performed evaluation steps and the outcomes of these evaluations, the following high-level conclusions emerge:

- the ABA ontology is a strong theoretical foundation for the implementation of PRA-ABA-based instruction. It provides a sufficient and complete conceptualization for the practical implementation of the instruction.
- the instructional components (three-term contingency, prompting, generalization, schedules of reinforcement) of the behavioral instruction are properly covered by the PRA-ABA architecture, and they offer a good foundation for practical implementation.
- further evaluatory implementations and usability experiments with actual students are required in order to fully determine the practicality of the working PRA-ABA solution, especially in the physically embodied setting.

Chapter 6

Conclusions and Future Work

6.1 Findings and Conclusions

The intent for the architecture described in this dissertation is to serve as a conceptual foundation for the implementation of a broad array of agent-mediated instructional applications. Our preliminary experiments have shown that our architecture's flexible and decoupled design can support implementation of different behavioral instructions with no substantial changes to any of the essential components. Of all the concepts covered in this dissertation, we highlight three that, we believe, significantly contribute to the advancement the knowledge in this area: (i) defining behavior measures that make the behavioral instruction computable, (ii) use of the statistical process control methods to track the progress of the instruction and the triggering of interventions, and (iii) inception of the ABA ontology as the formalization of the behavioral instruction. All three concepts warrant further research and development.

6.1.1 The Measures of Behavior

We treated behavior as a multi-dimensional phenomenon that can be mathematically described by the set of numbers organized, perhaps as a multi-dimensional vector or a matrix. This approach has been one of the pivotal choices of the entire research, as it allowed the agent to base its three-term contingency instructional reasoning on the calculation about the appropriateness of the behavior exhibited by the student. We used this approach in the problems where behavior was describable by simple tuples of numbers or sequences of characters. We did not, however, pursued a full theoretical examination of the possible complexity of behavior description or the recognition. We believe that problems in this area are worth further exploring, especially the problem of the possible intractability of comparing complex, behavioral representation structures (e.g. isomorphism of graph-representations of the spatial behavior).

6.1.2 Utility of Statistical Process Control (SPC)

Introduction of the SPC methods to track the progress of the student's learning, the planned (or unplanned) variations, and anomalies in learning is another area that we believe warrants further examination. We believe that our limited exploration of the SPC rules is a step in the right direction for both computer and educational science, but we also believe that it will likely need further exploration, modifications and experimental evaluations.

6.1.3 ABA Ontology and Reinforcement Instruction

Finally, we believe that the concept of ABA ontology is novel, and that it offers a potential for further research. In particular, it offers a foundation for a new sub-area in machine learning and agent-based systems which we speculate to call *reinforcement instruction*. We see reinforcement instruction as a complementary method whereby instead of agents learning how to perform the task, they learn how to most effectively instruct humans or other agents in a behavioral instructional setting.

6.2 Future Work

While addressing questions of a conceptual nature the architecture itself leaves three major areas to be further examined and developed, namely: (i) algorithms for behavior recognition, (ii) implementation of robust learning components, and (iii) development of reusable virtual and embodied frameworks based on the proposed architecture. We are already in process of addressing some of these issues in our ongoing research (Begoli et al., 2013), but we believe that they warrant examination by the broader research community.

6.2.1 Behavior Recognition

In our architecture we proposed the percept interpreter component while leaving it, except for the two small prototypes, largely unexplored. In the context of the proposed architecture, behavior recognition in the context of the proposed architecture will require a substantial and diverse effort involving image and motion recognition, behavior modeling, and coding. We further speculate that human behavior recognition, as a computing problem, has the potential to develop into its own sub-area of computer science. For example, it is not entirely unreasonable to expect that there might be areas of computer science dedicated to the formalization and improvement of machine recognition of spatial, *intensional* and temporal aspects of different classes of human behavior.

6.2.2 Learning Component

Current implementation of the PRA-ABA architecture has a minimal deliberate learning function which is based mostly on the student's learning history, preferences and basic statistic. We believe that, as part of the future enhancements, the system could benefit from a more robust learning capability; the agent could be enhanced to learn about the changes and characteristics of the environment, the instruction and the student beyond the currently implemented personalization and history features. This learning component could further enhance the agent's ability to act independently and adjust the features of the instruction (stimuli, prompts, consequences).

6.2.3 Full Virtual Reality Implementations

The research in this thesis, focused mostly on the development of the conceptual model and on the validation of the architecture. However, our overarching intention for the idea of agent-mediated behavioral instruction, is to advance it towards the long-term research and development program focused on the development of virtual and mixed-reality therapeutical applications. Therefore, our long-term plan is to develop a complete application development environment consisting of the programming API, a domain specific language (DSL) for coding of the ABA-based instructional scenarios, and a supporting compiler that will translate the DSL expressions into some form of executable 3D representation. One of the possible directions is collaboration with a *Virtual Human* project (Gratch et al., 2013) whereby some of the elements of the ABA architecture and its behavioral-instructional concepts would be integrated into a *Virtual Human* toolset and its executables.

Bibliography

AES (2014). Discrete trial trainer. Accelerations Educational Software (AES), http://www.dttrainer.com (Retrieved: 2-27-2014). ix, 7

Alberto, P., Troutman, A. C., et al. (2009). *Applied behavior analysis for teachers*. Merrill/Pearson. 9

Alur, R., Kannan, S., and Yannakakis, M. (1999). Communicating hierarchical state machines. In *Automata, Languages and Programming*, pages 169–178. Springer. 101

American Psychiatric Association (2000). *Diagnostic and statistical manual of mental disorders: DSM-IV-TR®*. American Psychiatric Pub. 1

Ashton, T. M. (2001). Assistive Technology: The Application of ABA to Technology: The Discrete Trial Trainer. *Journal of Special Education Technology*, 16(1):41–42. 7

Ayllón, T. and Azrin, N. (1968). *The token economy: a motivational system for therapy and rehabilitation*. Century psychology series. Appleton-Century-Crofts. 115

Baer, D. M., Wolf, M. M., and Risley, T. R. (1968). Some current dimensions of applied behavior analysis. *Journal of applied behavior analysis*, 1(1):91–97. 2, 8, 27

Baio, J. (2012). Prevalence of Autism Spectrum Disorders: Autism and Developmental Disabilities Monitoring Network, 14 Sites, United States, 2008. Morbidity and Mortality Weekly Report. Surveillance Summaries. Volume 61, Number 3. *Centers for Disease Control and Prevention*. 1

Bakker, A., Kent, P., Derry, J., Noss, R., and Hoyles, C. (2008). Statistical inference at work: Statistical process control as an example. *Statistics Education Research Journal*, 7(2):130–145. 99

Barakova, E., Gillessen, J., and Feijs, L. (2009). Social training of autistic children with interactive intelligent agents. *Journal of Integrative Neuroscience*, 8(01):23–34. 6

Barakova, E. and Lourens, T. (2013). Interplay between natural and artificial intelligence in training autistic children with robots. In *Natural and Artificial Models in Computation and Biology*, pages 161–170. Springer. 40

Bartoli, L., Corradi, C., Garzotto, F., and Valoriani, M. (2013). Exploring motion-based touchless games for autistic children's learning. In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 102–111. ACM. 36

Bass, L., Clements, P., and Kazman, R. (2003). *Software Architecture in Practice*. SEI series in software engineering. Addison-Wesley. 81

Begoli, E., Ogle, C. L., Cihak, D. F., and MacLennan, B. J. (2013). Towards an integrative computational foundation for applied behavior analysis in early autism interventions. In *Artificial Intelligence in Education*, pages 888–891. Springer. 136

Behavior Analyst Certification Board (2012). BCBA & BCaBA Behavior Analyst Task List, Fourth Edition. 82

Bernardini, S., Porayska-Pomsta, K., and Smith, T. J. (2013). Echoes: An intelligent serious game for fostering social communication in children with autism. *Information Sciences*, 260:41–60. 36

Blackman, S. (2013). *Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development*. Technology in action series. Apress. 127

Blumberg, S. J., Bramlett, M. D., Kogan, M. D., Schieve, L. A., Jones, J. R., and Lu, M. C. (2013). Changes in prevalence of parent-reported autism spectrum disorder in school-aged us children: 2007 to 2011–2012. *National Health Statistics Reports*, 65:1–11. 1

Bogin, J., Sullivan, L., Rogers, S., and Stabel, A. (2010). Discrete trial training (DTT) for children and youth with autism spectrum disorders: Online training module. Technical report, Sacramento, CA: National Professional Development Center on Autism Spectrum Disorders, M.I.N.D. Institute, University of California at Davis Medical School. 47

Breazeal, C. (2003). Toward sociable robots. *Robotics and Autonomous Systems*, 42(3):167–175. 37

Burns, R. (1980). Verbal and non-verbal communication. In *Essential Psychology*, pages 237–248. Springer Netherlands. 49

Busoniu, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172. 90

Cassell, J. (2000). *Embodied conversational agents*. The MIT Press. 34

Chang, Y.-J., Chen, S.-F., and Huang, J.-D. (2011). A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570. 95

Chowdhury, M. and Benson, B. A. (2011). Use of differential reinforcement to reduce behavior problems in adults with intellectual disabilities: A methodological review. *Research in Developmental Disabilities*, 32(2):383–394. 62

Chung, L., Gross, D., and Yu, E. (1999). Architectural design to meet stakeholder requirements. In *Software Architecture*, pages 545–564. Springer. 112

Cohen, H., Amerine-Dickens, M., and Smith, T. (2006). Early intensive behavioral treatment: Replication of the UCLA model in a community setting. *Journal of Developmental & Behavioral Pediatrics*, 27(2):S145–S155. 3

Cooper, J. O., Heron, T. E., Heward, W. L., et al. (2007). *Applied Behavior Analysis*. Pearson/Merrill-Prentice Hall Upper Saddle River, NJ. 2, 9

Cordeschi, R. (2013). Automatic decision-making and reliability in robotic systems: some implications in the case of robot weapons. *AI & society*, 28(4):431–441. 30

Cosgrave, G. (2013). Discrete Trial Training. Online: http://www.educateautism.com/applied-behaviour-analysis/discrete-trial-training.html. Accessed: 2013-12-24. 4

Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation? *AI magazine*, 14(1):17. 43

De Boer, R. C. and Van Vliet, H. (2009). On the similarity between requirements and architecture. *Journal of Systems and Software*, 82(3):544–550. 82

Dias, J. and Paiva, A. (2005). Feeling and reasoning: A computational model for emotional characters. In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence*, volume 3808 of *Lecture Notes in Computer Science*, pages 127–140. Springer Berlin Heidelberg. 36

Diehl, J. J., Schmitt, L. M., Villano, M., and Crowell, C. R. (2012). The clinical use of robots for individuals with autism spectrum disorders: A critical review. *Research in Autism Spectrum Disorders*, 6(1):249–262. 36, 40

Dixon, W. (2013). *OS X 10.8 Mountain Lion Client And Server: A Review and History*. Wayne Dixon. 117

Eikeseth, S., Smith, T., Jahr, E., and Eldevik, S. (2002). Intensive behavioral treatment at school for 4-to 7-year-old children with autism a 1-year comparison controlled study. *Behavior modification*, 26(1):49–68. 3

Eldevik, S., Hastings, R. P., Hughes, J. C., Jahr, E., Eikeseth, S., and Cross, S. (2009). Meta-analysis of early intensive behavioral intervention for children with autism. *Journal of Clinical Child & Adolescent Psychology*, 38(3):439–450. 9

Emerson, E. A. (1990). Temporal and modal logic. *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, 995:1072. 50

Feil-Seifer, D. and Mataric, M. J. (2005). Defining socially assistive robotics. In *ICORR 2005. 9th International Conference on Rehabilitation Robotics*, pages 465–468. IEEE. 37

Feil-Seifer, D. and Mataric, M. J. (2008). B^3IA: A control architecture for autonomous robot-assisted behavior intervention for children with Autism Spectrum Disorders. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 328–333. IEEE. ix, 38, 39

Field, S (2013). Designing and Implementing Social Skills Instruction for Individuals with Autism Spectrum Disorders. Presentation. Online: `http://goo.gl/f9tqN3` (Retrieved:3-1-2014). 112

Finkelstein, S. L., Nickel, A., Harrison, L., Suma, E. A., and Barnes, T. (2009). cMotion: A new game design to teach emotion recognition and programming logic to children using virtual humans. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 249–250. IEEE. 127

Foxx, R. M. (2008). Applied behavior analysis treatment of autism: The state of the art. *Child and adolescent psychiatric clinics of North America*, 17(4):821–834. 1, 3

Frost, L. and Bondy, A. (2009). A Common Language: Using BF Skinner's Verbal Behavior for Assessment and Treatment of Communication Disabilities in SLP-ABA. *Journal of Speech-Language Pathology & Applied Behavior Analysis*, 4. 117

Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *AAAI*, volume 87, pages 677–682. 32

Gratch, J., Hartholt, A., Dehghani, M., and Marsella, S. C. (2013). Virtual humans: A new toolkit for cognitive science research. In *Cognitive Science*, Berlin, Germany. 127, 137

Gregori, M. E., Cámara, J. P., and Bada, G. A. (2006). A jabber-based multi-agent system platform. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1282–1284. ACM. 116

Gruber, T. R. et al. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220. 44

Grüninger, M. and Fox, M. (1995). Methodology for the Design and Evaluation of Ontologies. In *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, April 13, 1995*. 82, 112

Gülbay, M. and Kahraman, C. (2006). Development of fuzzy process control charts and fuzzy unnatural pattern analyses. *Computational statistics & data analysis*, 51(1):434–451. 99

Haynes, S. N. and O'Brien, W. H. (1990). Functional analysis in behavior therapy. *Clinical Psychology Review*, 10(6):649–668. 46

Hilsen, L. (2011). *A Step-by-Step Curriculum for Early Learners with Autism Spectrum Disorders*. Jessica Kingsley Publishers. 118

Hofmeister, C., Kruchten, P., Nord, R. L., Obbink, H., Ran, A., and America, P. (2007). A general model of software architecture design derived from five industrial approaches. *Journal of Systems and Software*, 80(1):106–126. 81

Holt, G. L. and Shafer, J. N. (1973). Function of intertrial interval in matching-to-sample1. *Journal of the Experimental Analysis of Behavior*, 19(1):181–186. 54

Holz, T., Dragone, M., and O'Hare, G. (2009). Where robots and virtual agents meet. *International Journal of Social Robotics*, 1(1):83–93. 7

Howard, J. S., Sparkman, C. R., Cohen, H. G., Green, G., and Stanislaw, H. (2005). A comparison of intensive behavior analytic and eclectic treatments for young children with autism. *Research in developmental disabilities*, 26(4):359–383. 3

Hwang, J., Jung, J., and Kim, G. J. (2006). Hand-held virtual reality: a feasibility study. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 356–363. ACM. 125

Ingvarsson, E. T. and Le, D. D. (2011). Further evaluation of prompting tactics for establishing intraverbal responding in children with autism. *The Analysis of verbal behavior*, 27(1):75. 118

Jarrar, M. and Meersman, R. (2009). Ontology engineering–the DOGMA approach. In *Advances in Web Semantics I*, pages 7–34. Springer. 46

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41. 88

Johnson, R., Helm, R., Vlissides, J., and Gamma, E. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional. 105

Johnson, W. L., Rickel, J. W., and Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11(1):47–78. 33

Keating, M. (2011). Hierarchical state machines. In *The Simple Art of SoC Design*, pages 47–54. Springer New York. 101

Koegel, R. L. and Kern Koegel, L. (2006). *Pivotal Response Treatments for Autism: Communication, Social, and Academic Development.* ERIC. 5

Kripke, S. A. (1963). Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96. 51

Lai, K., Konrad, J., and Ishwar, P. (2012). A gesture-driven computer interface using kinect. In *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, pages 185–188. IEEE. 95

Lee, J. and Marsella, S. (2006). Nonverbal behavior generator for embodied conversational agents. In Gratch, J., Young, M., Aylett, R., Ballin, D., and Olivier, P., editors, *Intelligent Virtual Agents*, volume 4133 of *Lecture Notes in Computer Science*, pages 243–255. Springer Berlin Heidelberg. 35

Lord, C., Rutter, M., Goode, S., Heemsbergen, J., Jordan, H., Mawhood, L., and Schopler, E. (1989). Austism diagnostic observation schedule: A standardized observation of communicative and social behavior. *Journal of autism and developmental disorders*, 19(2):185–212. 14

Lovaas, O. I. (1987). Behavioral treatment and normal educational and intellectual functioning in young autistic children. *Journal of consulting and clinical psychology*, 55(1):3. 14

Lovaas, O. I., Ackerman, A., Alexander, D., Firestone, P., Perkins, J., Young, D., et al. (1981). *Teaching developmentally disabled children: The me book*. Pro-ed Austin, TX. 2, 4

MacGregor, J. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414. 93

Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(12):1321–1329. 7

Milne, M., Luerssen, M., Lewis, T., Leibbrandt, R., and Powers, D. (2011). Embodied conversational agents for education in autism. *A Comprehensive Book on Autism Spectrum Disorders*, page 387. 34

Mixamo Inc. (2013). Mixamo Animation Platform. Mixamo Inc. website: `http://www.mixamo.com/` (Accessed: 12-17-2013). 130

Mower, E., Lee, C.-C., Gibson, J., Chaspari, T., Williams, M. E., and Narayanan, S. (2011). Analyzing the nature of ECA interactions in children with Autism. In *In Proceedings of Interspeech*, Florence, Italy. 35

Myers, S. M., Johnson, C. P., et al. (2007). Management of children with autism spectrum disorders. *Pediatrics*, 120(5):1162–1182. 3

National Research Council (US). Committee on Educational Interventions for Children with Autism (2001). *Educating children with autism*. National Academies Press. 3

Nelson, B. L. (1992). Statistical analysis of simulation results. *Handbook of Industrial Engineering: Technology and Operations Management, Third Edition*, pages 2469–2495. viii, 99, 100

Oikonomidis, I., Kyriazis, N., and Argyros, A. (2011). Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11. BMVA Press. http://dx.doi.org/10.5244/C.25.101. 95

Poole, D. L. and Mackworth, A. K. (2010). *Artificial Intelligence: foundations of computational agents*. Cambridge University Press. 83, 84

Porayska-Pomsta, K., Frauenberger, C., Pain, H., Rajendran, G., Smith, T., Menzies, R., Foster, M. E., Alcorn, A., Wass, S., Bernadini, S., et al. (2012). Developing technology for autism: an interdisciplinary approach. *Personal and Ubiquitous Computing*, 16(2):117–127. ix, 36

Powell, S. and Jordan, R. (2012). *Autism and learning: A guide to good practice*. Routledge London. 6

Prendinger, H., Descamps, S., and Ishizuka, M. (2004). MPML: A markup language for controlling the behavior of life-like characters. *Journal of Visual Languages & Computing*, 15(2):183–203. 88

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *KR*, 91:473–484. 32, 89

Rao, P. A., Beidel, D. C., and Murray, M. J. (2008). Social skills interventions for children with Asperger's syndrome or high-functioning autism: A review and recommendations. *Journal of autism and developmental disorders*, 38(2):353–361. 14

Reidsma, D., de Kok, I., Neiberg, D., Pammi, S. C., van Straalen, B., Truong, K., and van Welbergen, H. (2011). Continuous interaction with a virtual human. *Journal on Multimodal User Interfaces*, 4(2):97–118. 88

Reynolds, J. C. (1974). Towards a theory of type structure. In Robinet, B., editor, *Programming Symposium*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425. Springer Berlin Heidelberg. 63

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall. 31, 87

Scassellati, B., Admoni, H., and Mataric, M. (2012). Robots for use in autism research. *Annual Review of Biomedical Engineering*, 14:275–294. 37, 38

Seber, G. A. and Lee, A. J. (2012). *Linear regression analysis*, volume 936. John Wiley & Sons. 26

Serenko, A., Bontis, N., and Detlor, B. (2007). End-user adoption of animated interface agents in everyday work applications. *Behaviour and Information Technology*, pages 119–132. 34

Shamsuddin, S., Yussof, H., Ismail, L., Hanapiah, F. A., Mohamed, S., Piah, H. A., and Zahari, N. I. (2012). Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pages 188–193. IEEE. 37, 40

Shaw, M., DeLine, R., Klein, D. V., Ross, T. L., Young, D. M., and Zelesnik, G. (1995). Abstractions for software architecture and tools to support them. *Software Engineering, IEEE Transactions on*, 21(4):314–335. 89

Shook, G. L., Johnston, J. M., and Mellichamp, F. H. (2004). Determining essential content for applied behavior analyst practitioners. *The Behavior Analyst*, 27(1):67. 112

Skinner, B. F. (1938). The behavior of organisms: An experimental analysis. *Appleton-Century*. 14, 15

Skinner, B. F. (1953). *Science and human behavior*. Simon and Schuster. 14

Sklar, E. and Richards, D. (2010). Agent-based systems for human learners. *Knowledge Engineering Review*, 25(2):767–777. ix, 33, 34

Smith, B. C. (1982). *Reflection and semantics in a procedural language*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science. 43

Smith, S. W. (1990). Individualized education programs (IEPs) in special education-from intent to acquiescence. *Exceptional Children*. 107

Smith, T. (2001). Discrete trial training in the treatment of autism. *Focus on autism and other developmental disabilities*, 16(2):86–92. 4

Smith, T., Scahill, L., Dawson, G., Guthrie, D., Lord, C., Odom, S., Rogers, S., and Wagner, A. (2007). Designing research studies on psychosocial interventions in autism. *Journal of autism and developmental disorders*, 37(2):354–366. 3

Soares, A., Sousa, J., Azevedo, A., and Bastos, J. (1997). Using an informal ontology in the development of a planning and control system âĂŤ the case of the virtual enterprise. In Camarinha-Matos, L., editor, *Re-engineering for Sustainable Industrial Production*, IFIP âĂŤ The International Federation for Information Processing, pages 107–118. Springer US. 11

Suárez-Figueroa, M. C. (2010). *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. PhD thesis, Informatica. 46

Tardif, C., Plumet, M.-H., Beaudichon, J., Waller, D., Bouvard, M., and Leboyer, M. (1995). Micro-analysis of social interactions between autistic children and normal adults in semi-structured play situations. *International Journal of Behavioral Development*, 18(4):727–747. 39

Tepper, P., Kopp, S., and Cassell, J. (2004). Content in context: Generating language and iconic gesture without a gestionary. In *Proceedings of the Workshop on Balanced Perception and Action in ECAs at Automous Agents and Multiagent Systems (AAMAS)*. 35

Texas Statewide Leadership for Autism Training (2012). Discrete trial training. TARGET: Texas Autism Resource Guide for Effective Teaching http://www.txautism.net/uploads/target/DTT.pdf. 30

The BCBA Board of Directors (2013). BACB Newsletter Special Edition on New Standards. The Behavior Analyst Certification Board http://www.bacb.com/newsletter/BACB_Newsletter_2-13.pdf. 82

Venkatesh, S., Phung, D., Duong, T., Greenhill, S., and Adams, B. (2013). Toby: early intervention in autism through technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3187–3196. ACM. 7

Vilhjálmsson, H., Cantelmo, N., Cassell, J., Chafai, N. E., Kipp, M., Kopp, S., Mancini, M., Marsella, S., Marshall, A. N., Pelachaud, C., et al. (2007). The behavior markup language: Recent developments and challenges. In *Intelligent Virtual Agents*, pages 99–111. Springer. 35, 88

Volkmar, F. R., Paul, R., Klin, A., and Cohen, D. J. (2005). *Handbook of Autism and Pervasive Developmental Disorders, Diagnosis, Development, Neurobiology, and Behavior*, volume 1. John Wiley & Sons. 1

Wainer, J., Dautenhahn, K., Robins, B., and Amirabdollahian, F. (2010). Collaborating with Kaspar: Using an autonomous humanoid robot to foster cooperative dyadic play among children with autism. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 631–638. IEEE. ix, 8

Watson, J. B. (1914). *Behavior: An introduction to comparative psychology*. Henry Holt & Company. 14

Welch, K. C., Lahiri, U., Warren, Z., and Sarkar, N. (2010). An approach to the design of socially acceptable robots for children with Autism Spectrum Disorders. *International Journal of Social Robotics*, 2(4):391–403. 6

Werf, R. (2008). The embodied conversational agent toolkit: a new modularization approach. Master's thesis, University of Twente. 35

Wille, R. (2005). Formal concept analysis as mathematical theory of concepts and concept hierarchies. In *Formal Concept Analysis*, pages 1–33. Springer. 46

Willett, J. B. (1988). Questions and answers in the measurement of change. *Review of research in education*, 15:345–422. 65

Wooldridge, M. (1997). Agent-based software engineering. *IEE Proceedings-software*, 144(1):26–37. 34

Yujian, L. and Bo, L. (2007). A normalized levenshtein distance metric. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1091–1095. 117

Vita

Edmon Begoli was born in Croatia, in 1972. He got his bachelors degree in Computer Science from East Tennessee State University in 1998 (*Magna Cum Laude*), and his masters degree in Computer Science from the University of Colorado-Boulder in 2003. From 1997 through the end of 2003 he worked as a Member of the Research Staff at Lucent Technologies. From 2003 through 2004, he worked as the Enterprise Architect for Laboratory Coporation of America (LabCorp). In 2004, Edmon joined Y-12 National Security Complex, US Department of Energy. In 2007, Edmon moved to Oak Ridge National Laboratory (ORNL) where, as a member of the R&D staff, he served as a Principal Investigator (PI) and Chief Architect on several major national security and national healthcare-related R&D initiatives. Edmon's work on the next generation of the healthcare analytic platforms was featured in the White House/Office of Science and Technology Policy special report on *Big Data* applications. Edmon was also the invited expert with W3C Multimodal Interaction (MMI) Group from 2010 to 2012 for the implementation of the new standard for expression of emotional behavior (EmotionML). In 2013, Edmon joined the newly formed PYA Analytics as its Chief Technology Officer with primary responsibility for technology vision, strategy, research and development of the technology products and solutions. Edmon is the author of seven conference papers, four technical reports, and nine professional technical publications. Edmon has been a founder and a host of the international workshop on large-scale architectures and applications for Knowledge Discovery from Data (AP-KDD).