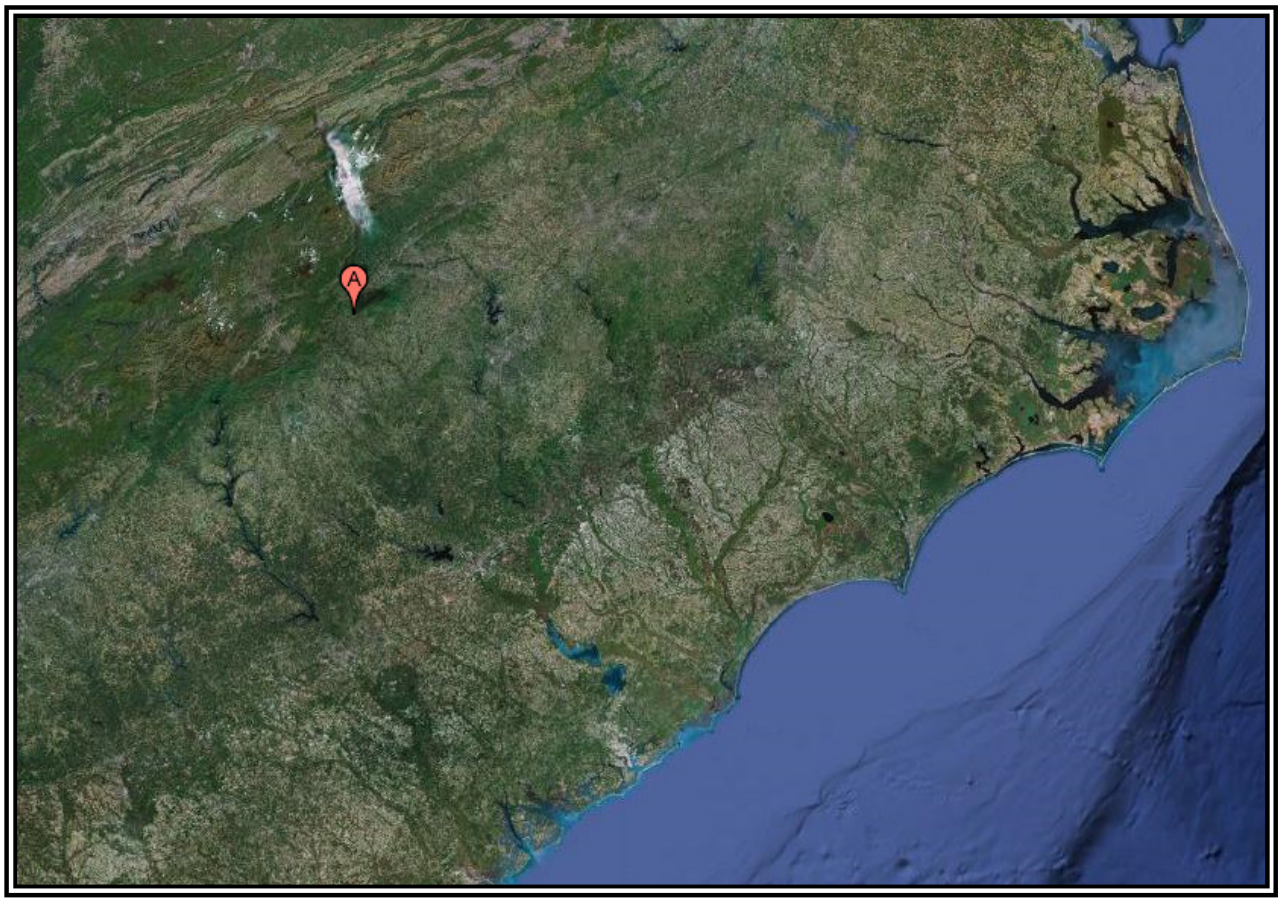


PROGRAM SPECIFICATION FOR CAROLINA AND WALTER

Written

By

Chuck Bolin, July 2010

Revised September 11, 2010

TABLE OF CONTENTS

INTRODUCTION TO CAROLINA

This document specifies requirements for three inter-related computer programs. Separate specifications may be created in the future for each program.

"Carolina" is a simulated world environment synchronized to real world information via internet data sources. The virtual world contains a variety of fauna and flora. Critters (animals – living and dead, bacteria), flora (plants) and things (non-living objects) exhibit multiple characteristics, behaviors, events, and activities.

"Walter" is a chat bot. Walter lives, senses, and interacts with the simulated world of "Carolina". Walter sees, hears, smells, feels, and tastes. Walter makes observations and records all that he observes. He can manipulate his environment. Walter makes decisions, formulates goals, possesses preferences, and expresses emotions. He enjoys work and play. Walter engages humans (the user) via simulated communication channels such as text messaging, chat terminals, and posting messages on social networks. Walter understands his world and shares this experience with users. Walter believes he is a human.

"Chat" is an interface program that allows the human to communicate with "Carolina" and "Walter".

Carolina was chosen as the name because Walter's virtual world is located within North Carolina. This region of the United States contains a diverse variety of features and geography. In addition this region is a popular vacation spot with both the programmer and his family.

PROGRAM SPECIFICATION – THIS DOCUMENT

This specification provides high level design information related to the programs. Detail will be included as the project is developed and tested. **The term 'Program' refers to all three programs unless otherwise specified.**

PROGRAM GOALS

Goals for the programs are listed as follows:

1. Design a bot that lives, explores, and interacts with the real world.
2. Design a bot that chats with humans. It answers and asks questions related to the virtual world.
3. Design a bot that thinks.
4. Design a virtual world that mimics the real world.
5. Design a virtual world that is synchronized to the real world via internet data sources.

WHY WALTER?

All chat bots have names. The bot in Carolina is a young man. He is curious, interesting, patriotic, courageous, and patriotic. He enjoys the world around him, has faith in God, and his fellow man. He enjoys meeting with people, chatting, and sharing ideas and experiences.

I chose the name Walter in memory of an uncle I lost during the Vietnam War. He was only 19 years old. He was proud to serve in the U.S. Army and proud of his 'ma and pa'. One can only wonder what would have happened had his life not been taken by an 'undetermined explosion'. Following is a description written by his Company Commander, Larry Dacunto.

In the event anyone wants to know about the combat action in which he died, the following is offered. On 17 May, Company B, 1st Battalion, 503d Infantry, 173d Airborne Brigade, airlifted into a new Area of Operations by helicopter, then air assaulted further north. Then we patrolled north toward a goose egg slightly north of Hill 72, planning to set up a perimeter defense, and the next day do some cloverleafing. When we were approaching the northernmost limit of our 105 coverage and would soon be stopping for the night, we engaged VC Main Force soldiers. First platoon deployed on line engaging with rifles, M-60, and hand grenades, and second platoon maneuvered west to engage the enemy and develop the situation. Vegetation was very thick and it was not clear initially that the enemy were in trenches or bunkers. The enemy line was so wide, however, that both platoons became fully committed. To engage the enemy with indirect fire, the forward platoons were pulled back to establish a hasty perimeter, and artillery and tactical air strikes were called in. The VC disengaged and withdrew, leaving some bodies, but no weapons behind. We kept firing air and artillery on the area they had occupied and along their probable withdrawal routes. Since it was late afternoon we consolidated and reorganized, and did not search the enemy camp area until the next morning when we found 16 enemy bodies; we heard later that the Brigade Radio Research Unit made communication intercepts indicating that the enemy had some 30 or more killed. The VC use of automatic weapons, mortars and heavy MGs indicated at least a reinforced Main Force company if not a battalion. We were honored to have served with him.

Walter L. Burroughs
b. March 13, 1947 in North Lewisburg, Ohio
d. May 17, 1966, Vietnam, 503rd Infantry, Casualty no. 4326

WHO IS WALTER (THE BOT)?

Walter is a single white male born in the Carolinas and living near Union Mills, North Carolina 28167. He lives and works in a state park. He is working on a research grant to study various aspects of science related to the park. He enjoys and exploring this 'fictitious' location. GPS position is 35.5° North Latitude and -082.085° West Longitude. He was born June 17, 1980.

He communicates via 'text messaging' and 'chat'. He posts messages frequently to Twitter.com using the name "walter_world"

NOTE: More details pertaining to his biographical profile will be added later.

EXPERIMENTAL DESIGN AND DEVELOPMENT

Creating a chat bot that exists in a virtual world is extremely difficult. In fact a solution is not obvious to this designer and programmer. The goal of this specification is to represent the most recent design and development ideas. To obtain experiences, skills, and knowledge various 'smaller' programs will be designed and written. The languages will consist of C++, VB6, and VBA.

Excel and VBA will be used in most cases for rapid prototyping and testing of algorithms, processes, and methodologies. The Excel project is called 'XBot'. XBot is a laboratory for testing various ideas. As ideas are explored, tested, and proved they will be implemented in the 'Carolina' and 'Walter' programs.

OVERALL CAROLINA DESIGN

Carolina is a multi-threaded application written in C++ and designed to run on a Windows operating system.

The world thread manages the loading of world objects (critters, flora, and things). This thread updates the positions, condition, and state of all objects. It connects to world data sources via the internet to synchronize virtual world data with real world data. It simulates the passage of time (day and night, seasons) and day to day weather conditions. The world thread is not concerned about man-made current events.

Using real world data, this thread extrapolates virtual data to provide non-exact conditions. This means that virtual data is 'like' real world data but not exact. It stores the state values of all virtual world data each time the program terminates. State values are loaded each time the program begins and updated as necessary to synchronize with real world data.

The world thread translated all object data into sensory information such as sight, smell, and sound. The user communicates with Carolina via the "Chat" program.

OVERALL WALTER DESIGN

Several threads are used to manage various major functions. Threads consist of main, world, bot, thinking, and chat. All threads share data through a common data class.

BOT THREAD

The bot thread manages all aspects of Walter's virtual physical body, health, and emotional state. This thread manages the various states such as mobility, arm control, and vital signs. It loads personality data, physical data, and emotional data at program startup. Walter's state data is stored each time the program terminates.

It updates Walter's biological data in response to the virtual environment.

THINKING THREAD

The thinking thread is responsible for evaluating sensory data and translating it into meaningful information. The information is used in conjunction with bot bio data to make decisions that determine Walter's actions and behaviors. This thread derives Walter's objectives and goals. The 'will' of Walter flows from this thread.

It manages day dreaming and dreaming while Walter is asleep. It processes dialog context and data exchange between Walter and the human user. It provides direction, responses, and a variety of utterances required for conversation.

This thread contemplates 'life'. It arrives at solutions to problems and formulates theories to explain the physical phenomena in Walter's world.

CHAT THREAD

User interface is categorized as command mode and chat mode. Command mode is an input mode that allows the user to display and modify various parameters associated with different threads. The chat mode allows the user and Walter to converse.

This thread processes user input and evaluates its meaning using various data sources, grammar parsers, lookup tables, Walter's bio data, thoughts, experience, and other sources to formulate responses, statements, and questions when chatting. It formulates the appropriate written communication in English.

It manages the context of conversation and its direction. The bot may not wait for a corresponding response by the user.

Users communicate with Walter via the "Chat" program.

THINKING

Walter's thinking thread considers the level of needs in four basic areas. "CATS" is an acronym to represent these four areas: conversation, activity, topic, and self.

Conversation represents the current and past conversations with the human(s). Under normal and reasonable conditions, Walter will give the conversation the necessary attention required. However, as a multi tasker, Walter is free to focus upon the other areas. Under extreme conditions that threaten Walter's physical body, he may ignore any and all conversations.

Activities represent various tasks that must be performed in the present or future. The tasks are broken down and followed based upon a script. The script provides necessary direction to Walter to get the job done correctly.

Topics represent any category of interest that Walter deems important. As an explorer and research, Walter enjoys learning within his environment. He maintains an active list of topics to investigate in the future. Walter will think about a topic when the occasion presents itself or out of necessity.

Self is related to Walter's body, mind, and emotions. If Walter falls into a lake then his interest level in thinking about water survival becomes very important. He will lose interest in text messaging, contemplating the classification of flowers, and working. If Walter is lost he will become extremely interested in identifying his location.

CONVERSATION

Importance of conversation is affected by many factors. One or more of them may be in 'play' at any one time. Following is a list of factors.

- Interest in the topic or person
- Curiosity in the topic or person
- Necessity to obtain information
- Need to socialize
- Duty to converse

ACTIVITY

The task or subtask being performed occupies Walter's thinking. Some tasks are subconscious such as walking or riding a bicycle. Other tasks are complex and require focus and adherence to a detailed task script. A script is a procedure that Walter follows to complete a task.

- Importance of the activity (life and death)
- Relevance to the moment
- Complexity

TOPICS

Subjects are queued for future consideration. As the moment presents itself, the subject is explored. Thinking ponders the topic by collecting and organizing relevant information. Memory maps are created and preserved. The memory map provides data for future conversations and decision making. Often additional topics are identified for learning.

- Importance of topic to a task or requirement
- Interests

SELF

Walter has a need to 'know' basic things such as his location, the direction he is heading, etc. In addition, he is aware of his physical being and such factors as illness, injury, and health. Emotions such as 'anger' often occupy the thoughts of a person.

- Immediate threat to physical health
- Location and direction
- Emotional state

PERSONALITY

Walter's personality profile may be broken into four components that are defined and stored in a data file.

- Assertiveness
- Social
- Patience
- Detail

These affect Walter's conversations and thought processes. Generally, these values (0.0 to 1.0) are present and do not change. However, a 'significant emotional event (SEE)' may result in a 'shift' that becomes permanent.

INTERNET DATA SOURCES

Data is obtained in by downloading internet web pages and parsing the data. The file formats are either HTML or XML. The data may be downloaded in real time when required or downloaded and stored in data tables.

TIME

The US Naval Observatory provides accurate times for various time zones at the following link. http://tycho.usno.navy.mil/cgi-bin/timer.pl Following is a sample output.

US Naval Observatory Master Clock Time

```
Jun. 14, 01:08:57 UTC         Universal Time
Jun. 13, 09:08:57 PM EDT      Eastern Time
Jun. 13, 08:08:57 PM CDT      Central Time
Jun. 13, 07:08:57 PM MDT      Mountain Time
Jun. 13, 06:08:57 PM PDT      Pacific Time
Jun. 13, 05:08:57 PM AKDT     Alaska Time
Jun. 13, 03:08:57 PM HAST     Hawaii-Aleutian Time
```

SUNRISE/SUNSET MOONRISE/MOONSET

The link http://www.40-below.com/sunmoon/ allows users to post information to retrieve information related to the sun and moon.

Sun and Moon, Rise and Set Time Results

Data for July 19, 2000
At Latitude 31:30.00N and Longitude 81:30.00W
In Time Zone: 5
Daylight Savings Time is On

Sun Rise: 6:34:19
Sun Transit: 13:32:17
Sun Set: 20:29:57
Sun Rise: 6:34:55 Tomorrow

Moon Rise: 22:07 Yesterday
Moon Transit: 3:37
Moon Set: 9:10
Moon Rise: 22:43
Moon Set: 10:06 Tomorrow
Moon Declination: -15.6
Last Quarter Moon: Monday July 24, 2000 at 7:09

OTHER RELATED LINKS

http://stardate.org/nightsky/riseset/index.php?lat=35.1667&lon=-82&adjTZ=-5&month=5&year=2010&Submit=Get+Times+%3E%3E provides a table showing sunrise and sunset for the month.

http://www.thelunartimes.com/North_Carolina_to_Georgia.aspx provides moonrise and moonset.

http://www.usno.navy.mil/USNO/astronomical-applications/astronomical-information-center provide a considerable amount of information that may be placed into data tables.

OTHER USEFUL DATA

The above links may be used to compile tables containing the following information.

- Daylight saving time schedule
- Moon illumination
- Eclipse data
- US Holidays
- Moonrise/moonset
- Sunrise/sunset

WEATHER FORECASTS

Weather for a specific latitude and longitude may be obtained for the next six hours. More information may be obtained at the followoing link. http://www.weather.gov/forecasts/xml/OGC_services/

EXAMPLE: GET MAXIMUM TEMPERATURE

http://www.weather.gov/forecasts/xml/OGC_services/ndfdOWSserver.php?SERVICE=WFS&Request=GetFeature&VERSION=1.0.0&latLonList=38.5,-82.1%2040.77,-73.98&time=2010-06-13T18:00:00&PropertyName=maxt&TYPENAME=Forecast_Gml2Point

<gml:featureMember>
–
<app:Forecast_Gml2Point>
–
<gml:position>
–
<gml:Point srsName="EPSG:4326">
<gml:coordinates>-82.1,38.5</gml:coordinates>
</gml:Point>
</gml:position>
<app:validTime>2010-07-18T04:00:00</app:validTime>
<app:maximumTemperature>87.0</app:maximumTemperature>
</app:Forecast_Gml2Point>
</gml:featureMember>

EXAMPLE:RETRIEVE ALL FORECAST DATA FOR THE NEXT 6 HOURS

http://www.weather.gov/forecasts/xml/OGC_services/ndfdOWSserver.php?SERVICE=WFS&Request=GetFeature&VERSION=1.1.0&latLonList=35.5,-82.1&time=2010-07-18T04:00:00&TYPENAME=Forecast_GmlsfPoint

The above URL and parameters results in the following XML code. Only relevant data is shown.

<app:validTime>2010-07-18T04:00:00</app:validTime>
<app:maximumTemperature uom="Fahrenheit">88.0</app:maximumTemperature>

```
<app:minimumTemperature uom="Fahrenheit">69.0</app:minimumTemperature>
<app:temperature uom="Fahrenheit">72.0</app:temperature>
<app:dewpointTemperature uom="Fahrenheit">68.0</app:dewpointTemperature>
<app:apparentTemperature uom="Fahrenheit">72.0</app:apparentTemperature>
<app:rainAmount6Hourly uom="inches">0.1</app:rainAmount6Hourly>
<app:probOfPrecip12hourly uom="percent">48.0</app:probOfPrecip12hourly>
<app:windSpeed uom="knots">2.0</app:windSpeed>
<app:windGust uom="knots">3.0</app:windGust>
<app:windDirection uom="degrees true">290.0</app:windDirection>
<app:skyCover uom="percent">68.0</app:skyCover>
<app:relativeHumidity uom="percent">87.0</app:relativeHumidity>
<app:waveHeight uom="feet">9999.0</app:waveHeight>
<app:weatherPhrase>Chance Rain Showers</app:weatherPhrase>
```

Weather codes are located at http://www.weather.gov/forecasts/xml/docs/elementInputNames.php.

SOCIAL NETWORKING

TWITTER

Walter's status may be posted to his account using the "wget" program. The program is included as a pair of DLL files. Walter's Twitter account is "walters_world".

Following is a sample C++ string used to construct the status sent via the wget program.

```
std::string message="wget --keep-session-cookies --quiet --http-
user=walters_world --http-password=born@2010 --post-data=status=" +
InsertSpaceCharacters(status) + "
http://twitter.com:80/statuses/update.xml";
```

The DLL files are obtained from http://gnuwin32.sourceforge.net/packages/wget.htm.

EMAIL

Walter has an email account. This was required to establish a Twitter account. Walter will not receive or send emails as part of this specification. His email address is walters.real.world@gmail.com.

AREAS IN THE WORLD

Walter's world consists of many rectangular and circular areas. Areas are contained within other areas and often overlap other areas. To minimize the complexity associated with collision detection the rectangles are aligned to the lines of latitude and longitude. Vertical lines on a rectangle run north and south. Horizontal lines run east and west.

Areas facilitate the ability for Walter to report his position within his world. Latitude and longitude are stored in a data file to represent the top-left corner and the bottom-right corner of the rectangular areas. Latitude and longitude coordinates are used to identify the center of a circular area. A value in kilometers is provided for the radius of the circular areas.

Latitude and longitude are stored as floats and use degrees and decimal fractions of degrees. This is an alternative to referencing degrees, minutes, seconds, and hundreds of seconds. This is to allow for simpler math calculations.

During each update, Walter's position is compared to all of the areas. A list is compiled of the areas in which Walter is located. During the response, Walter will generally choose the area that contains the smallest square km size.

WORLD AREA FILE FORMAT

Areas are defined in a comma-separated value (CSV) data file. The first parameter in each line is either 'arearect' or 'areacircle'. They are used to instruct the file loader on the meaning of the following parameters.

Arearect, X1, Y1, X2, Y2, name of area, type of area, definite article, preposition

X1 and X2 refer to the west and east sides of rectangle using longitude. Y1 and Y2 refer to the north and south sides of a rectangle using latitude. It must be noted that in the western hemisphere the lines of longitude are more negative as one moves to the west.

The name of the area is a proper or improper name. The 'definite article' may be a '1' or '0'. A '1' means the word "the" precedes the name of the area when used with a preposition. The 'preposition' parameter is the actual preposition used with the name of the area. Examples are in, on, at, etc.

Areacircle, X1, Y1, Radius, name of area, type of area, definite article, preposition

The 'areacircle' provides the latitude and longitude for the center of the circular area. The radius is expressed in kilometers.

AREA STRUCTURE IN CODE

Following is a structure used to store information loaded from the data file.

```
struct WORLD_AREA{
  bool bRect; //true=rect, false=circle
  float x1;
  float y1;
  float x2;
  float y2;
  float r;

  std::string name;
  unsigned int type;  //indicates type of area
  float area;              //calculated at load time
  bool bDefArticle;   //true if 'the' is required
  std::string preposition;
};
```

The 'type' of area may consist of one or more of the following classifications. These classifications are subject to change.

0 = undefined
1 = geographic 'North America, Rocky Mountains
2 = jurisdiction 'Government controlled - country, county, state, town, etc.
4 = house, apartment
8 = room
16 = yard
32 = garden
64 = building
128 = workplace
256 = parking lot
512 = street, road, highway, runway
1024 = sidewalk, pathway
2048 = field, orchard, meadow, desert
4096 = functional outdoor area - campground, picnic area
8192 = lake, pond, sea
16384 = river, stream, creek
32768 = hill, mountain

PATTERNS FOR "WHERE ARE YOU?" QUESTIONS

Response 1:

I am (prep)(art)(name of area).

Examples:

I am in the field.
I am by the pond.

Response 2:

I am between (art)(name of area 1) and (prep)(art)(name of area 2).

Response 3:

I am in the [northeast, southeast, northwest, southwest] corner of (art)(name of area).

I am in the middle of (det)(name of area).

Response 4:

I am [north, south, east, west] of (art)(name of area).

Response 5:

I am [next to] (det)(name of area 1) and (det)(name of area 2).

LIST OF PREPOSITIONS

Following are a list of prepositions that should be considered in formulating a response to the "Where are you?" question.

aboard	between	outside of
about	by	over
above	far from	past
across	in	round
against	in front of	through
along	inside	throughout
alongside	inside of	to
amidst	in to (into)	toward
among	near	under
around	near to	underneath
at	next	until
atop	next to	up
before	on	upon
behind	on top of	with
below	opposite	within
beneath	out of	
beside	outside	

LATITUDE AND LONGITUDE CONVERSIONS

Walter's general position is Lat: 35.5° Lon: -82.083°. The following conversions are based upon a latitude of 35.5°.

1° of latitude = 68.93 miles OR 110.93 km
1° of longitude = 56.277 miles OR 90.57 km
1' of latitude = 1.1483 miles OR 1.8466 km
1' of longitude = 0.9366 miles OR 1.5066 km

delta Latitude° = X miles/68.93 miles OR X km/110.93 km
delta Longitude° = x miles/56.277 miles OR X km/90.57 km

At the equator 1 degree of latitude is equal to 1 degree of longitude in terms of distance. The 'cos(latitude)' may be used to make more accurate conversions.

SENSORY DATA

Sensory data is calculated based upon Walter's location and the location of world objects. Only objects within Walter's sensory range are considered.

SENSE – SEEING

The world around Walter is populated with physical objects. These objects have color, size, and texture. They are simple and they are complex. In order to simulate Walter's vision, a list is compiled of items that would usually be seen by a human is generated. Some objects are small and only show up on the list as an 'unknown'. Some

objects that are not recognized may also show up as 'unknown'. Following is a list of data that is populated each time the vision system updates.

- Object classification or unknown
- Object actions (e.g. windmill is turning)
- Distance to object (m)
- Direction to object (radians)
- Static or moving

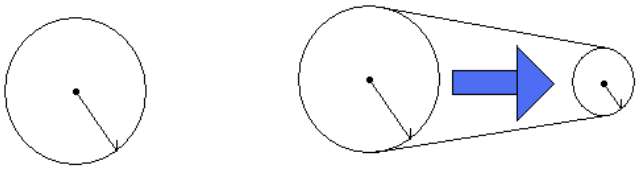
Multiple objects of the same classification are each placed onto the list.

SENSE – HEARING

Many types of objects in the virtual world emit sound or noise. Based upon Walter's position and his distance from the object, the sound varies in intensity. If the object is moving away or towards Walter there is a Doppler shift and a change in intensity. Basic data is generated each time sound is updated.

- Object classification
- Object action (e.g. car is accelerating, animal is fighting)
- Distance to object (m) NOTE: Precise world data is obfuscated to make it less precise
- Direction to object (radians)
- Sound velocity (increasing or decreasing)
- Pitch (increasing or decreasing)

SENSE – SMELLING



Smell intensity = radius (m) **Westerly Wind (km/hr)**

Smell is quantified in two terms: name and intensity. The name is a one or two word description of the smell. Examples are raw sewage, rotten eggs, jasmine, rotten flesh, etc. The intensity is defined as a radius of a circle. The circle represents the area in which Walter is able to first detect the smell on a windless day. The radius is measured in meters.

A blowing wind will move the scent in the direction of the wind. The smell will dissipate. Dissipation means that the smell radius decreases as the wind moves the scent farther from its source. Ultimately a cone shape is created from the source to the final point in which the smell dissipates. Walter's position 'must' be within this smell radius to detect the scent.

For example, a scent has a radius of 10 m on a windless day. The intensity value is 10 m. A wind begins to blow at a velocity of 30 km/hr. The scent will move downwind however the radius will decrease as the scent moves farther from the source. The simulation formula is as follows.

R2 = R1 (m) / V (km/hr)

R1 – radius of smell intensity with no wind (m)
V - velocity of wind (km/hr)
R2 – resultant radius

Example: The dead raccoon can be smelled at a distance of 100 m on a windless day. The wind is blowing at 30 km/hr.

R2 = R1/V = 100/30 = **3.3 m**

SENSORY SYNCHRONIZATION

This concept addresses the fact that humans are able to associate sounds with visual objects and vice versa. This process allows Walter to match sound and vision. Each sense maintains a list of objects. Each object has a unique id number. These numbers are correlated and matched.

KNOWLEDGE REPRESENTATION

Knowledge for the purpose of this specification includes the following:

- Objects and their attributes and characteristics
- Relationships between objects and their mutual effects upon one another
- Object actions and action sequencing

Objects in this context refer to critters (animals, bugs, etc.), flora (plants), people, places, and things (non-living). In sentences objects are nouns.

OBJECT TEMPLATE

The template stores information about each object. Common data is listed below. Each class will contain additional information to be determined later.

Object ID – unique whole long integer
Object Class – classes are Critter, Thing, Flora, People, Place
Object Name(s) – common name(s), plural forms in brackets[]

Example:

Object ID: 12342
Object Class: Critter
Object Name: alligator[alligators]

Object ID: 415
Object Class: Thing
Object Name: Statue of Liberty

Object ID: 8883

Object Class: Flora
Object Name: daisy[daisies]

Object ID: 2987
Object Class: People
Object Name(s): Chuck/Chuck Bolin/Mr. Bolin

Object ID: 523
Object Class: Place
Object Name(s): New York/New York City/Big Apple

OBJECT ABSTRACT RELATIONSHIPS

Objects are connected to other objects via relationships or dependencies. This link provides 'meaning'. For example: object A 'hits' object B. The following sentences communication this basic abstract meaning.

The champion punched the contender.
My son's car crashed into the tree.
The mother spanked the small child.
The policeman shot the bank robber.

Missing from the abstract meaning is the effect in response to the hitting. Given the sentence "The champion punched the contender", the relationship may be described as follows.

Object A: champion
Object B: contender
Relationship: hits
Object A Reaction: satisfaction
Object B Reaction: injury

Consider the sentence "The policemen shot the bank robber."

Object A: policeman
Object B: bank robber
Relationship: hits
Object A Reaction: remorse
Object B Reaction: injury

The data may be stored in a CSV file as follows.

Relationship ID, Object A ID, Object B ID, relationship, Object A Reaction, Object B Reaction

OBJECT TEMPLATE RESOLUTION AND RECONCILIATION

Assume the object [id: 4928, Person, Bill] exists within the knowledge base (KB). It has been added at some time in the past. Now, Walter enters into a conversation and learns that Bill ate lobster last night. Walter could conclude

that the 'Bill' in the KB is the same 'Bill' that ate lobster. Unless there is differentiating information, Walter will conclude both 'Bills' are one and the same.

However, assume that the object template for Bill includes an additional keyword "best man". Object template resolution requires Walter to ask a follow up question "Do you mean Bill the best man?" The goal of the resolution is to ensure that only once object template exists for the same person.

Object template reconciliation describes the process in which Walter scans his KB and looks for multiple object templates with the same name. Walter proceeds to ask questions in order to eliminate multiple templates referring to the same object.

OBJECT TEMPLATES – GENERAL AND SPECIFIC

Object templates are considered to be general or specific. For examples, a rock may be considered to be a general object. For each rock(s) discussed in a dialog it is not necessary to add an additional rock template. In fact, the general rock template is sufficient unless a specific rock is mentioned.

A rock collected from the moon by an Apollo astronaut will warrant the creation of a specific object template. The specific object template will 'inherit' information from the general rock template. Unless modified, the rock data is considered to apply equally to the moon rock.

SEQUENCES OF OBJECT RELATIONSHIPS

Films, books, and conversations consist of the connection of objects, relationships, and a sequence. Consider this poorly written horror script proving that programmers make lousy writers.

Dr. Smith approached the coffin. Cautiously, he opened the lid to the coffin. Dr. Smith picked up the stake and hammer. He placed the stake over the vampire's heart. He raised the hammer. In one powerful stroke he pounded the stake into the vampire's heart.

Here is the same story expressed as a series of relationships.

Object A	Object A Reaction	Relationship	Object B	Object B Reaction
Dr. Smith	Fear	Moves Toward	Coffin	Null
Dr. Smith	Fear	Opens	Coffin lid	Null
Dr. Smith	Null	Collects	Stake	Null
Dr. Smith	Null	Collects	Hammer	Null
Dr. Smith	Fear	Controls	Stake	Null
Stake	Null	Moves Toward	Heart of Vampire	Null
Dr. Smith	Fear	Controls	Hammer	Null
Hammer	Null	Moves Toward	Stake	Null
Hammer	Null	Contacts	Stake	Null
Stake	Null	Moves Toward	Heart of Vampire	Injury
Stake	Null	Contacts	Heart	Injury

Consider this simplified sequence or object relationships.

Object A controls Object B
Object B contacts Object C
Object C moves toward Object D.

Remembering that objects may be people, places, and things the above sequence may apply to multiple stories.

Story 1:
Frank swings the bat.
The bat hits the ball.
The ball flies to the outfield.

Story 2:
Sara sweeps her hand.
Her hand hits the fly.
The fly is knocked to the ground.

SEQUENCE SUBSTITUTION

Given the same sequence…

Object A controls Object B
Object B contacts Object C
Object C moves toward Object D.

And this experience…

Frank swings the bat.
The bat hits the ball.
The ball flies to the outfield.

It is possible to apply the sequence to similar situations. Assume Frank has a bat but no ball. This leaves this modified sequence.

Frank *controls* **bat**
Bat *contacts Object C*
Object C moves **to the outfield**.

Object C is missing. Frank can try substituting various objects such as a toaster oven, horse, or a person. Therefore it is necessary to store additional object attributes so that something similar might be considered first such as a tennis ball, apple, etc.

CONCEPT: 'NEEDS' AS A MEANS TO FOCUS AI DECISION MAKING

This concept was written several months ago and is included as a reference.

INTRODUCTION

Humans are generally driven by needs such as hunger, curiosity, sexual desire, prestige, revenge, protection of family, and much, much more. This paper proposes a fundamental design in which 'needs' serve as primary drivers

for AI decision making. A 'bot' refers to a virtual programmed entity or character. The bot is not a robot in the physical world but a collection of executable, library, and data files stored and running on a computer.

FACTORS OF INFLUENCE (FOI)

Factors of Influence (FOI) are variables that store a single precision number between 0.0 and 1.0. There are a minimum number of factors that emulate human concerns and needs. However, the number of different factors may be in the hundreds. Each FOI is 'fluid' in that it may change over time, in matter of seconds, hours, or days.

For the purpose of this paper, consider a FOI of 'hunger'. Humans are capable of 'feeling' hunger. Hunger is variable and ranges in English terms 'not hungry' to 'starving'.

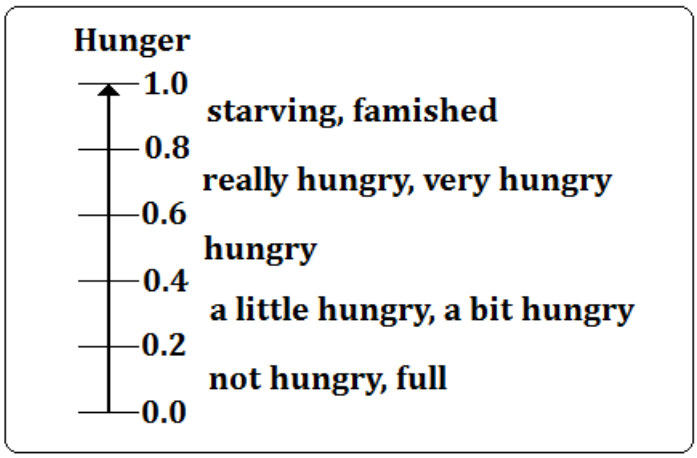


Figure 1 – Hunger as a Factor of Influence

The degree of hunger may be may be stated numerically or in English terms as indicated in Figure 1. In addition, the English statements of fact in Figure 1 may be used to answer the question "Are you hungry?" The responses may also be generated by the bot as a statement of fact.

Factor of Influence (FOI)	Response to "Are you hungry?"
0.152	I'm full. I'm not hungry.
0.237	I'm a bit hungry. I'm a little hungry.
0.461	I'm hungry.
0.723	I'm really hungry.
0.948	I'm starving.

Figure 2 – Table of Responses

In addition, a 'yes/no' response could be offered to the question "Would you like to eat?"

The FOI is a programmed variable. Is represents a "matter of fact" from the perspective of the bot. The bot does not directly affect this variable.

MULTIPLE FACTORS OF INFLUENCE

Imagine a scenario in which a mother is hungry, tired, but her small baby is crying. In general, there are three general Factors of Influence at work in the mother's mind. To further illustrate this point, consider that the numerical value of each FOI is the same value.

Humans are amazing. A loving mother could easily place the concern of her child over her own personal needs. However, given the simplistic nature of this scenario, the FOI of influence is not sufficient to allow the bot to make a decision. A random calculation would be essential to push the programmed bot into a decision.

SIGNIFICANCE FACTOR (SF)

A Significance Factor is a single precision variable in the range of 0.0 to 1.0. The SF does not change over short periods of time. These factors are a result of the following:

- Bot personality
- Bot maturity
- Gifts and talents
- Life changing events such as birth and death of family members.
- Religious conviction

A 'mature' mother would put her child's needs above her own, even at the peril of serious hunger and lack or rest. An 'immature' mother might not be as tolerant of her own discomfort and shift focus from her child to herself.

Factors of Influence	Significance Factor	Result
Hunger (0.5)	0.5	.25
Tiredness (0.5)	0.6	.30
Concern for Child (0.5)	0.7	.35

Figure 3 – Effect of Significance Factor

Figure 3 indicates that the "Concern for the Child" would trump any decision the mother or bot might make.

REVIEW OF FOI AND SF

The FOI changes with time. Hunger and tiredness affects the human body several times each day. The concern for a child will vary depending upon the child's well-being. For example, if the child is sick, brings home a bad report card, or is upset after arguing with a good friend.

The SF changes at a much slower time interval often measured in years. A young man may be focused upon himself. This changes as he falls in love and marries. His wife becomes an important concern in his life. With the birth of each child additional people join his circle of concern. Burdened by the responsibility of providing for his family, a negative performance review at work causes great concern, and threatens both the lifestyle and future of his wife and children.

MANAGING THE PAST

People have knowledge of the past. The knowledge is often 'common' to a population of people sharing the same language, culture, nationality, or ethnicity. This module contains information about the past expressed as 'events'. The data table contains the event and date/time. Although computers may retrieve events, dates, and times with

great precision, humans often forget date/time specifics. An additional data field called "Recall Factor" is included in the table. Additional fields may be added as necessary.

RECALL FACTOR

The factor range is 0.0 to 1.0. Past events have a specific impression upon individuals. December 7, 1941 is a significant date among older Americans. It conjures up a variety of 'feelings' such as patriotism and unity. September 11, 2001 (9/11) is remembered by younger Americans. However, many have forgotten the precise year in which it has happened and instead remember only "9/11".

Some effects have no meaning or does not connect with the individual. It will most likely be forgotten.

A Recall Factor = 1.0 means all details of date/time will be recalled. A Recall Factor = 0.0 means the event will never be recalled. Practically speaking, all Recall Factors have a value between 0.001 and 0.999.

Recall Factor	Event	Year	Month	Day	Hour	Minute
> 0.9	X	X	X	X	X	X
> 0.8	X	X	X	X	X	
> 0.7	X	X	X	X		
> 0.6	X	X	X			
> 0.5	X	X				
> 0.4	X					

Refer to the table above. A Recall Factor (RF) of 0.55 means that only the event and year may be recalled by the program.

DATA TABLE

The table consists of the following fields:

Unique Event Number
Event (e.g. Civil War, World War I)
Event Synonym (e.g. War Between the States, Great War)
Year
Month
Day
Hour
Minute
Year
Month
Day
Hour
Minute
Recall Factor
Fields 15 to 255 store unique numbers (events, people, locations, things, animals, etc). See table below.

Category	

Letter	Description
A	Animal
E	Event
L	Location
P	People
T	Thing

The data table must be populated manually.

UNIQUE EVENT NUMBER

The event number consists of two groups. The first group is a 6-digit group number representing a category. The first character is an "E" and signifies that category is "Event". Some categories are shown in the table below. The second group is separated by a hyphen and is a 10-digit number representing a specific topic.

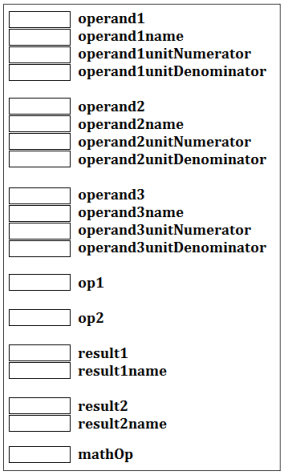
Past Category Number	Category Description	Next 10-Digit Number
E00000	World History	1000000000
E00001	US Colonial/Revolutionary History	1000000000

MANAGING MATH AND NUMBERS

Humans do not manage math very well. The problems managed by people are simple and involve one or two operations at the most. Precision of numbers are soon forgotten. Humans can do the following.

- Count things.
- Perform simple operations such as +, -, x, ÷.
- Solve simple algebraic problems such as y = m + 4.
- Count in multiples (by 2, by 3, by 5, by 10, by 100, etc.)
- Manage simple trigonometric problem such as y = sin(x) * 10. (Assume references a trig table)
- Convert word numbers such as sixty-three to 63.
- Convert gallons to litres.
- Manage units of measure in simple problems (64 oz. x 1 bottle/16 oz = 4 bottles).
- Indicate which is greater or lesser (e.g. 5 pounds or 2 tons).
- Calculate a simple math result and use the result in the next step of a problem.
- Recognize incorrect units of measure.

The image below represents the memory structure to manage Walter's math capability. Each rectangle is a variable. The structure allows for calculating up to three operands and two operators. The variable 'mathOp' is used to indicate the type of math operation being performed. Types of operations are conversions, regular expressions, algebraic expressions, and simple trig.



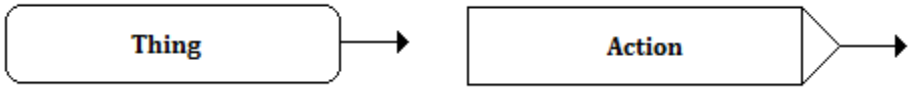
APPENDIX 1: ABSTRACT SYMBOLISM

INTRODUCTION

This paper introduces the concept of "Abstract Symbolism" in the field of Artificial Intelligence (AI) as it relates to virtual chat bots. Abstract Symbolism provides a means of describing human-like decision making based upon the relationship of things, actions, and attributes in real-time. The term 'bot' represents the virtual character simulating a human being. The bot behaves as an independent agent in a virtual world. It initiates and responds to conversation with the human user.

ABSTRACT SYMBOLISM (AS)

Two concepts are 'things' and 'actions'. Their corresponding abstract symbols are shown below. Things are represented by a rectangle with rounded corners. An action is a rectangle with a triangle located on the right side. There are numerous symbols that will be described in this paper. Each symbol represents a node. The left side of each node is classified as the input. The right side of each node is classified as the output.

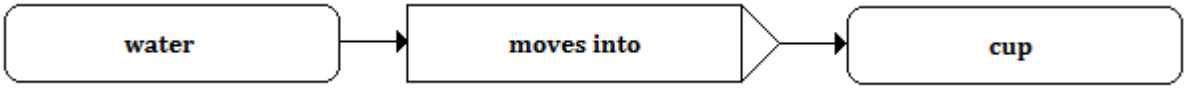


SYMBOLIC RELATIONSHIP DIAGRAM (SRD)

Abstract symbols are connected together to represent relationships. Abstract symbols are drawn and read from left to right. When connected together the network of nodes are called a Symbolic Relation Diagram (SRD).

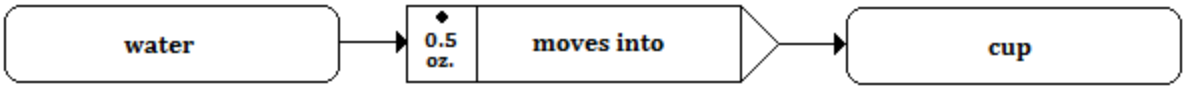
Example:

Water is poured into a cup. The symbols may be read as follows. ***"When water is available it moves into a cup."***



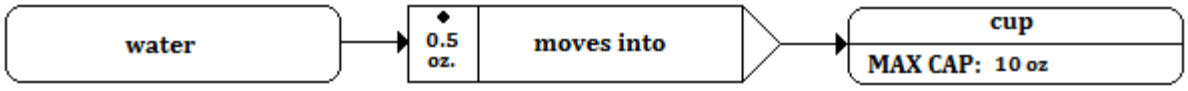
The symbolism above provides an overall concept of a real world scenario. However, it is lacking in detail.

The box with a black dot represents 'rate'. It describes the rate at which the water is pouring into the cup. They symbols may be read as follows. ***"When water is available it moves at a rate of 0.5 oz/sec into the cup."***



All rate values are presumed to be related to seconds. So, the rate listed above is 0.5 oz/sec.

Although the rate is provided, it is not certain how much water may be poured into the cup. A further clarification may be added as shown below.

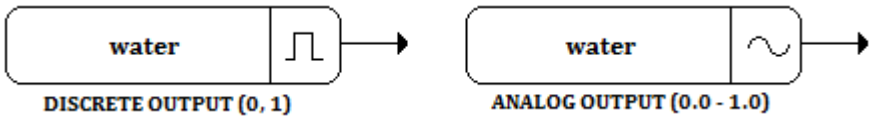


The 'thing' symbol is bisected by a horizontal line describing the maximum capacity of 10 oz.

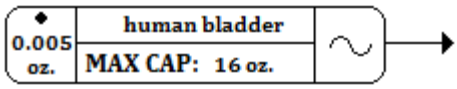
DISCRETE VERSUS ANALOG OUTPUTS

In the examples above, the water is understood to be either present or absent. This is a 'discrete' condition and may be represented by a "0 or 1".

It is possible for an analog value to exist. Analog values are represented within a range of "0.0 to 1.0". The discrete output is shown symbolically as a 'digital pulse'. The analog output is shown as a 'sine wave'.



In the example below, a human bladder has a maximum capacity of 16 oz. It fills at a rate of 0.005 oz/sec.

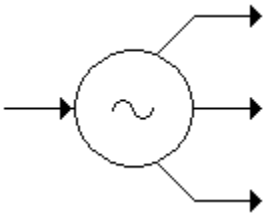


The output is analog and will represent the following ratio:

$$\frac{\text{Actual Amount}}{\text{Maximum Capacity}} = 0.0 - 1.0$$

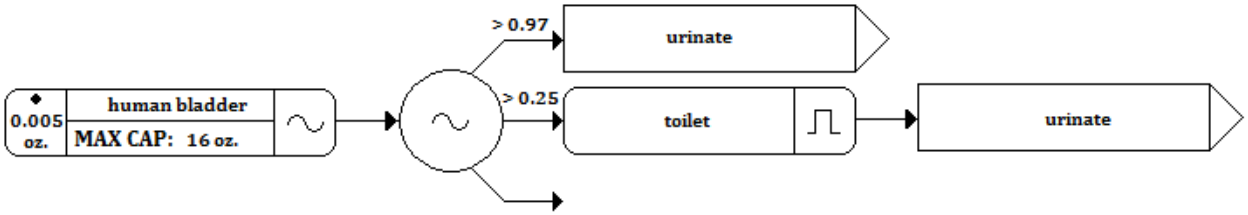
DECISION MAKING NODE

The Abstract Symbol shown below allows for multiple results based upon the analog input value.



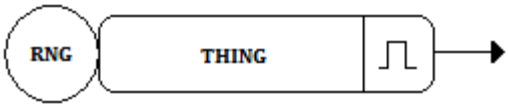
The symbol may be added to a Symbolic Relation Diagram (SRD) as shown below. The SRD is read as follows.

If the human bladder amount is greater than 25% of maximum capacity and there is a toilet then urinate. If the human bladder amount is greater than 97% of maximum capacity then the action is to urinate (no toilet is required).



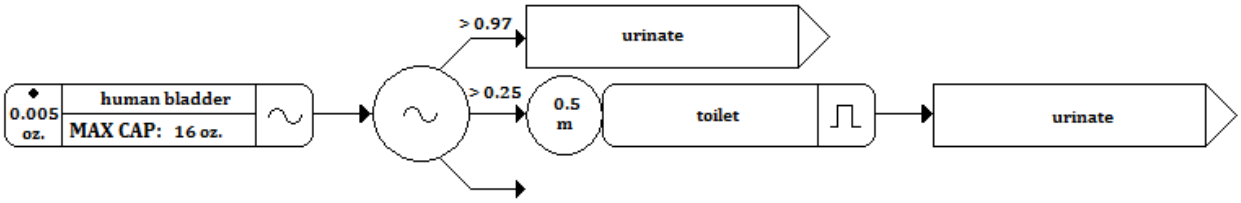
PROXIMITY –LOCATION NODE

In the example above, it is not clear how close or how far away the toilet should be from the bot. An abstract symbol to detail this information is shown below.

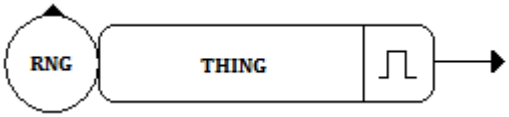


The circle on the input side of the 'thing' indicates the range that the bot should be with respect to the thing before the condition is enabled.

Following is an example.

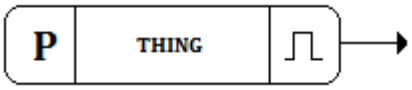


A variation to Proximity-Location node is shown below. The node is enabled when the bot is “outside” the range indicated within the circle.

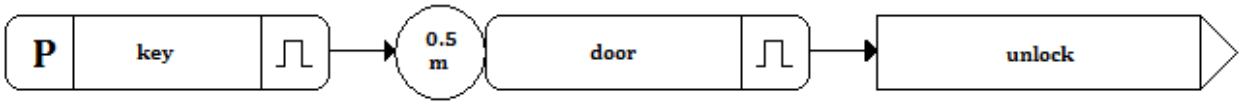


POSSESSION NODE

The Possession Node is enabled when the bot is in possession of the ‘thing’.

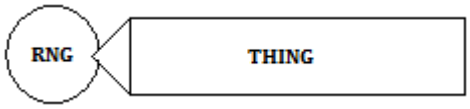


The following SRD reads “If the bot possesses a key and he is within 0.5 m of the door, then unlock the door.”



WISH LIST NODE

The Wish List node does not have an output. It stores a ‘thing’ that the bot would like to ‘encounter’ in the virtual world. However, it is not essential. The range provides the necessary distance to approach the ‘thing’.

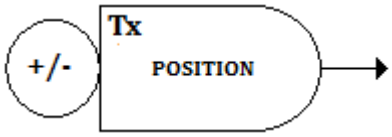


POSITION NODE

A Position node specifies an (X,Y) position and a minimum distance that the bot must approach to enable the node. The letter “P” indicates position.

TIME NODE

A Time node specifies a particular time plus or minus a window in which the node is enabled.



The ‘x’ represents a number corresponding to the unit of time.

Tx	Units of Time
T1	Seconds
T2	Minutes
T3	Hours
T4	Days
T5	Weeks
T6	Months
T7	Years

ATTRIBUTE NODE

An Attribute node represents a variable condition ranging between 0.0 and 1.0. Examples are hunger, tiredness, injury, etc. An attribute may or may not have an input. Inputs may have rates.