

“SCRIPT WRITER” - ENVIRONMENT FOR AUTOMATIC TEXT ANIMATION AND SIMULATOR OF HUMAN IMAGINATION

Vadim Astakhov, Tamara Astakhova
University of California San Diego

ABSTRACT: *Imagination is the critical point in development of realistic artificial intelligence (AI) systems. One way to approach the problem would be simulation of its properties and operations. We have started the development of software that should help us to emulate the process of imagination through a text animation. The purpose of this paper is demonstration of our model that based on several proposed concepts such as Actor-Action text filtering, Ontological Graph Holographic Blending and Ontological Confabulation. We are presenting “ScriptWriter” software-simulator of human imagination that automatically generates an animation from a simple English text.*

Keywords: *imagination, text processing, artificial intelligent, animation*

1. ARTIFICIAL INTELLIGENT (AI) AND PROCESS OF IMAGINATION

Human are exceptionally skill-full in integration of different visual, audio and kinesthetic inputs to create new emergent structures, which results in new tools, technologies and ways of thinking. Philosophers Gilles Fauconnier and Mark Turner proposed term “two-sided blending” to describe that process [1].

Even in the absence of external stimulus, the brain can run imaginative stimulation. Some of these are dreams and imaginative stories but imaginative process always at work in even the simplest construction of meaning.

One big area of human behavior mutually influenced by imagination is human “beliefs”. The statement “person X believes Y” is equivalent to a series of conditional statements that asset how X would behave under certain circumstances. In that sense, beliefs are none observable entities case some observable relations in human behavior. Those none observable entities originated from

facts of observation, from memory, from self-knowledge, and from experimentations. All of these blend in imagination and lead to “a belief” through individual inference rules. Also, the result of merging and blending of various concepts are always imaginative and it’s easy to realize that the imagination is always at work in ways that consciousness does not always catch the process. Consciousness usually catches only a product of what the mind is doing. Most specialists in area of science, engineering, economy and politics have impressive knowledge but are also unaware of how they are thinking. And even though they are experts, will not find out just by asking themselves. It leads to obvious conclusion that *Imagination* is the crucial feature of the creative human mind. It provides a basis for other mind functions. To build really powerful AI system we should address this issue.

We analyzed and implemented some aspects of imagination such as integration and identity in the software. We see integration as a finding identities and oppositions as parts of more complicated process which has elaborate structural and dynamic properties of a concept. It typically goes entirely unnoticed since it works fast in the backstage of cognition. On the other hand the identity is the recognition of identity and equivalence that can be mathematically represented as $A=A$. It is a spectacular product of complex, imaginative and none-consciousness work. Identity and none-identity, equivalence and differences are apprehensible in consciousness and provide a natural beginning place for formal simulation approach. Identity and opposition are final products provided to consciousness after elaborate unconsciousness work and they are not primitive starting point.

These operations are very complex and mostly unconsciousness for human but at the same time play basic role in emergence of meaning and

consciousness. From everyday experience of meaning and human creativity we can conclude that meaning and basic consciousness operation lies in the complex emergent dynamics triggered in the imaginative mind. It seems reasonable to imply that consciousness and mind prompt for massive imaginative integration.

We believe that simulation of imagination is a first step for building powerful AI system. To accomplish that step pluggable architecture “ScriptWriter” was developed. That provides us with ability to simulate imagination through the process of text animation.

2. ONTOLOGICAL MODEL FOR BLENDING PROCESS OF IMAGINATION

Concepts: To construct an algorithm and dynamic system which emulates mental functions fundamental theoretical units must be chosen. We propose term “concept” to represent an object occupying space and time, object attribute specifying what an object is or does and relations between objects. Example: concepts “woman”, “walk”, “beach” can lead to conceptualization “Woman walk on the beach” that will lead to animated image of a woman walking on the ocean beach. This conceptualization will imply many “beliefs”. One possible belief here can be - “the woman wears something”. Many of us intuitively “believe” that people usually wear something when they “walk” if opposite not mentioned. That “belief” will cause imagination of many people provide image of the woman walking on the beach and wearing some cloth even if nothing was mentioned about cloth. It is totally different case where we have sentence: “naked person is walking of the beach”. That imply to some rules in the system such as:

IF X like Y then X seeks Y.

IF Y disturb X then X avoid Y

But sometimes such rules can be in conflict that leads to emergence on new blending structures.

Universal Structure of objects in the scene: A participating in the scene entity is assumed to be in one of the three states (Active Actor, Passive Actor and Action) with a binding pattern for every disclosed relation. Every entity’s element

(relations and attributes) has a descriptor for keyword search, and a so-called semantic-type that can be used to map the element to ontology. For example: Relation – behind (far behind) has a type-position/orientation. Another example: Attribute – red has type-color.

Further, an entity may disclose a set of functions that are internally treated as relations with the binding pattern (b, f) where b represents a set of bound arguments and the single f is the free output variable of the function. For example, “take the ball” can be treated as a human specialized function which used to raise human actor hand in a set of specified scenes. Such function will depend on set of binding parameters “b” those characterize position of the ball and human hands before action and return “f”-position of the hand at specified moment of time. Such model let us treat animation as a Petri Net dynamics with computations where actors-nodes take different states in time.

Mental Space: We use a term “mental space” as small packets of concepts which are constructed as we think and talk. Also, we have *Conceptual integration* as critical part of imagination. It connects input *mental spaces*, project selectively to a blended imaginary space, and develops emergent structures through composition, competition, and elaboration in the blend.

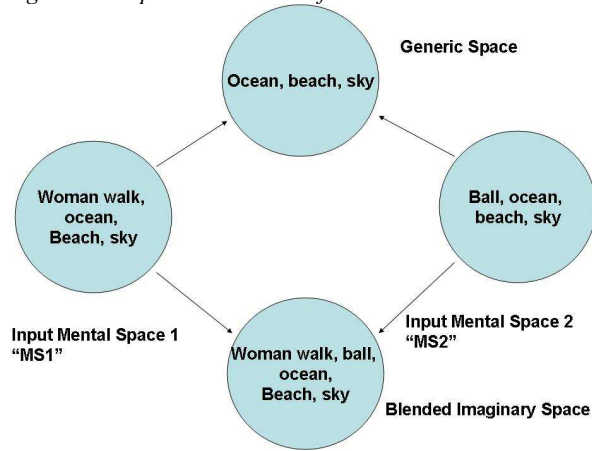
For example, set of sentences: “The blue ball was left on the beach. Woman walks on the beach”, imply two impute concept spaces “Woman walks on the beach” and “ball left on the beach”.

We perform *Cross-Space mapping* which connect counterparts in the input mental spaces and then construct *Generic Space* that maps onto each of the inputs and contains what the inputs have in common: beach, ocean, and horizon. The final *blending* does projection of the ocean beach from the two input mental spaces to the same single beach in blended imagination space.

Such blending develops emergent imaginative structures those are not present in the inputs like “woman walks toward the ball” or “woman walks relatively close to the ball”. It seems intuitively obviously that imagination can create integrated scene with all mentioned objects as a result those two sentences. Fig.1 represents input mental

spaces for two sentences text, generic space and resulting blending space that is a result of imagination process.

Fig.1 Mental spaces constructed for two sentences.



To keep semantic relations among concepts, we have designed ontological database. By using ontological information we can integrate, and interrelate various objects in the scene without having to examine the low-level details.

Ontologies: Ontology is a term-graph whose nodes represent terms from a domain-specific vocabulary, and edges represent relations that also come from an interpreted vocabulary. The nodes and edges are typed according to a simple, commonly agreed upon set of types produced by test bed scientists. The most common interpretation is given by rules such as the transitivity of *is-a* or *has-a* relations, which can be used to implement inheritance and composition. However, there are also domain specific rules for relationships such as *region-subpart* (rock-region -> mountain-region) and *expressed-by* (emotion-state -> face) that need special rules of inference. For example, if a rock-region participates in imaginary scene (such as “he climbs the rock”) and the human-emotion is expressed-by a face, then the climbing will case emotion that will be expressed on the face.

In the current *ScriptWriter* framework, ontologies are represented as a set of relations comprised of a set of nodes and a set of edges, with appropriate attributes for each node and edge. Other operations, including graph functions such as path and descendant finding and inference functions

like finding transitive edges are implemented in java.

We build ontology by extracting pair wise relations between English words like “head –part of -body”. We also assign wait for each relation that reflect probability to have two words in one sentence or in two concurrent sentences. To perform a calculation we have collected test cohort of the fiction texts. The probability was extracted as a frequency of pair wise occurrences of the two specified words.

Ontological dK-series and dK-graphs. Ontological graph created the way described above is dependant on cohort of the text and can not pretend to be generic enough. Also, even for small dictionary of English words is extremely complex. Here, we need a way to approximate properties of generic ontological graph that can be build on limited text cohort but capture topological properties of generic English text.

To capture such complexity of graph properties, we use dK-graphs approach [6]. This approach demonstrate that properties of almost any complex graph can be approximated by the random graph build by set of dK graphs :0K, 1K, 2K and 3K where “K” is notation for a node degree and d-for joint degree distribution that d node of degree “k” are connected.

Based on our text cohort, we estimate 0K-average node degrees as average frequency of a word, 1K –node degree distribution as frequencies for the words in the text cohort, 2K and 3K – joint degree distribution were extracted as pair wise and triple-wise frequencies of having two/three words in two/three sub-sequent sentences. Those values were assigned for each dK graph of our ontology and later used during operation of *ontological confabulation*.

Term-Object-Map: We have a specific source that is called the *term-object-map* and maintains a mapping between ontological terms and 3D-animation objects library developed for Maya animation environment. These objects are used by the system to build an animation.

Mapping Relations: In the industry, animation specialist tries to capture the requirements of the script at hand. This leads to the pragmatic problem such as the relationships between attributes disclosed by different objects and between object

parameters are, quite often, not obvious. To account for this, the system has created additional *mapping relations*. Currently there are three kinds of mapping relations.

- The *ontology-map* relation that maps data-values from a object to a term of the ontology
- The joinable relation links attributes from different objects if their attribute types, relations and semantic types match.
- The *value-map* relation maps a fuzzy parameter value (speed fast) to the equivalent attribute-value disclosed by the animation software.

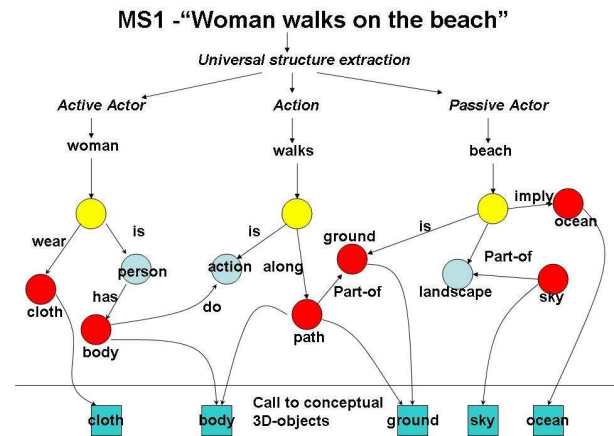
3. UNIVERSAL TEXT FILTERING AND dK-ONTOLOGY CONFABULATION

Text filtering: ScriptWriter use simple English text as an input and generate output animation. First, it performs text processing to extract semantic relations among words in the sentences. Mental space is created for each sentence. For example, we consider simple text of three sentences: “Woman walks on the beach. The blue ball was left on the beach. Woman takes the ball”.

Fig.2 represents details of the first mental space that is built from the sentence “Woman walks on the beach”. The sentence is processed and *Universal structure* is extracted: “Active Actor”-woman, “Action”-walk, “Passive Actor”-beach. Instances of the universal structure (woman, walk and beach) are anchored (colored by yellow) to ontological graph that represents relations among concepts. We perform graph expansion operations to integrate all relevant objects (colored by red) that are required for the mental space such as *ocean*, *sky* and *woman cloth* which are not mentioned in the sentence.

Each concept such as “beach”, “woman”, “ball” represents sub-graph that connects all concepts relevant to specified terms.

Fig2.Part of ontological graph that represent mental space for sentence “Woman walks on the beach”



Same operations are performed for the second and third sentences (Fig3 and Fig4 respectively).

Fig3.Part of ontological graph that represent mental space for sentence “The blue ball was left on the beach”

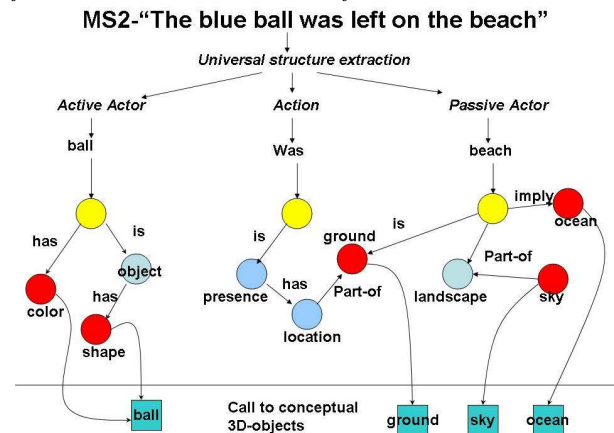
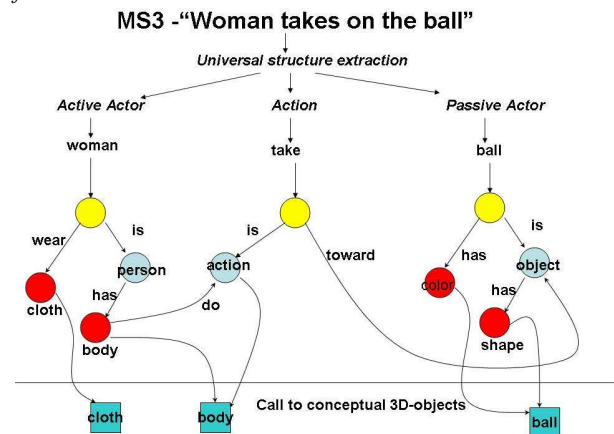


Fig4. Part of ontological graph that represent mental space for sentence “Woman takes the ball”



Ontological Graph Holographic Blending And

Generic Space: Generic space is constructed by mapping of objects such as “woman”, “beach” and “ball” which are co-occurred in different *mental spaces*. Start from those concepts we perform node expansion as a procedure of finding neighbor concepts connected to generic concepts for example: “color” of the ball and “body” of the woman. We perform node expansion by using various relations (represented by edges on the ontological graph) such as analogy/disanalogy, cause-effect, representation, identity, part-whole, uniqueness, similarity, and various properties. All these classes of relations represented by various grammar constrains in the English texts. Follow the ideas of “Universal grammar” and “Distributed Reduced Representation” proposed in papers of Paul Smolensk [8], we define any semantic space as a convolution of some vectors-concepts providing the space reduced representations. Rather than using Smolensky’ Tensor [7] that has variable length we decided to use Holographic reduced representation techniques [8]

Each node and edge were assigned to random vector from 512-dim space then any combination of connected nodes and edges were defined as a result of circular convolution on the vectors [8]:

$$z=x @ y, \text{ where } z_j = \sum x_k * y_{j-k}$$

Example: sub-graph “woman-wear-cloth” was represented as $z=x @ y = x @ (m @ n)$; where “m”-represents vector “woman”, “n”-“wear”, “cloth” and convolution of “m” and “n” gave us “woman-wear-” open-end sub-graph that has one node and one edge.

Due to high complexity of the ontological graph we perform “graph compression” that resulted in elimination of some redundant links. We decided to perform compressions just over the vital relations such as Time, Space, Identity, Role, Cause-Effect, Change, Intentionality, Representations and Attributes. Mathematically, that operation was implemented as circular correlation: $y=x\#z$, where $y_j = \sum x_k * z_{k+j}$. Taking the previous example, that operation is equivalent to “woman-wear-”#”woman-wear-cloth”=”cloth” and will return node “cloth”.

Holographic reduced representation let us quickly compute “generic space”.

The final ontological sub-graphs blending and generation of resulting imaged space were performed as confabulation on the *generic space*.

dK- Ontological Confabulation: We extended operation of Confabulation previously proposed in paper [7] and develop extended version of this operation for ontological graphs. This operation extracts concepts not mentioned in the text message. Consider our example: “Blue ball was on the beach. Woman walks on the beach. She takes the ball and kicks it”. It seems clear that imagination should build the picture of the woman that binds her body to take the ball by hands even nothing in the text did mentioned that bio-mechanical process. To do that, *Imaginaser* will use *ontological confabulation* for extracting knowledges associated with provided concepts. Confabulation was defined [9] as maximization of probability to start from nodes a, b and c and get node d: $p(abc|d) \sim p(a|d) * p(b|d) * p(c|d)$

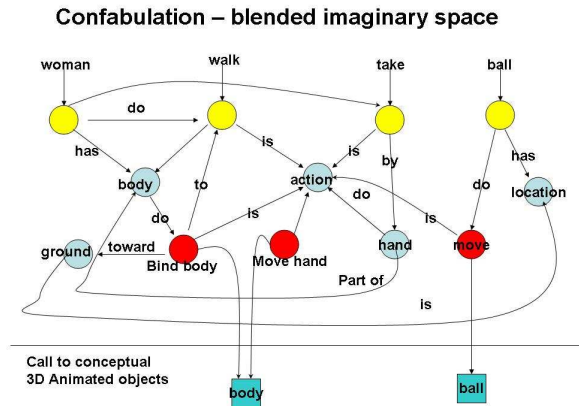
We start from any input node A then do random walk and calculate probability to get to some node B. That probability obviously depends on the order degree for node B. The more nodes connected to B through some path the higher probability to get there. Initial algorithm proposed in [9] use just weight of edges those calculated from pair wise frequencies of two nodes in some text.

Here we propose new algorithm to calculate probability of transition by using node degrees and joint probabilities of dK –series (0K, 1K, 2K, 3K) that was extracted from cohort of the text during building of the ontological graph. Due to analysis [6] that any properties of complex graph can be reconstructed by random graph with identical statistical properties for 0 to 3K sub-graphs, we suggest to calculate probability of a transition from A to B through some intermediate nodes as a sum over degree distributions for all intermediate 0K,1K,2K and 3K sub-graph between A and B.

If we start from several input nodes then we total probability to get to the node B is sum over all probabilities calculated for each input node. The B node with highest probability will be taken as a part of new blended space. The next less probable node was taken as a part of imagenized space if probability was higher then some

threshold. That threshold was estimated heuristically.

Fig5. Final imaginary space emerge as a result of ontological confabulation.



4. “SCRIPT WRITER” AND ANIMATION PROCESS

ScriptWriter is an attempt to simulate visual imagination through creation of 3-D animation of English text. It is done in a way how it probably would be performed by a human reading the story and imaging it from a first person perspective.

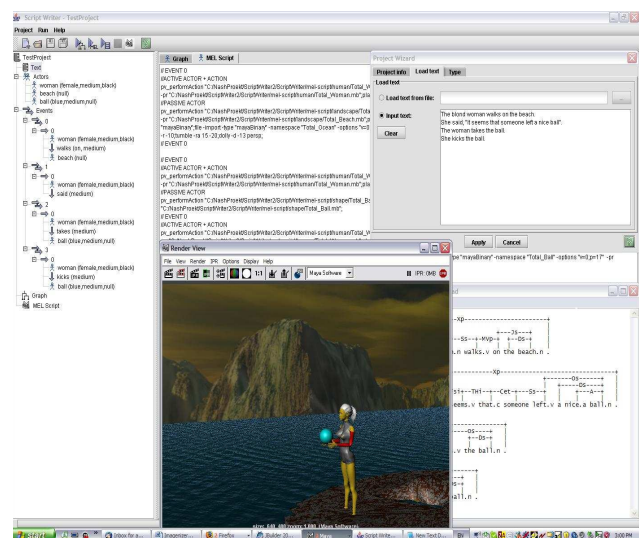
The first limited version of the software called *ScriptWriter* was build and provides us with ability to “imagine” short stories with primitive objects such as human- actors and several landscapes. *ScriptWriter* performs text processing, semantic extraction and animation planning that utilize some approaches tested in the various areas of Interactive Virtual Environment development [2,3,4,5]. Those were designed specifically for development of believable agents – characters which express rich personality, and which, in our case, play roles in an imaged animated world.

ScriptWriter provides a set of libraries and API those going to be converted in formal scripting language. It provides a connection to sensory-motor system of agents – “Actors” and support multi-agent coordination. *ScriptWriter* scenario was organized as collection of behavioral actions of the actors or simply “Actions”. During imagination process some imaged action encapsulates some other actions in the some kind of nested way that will produce sequential behavior. An example of sequential behavior is

shown below: “Woman walk on the beach. There was a blue ball on the beach. She kicks the ball.” We easily can imagine scene where woman walk to the ball left on the beach by someone and how she is kicking the ball. That text can be animated now by *ScriptWriter* with minimal human intervention.

5. THE DEMONSTRATION AND FUTHER DIRECTIONS

The demonstration will present to the user the “ScriptWriter” performing animation for short texts of 2-4 sentences. This will include demonstration of text processing and building semantic relations as well as generation of scenario for the animation. This scenario will be used to automatically generate script for building animation in the “Maya” animation environment.



This project raises a number of interesting AI research issues, including imagination process management for coordinating visual object interactivity, natural language understanding and autonomous agents (objects, landscapes and their interactions) in the context of story. These issues were partially answered in current version and will be refined in the next generations of the software.

References

- [1] Fauconnier, G., Turner Mark Book: The way we think. 2003

- [2] Bates, J. 1992. Virtual Reality, Art, and Entertainment. Presence: *The Journal of Teleoperators and Virtual Environments* 1(1): 133-138.
- [3] Bates, J., Loyall, A. B., and Reilly, W. S. 1992. Integrating Reactivity, Goals, and Emotion in a Broad Agent. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, Indiana, July 1992.
- [4] Mateas, M. and Stern, A. 2000. Towards Integrating Plot and Character for Interactive Drama. In *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*. AAAI Fall Symposium Series. Menlo Park, CA: AAAI Press.
- [5] Sengers, P. 1998b. Do the Thing Right: An Architecture for Action-Expression. In *Proceedings of the Second International Conference on Autonomous Agents*. pp. 24-31.
- [6] Priya Mahadevan, Dmitri Krioukov† Kevin Fall‡ Amin Vahdat: A Basis for Systematic Analysis of Network Topologies.
<http://www.krioukov.net/~dima/pub/index.html>
- [7] Alan Prince* and Paul Smolensky*
 Optimality: From Neural Networks to Universal Grammar. *Science* vol. 275 14 March 1997.
- [8] Tony A. Plate
 Holographic Reduced Representation. *CSLI Lecture Notes Number 150, 1997*.
- [9] Rumelhart, McClelland, and the PDP Research Group. *Parallel distributed processing. Book 1,2*.
- [7] Robert Hecht-Nielsen.: A Theory of Cerebral Cortex. *UCSD Institute for Neural Computation Technical Report #0404*