SUPRAMIND

Learning, Ordering and Probability Logic: A Unified Framework

Robert B. Lisek
Fundamental Research Lab
and
Department of Logic Wroclaw University

"If for any given group of objects the probability of a randomly-chosen object not having a given value is less than 1, then there must exist an object which has a given value." Paul Erdoes.

Categories and Subject Descriptors (according to ACM CCS): I.2.6 [Artificial Intelligence]: Learning - Concept Learning;I.3.6 and Universal Algebra, Probability Logic, Ordered Sets, Machine Learning, Natural Language Procesing,, Teleimerssion and Telerobotics

This is not a technical document, but rather an abstract, conceptual description to which more detailed information will be added in the future.

Abstract
The goal of the project SUPRAMIND is to build a general-purpose intelligent framework for the development of a class of intelligent systems. The concrete goals of project is create general-purpose software 'implants' i.e., programs that can be used to solve problems in many completely different fields:
• virtual agent control in simulation worlds (both pets and humanlike avatars and performers)
• natural language question answering (ex. in searching engines or in robotics)
• recognition of patterns in large bodies of economic and scientific data
• decision-making in multi-parameter processes (telerobotics and robotics),
• understanding and generating new stories
and others.
If you are interested in applications and selected implementations please start read this paper from the last chapter.

Introduction
I'm interested in creating software which will be able to handle a great many different problems quicket and more effectively than man. Here I'm presenting a conceptual framework for the development of a ‚class' of intelligent sytsems. I emphasise a ‚class', rather than a single system. A system is intelligent not because of what it can do, but because of what it can learn to do. An intelligent sytem adapts itself to its environment and works with incomplete knowledge and resources. More precisely, the system has a set time and is always ready for new tasks, processing them in real time and learning from its own experience.* The key idea is in my framework is to fully integrate the top-down (logical problem solving and reasoning) based approach with the bottom-up unsupervised, reinforcement (learning-based statistical pattern recognition).

A Kernel of framework
A kernel of framework should be unified, general purpose, and consistent. The aim is not to program specific logics and inferential techniques from the start, but rather to **„teach the system to learn'.** Specific structures and inferential techniques will surface during the learning process. A specific logic with its attendant inferential mechanisms will emerge as a result of the result of the learning process, and an appropriate logic will be chosen to deal with the appropriate problem. The method of learning must not be programed in advance, but emerge from the learning process itself. Kernel will decide which tools to use for specific problems, monitor the variuos processes, and check the results. Kernel should be unified, rather

than a hybrid of different technologies. Nevertheless, kernel will provide a platform for various heterogenic approches, which in turn will be added to it, becoming parts of the system itself.

To recap, Supramind will be a conceptual framework for the development of a class of intelligent systems. My aim is to conduct experimental research into these systems. I need an appropriate environment in which to carry out this research easily, quickly and thoroughly. Here I'm talking about a general environment, in which to construct and test various sytems. The Framework will enable the comparison of various different systems (FOL, modal logic, paraconsistent logic, many-valued logics, event calculus), as well as of neuron networks and probability networks. The Framework will enable us to choose an appropriate system for a given situation.

The basic characteristics of the Framework are:

0. Development Environment
1. Uncertain knowledge processing (randomness, fuzzines, ignorance and others types)
2. Adaptative, distributed network
3. General, consistent Kernel
4. Dynamism (morphisms processor)
5. Learning and meta-learning
6. **Evolution and creativity (genetic programing)**
7. **Experience Grounded semantic**

0. Development Environmment

0.1 build a development environment to rapid construction and comaparison of different systems with local rules and global behaviour
0.2 use empirical tests as the way to find the appropriate mechanisms
0.3 guide this process by visual insight and graph manipulation

The Development Environment would enable me to build systems [with different types of inference], compare them in detail and then observe them in action. My general intention is to build systems which have local rules, and then to observe their global behaviour. The environment would enable comparison of how changes in local systems will affect their global behaviour.

Example: In DE I could construct an inferential mechanism, and then look at different versions of that mechanism. Or I could construct, for example, a concept-learning mechanism, whose variable parameters would be able to dreate differnet versions of the mechanism. DE is an environment which could simulate these experiments quickly, without hacking on low-level code. Such an environment allows the user to focus his attention on any structural component and say „change it". The Framework will look at the definition of the components and will try to find a way of making variations. Next the Framework suggests certain structures to the user. The user will be to experiment manually, understand the structure and then logically select the best version.

The Interface: formulas and graph manipulation
The interface will consist of fields for text and logical formulas [parsed and interpreted], and graph manipulator. Under construction is the NL module enabling the use of a natural language, such as English, or a simplified representation with the help of Stories. I'm writing this module with programmers from Institute of Informatics at the Wroclaw University, software is based on Speagram, a powerful parser and functional programming language [Kais 05]. But the crucial part of the interface will be the tools for experimenting with various ordered sets, in the form of graphs in which the user can easily test specific techniques of problem-solving by changing the parameters. A collection of elements [my definition of the active processes which represent concepts], which interact with each other, could be structured in, for example, two groups. This will be represented graphically as two blob-like zones connect by a narrow bridge. The user can point to the blobs or the bridge and say "show me some variations". The user could then instruct the system to use, for example, the Monte Carlo Method, in which the system chooses specific

examples from the variation-space, and then extrapolates zones within the space which might interest us. All this should happen without entering into the source code of Supramind.

A frequent mistake is the simulation of an environment in which there is one virtual robot separated from the environment [Ha 03]. More effective is to put the emphasis on the relationship between objects and its representation as graph. Each user creates new structures by adding new nodes and links between them. The concept is to generate many different orders and to test the connections between them. The data could also have different characteristics and parameters, such as location, creation and distribution process, links, and critical elements which will form a new graph when suitably connected. The result is that different collections of data can be linked in various ways. Apart from an exact relationship, data can be connected in a parallel way which takes into account evolution, context and extrapolation [ilustration].

Dynamic, self-organizing environment
So it's not about building a static environment, but a dynamic, self-organizing environment. The user is an integral part of the environment, and his behaviour has an influence on it's look. A user operating in the environment leaves traces – length of visit, number of graphs, problems he is interested in , type of structure used, etc. This data is then analized, and parsed into VRML. As a consequence, a network of splines and nurbs is generated, representing the strength and tension between the data.
Chemical analogy. In a way this is parallel to building new chemical structures by the automatic analysis of the different chemical connections between elements. When a new chemical is formed, some elements fit each other, while others don't.
FLOAT & FACE. When it comes to working with graphs, I have a great deal of experience. Take a look at FLOAT, where I used Directed Acyclic Graphs. This project has been presented in many places, including the Lower Manhattan Cultural Council in NYC: http://fundamental.art.pl/float_00.htm. Another of my projects is FACE [Free Art Concept Exchange] – which I'm preparing for Neuspannung.org in Bremma and Cologne – a tool for the easy creation, transformation and exchange of artistic concepts.

1. Uncertainty processing (fuzziness, randomness, ignorance and others types)

In everyday life we often come across phenomena and concepts of an ambiguus and imprecise nature (e.g. fuzzy modifiers like "more", "very", "many", "some" "most", "a few", or "likely", "highly likely", etc). We are unable to describe formally such concepts by sing the classic set theory and first order logic (FOL). There is uncertainty in the tasks, concepts, rules of deduction, and in the deduction process itself. The various forms of uncertainty include randomness (in probabilistic theory), fuzziness (in fuzzy logic), and ignorance. Conflicts can arise when we use them simultaneously. We need a common base which takes into account the various kinds of uncertainty. In other words we want different methods of representation within one framework, which all have a common mechanizm allowing us to move between them according to need. A system which uses quantitive truth values (NTS - Numeric Truth Value), in which the rules are effective, i.e. programable. In FOL all statements can be inferred from a contradiction. In my plan this would not happen. We use the rules of addition, multiplication and Bayes rules and valuations in the style of G.F. Rota, placing great emphasis on inference and priority distribution.

Fuzzy logic networks. With the aid of fuzzy sets we can formally define such ambiguous and imprecise concepts as "high temperature", "young man", "average growth" or "big city". In the case of the concept "a lot of money", one amount will be considered "a lot" if we limit ourselves to the universe of discourse [0; 1000000 $], and a different one if we use division [0, 1 mln $.]
In many problems concerning steering of processes, the selection of a model for a particular process is essention. A knowledge of the process allows the selection of the correct driver, but on the other hand finding the correct model often proves difficult. The use of fuzzy sets as a driver does not require familiarity with these models – we only need to formulate the operating rules along the lines of IF … THEN …. This approach also allows us to resolve the problem of classification. Fuzzy drivers and classifiers are particular examples of fuzzy inference systems, consisting of the following elements: 1 : a set of rules [linguistic model] 2 : [fuzzyfication block], 3. [block of inference], 4. [sharpening block]. Such a system is shown in the diagram below.
I have spent a lot of time studying and testing various neuron networks, including Hopfield and Hamming networks, multilevel with backpropagation networks, BAM networks, self-organizing networks of the

WTA and WTM type, ART networks, and radial networks. I'm of the opinion that we should concentrate on two types - Fuzzy Neuron Network and Probabilistic Neural Network. – and use their special properties. Various forms of neuron-fuzzy networks have been presented in the last few years. These kinds of network are different from neuron networks in that they provide a clear representation of knowledge through the use of fuzzy rules. The crucial difference is the possibilty of introducing linguistic information - which is naturally included in logical inference. Besides, neuron-fuzzy networks can be taught using, for example, the backpropagation of errors method. Teaching is most often subject to the parameters of the membership function of the IF and THEN parts of fuzzy rules. There is also the posssibility of of using evolutionary algorithms to teach not only the parameters of the fitness function, but also the fuzzy rules themselves. The majority use the Mamdam deduction system or the Takagi-Sugeno method, which depend on the connection of predecessors and successors within the rules with the aid of t-norms [most often of the min and product type]. In the example of the Takagi-Sugeno method, the rules' successors do not have a fuzzy quality, but are input variable. Nevertheless, during the course of our work we tested a logical form of induction which depends on the connection of predecessors and successors with the aid of fuzzy implication. In logical induction the aggregate of individual rules operates with the aid of t-norms. A neuron-fuzzy system should have good generalization properties. This type of network can be used to tackle difficult problems using nonlinear processes, such as economic processes, decision-making in multi-parameter processes, and many others.
It should be emphasised that the theory of fuzzy sets describes uncertainty in a way different from the calculation of probability. The only similarity between the theory of fuzzy sets and the calculation of probability is the fact that the membership function of a fuzzy set, as well as that of probability, have a value in the set [0,1]. From a strictly formal point of view, probability is a additive normed measure on the Boolos algebra.
Initially, people accept that uncertain reasoning could arise through the connection of FOL (first order logic) with TP (probabilistic theory), but in such an approach either one of them could function erroneously. But even starting with a FOL we can sometimes produce an effective probability model. I think that here it's worth examining the Markov Logic Network., a special structure which is a probablilistic FOPL solution. The connection of FOL with probability according to the Markov Netwrok is a kind a step forward, but it doesn't solve all the problems, and actually introduces new ones. Nevertheless, this approach could be useful, and forms part of my SUPRAMIND framework. I use a Markov network, an undirected graphical model where the nodes represen random variables and the edges joint probability constraints relating those variables. And next I use local belief updating techniques, in particular, 'loopy' belief propagation. A Markov Logic Network is a first-order knowledge base with a weight attached to each formula, and can be viewed as a template for constructing Markov networks. From the point of view of probability, MLNs provide a compact language to specify very large Markov networks, and the ability to fexibly and modularly incorporate a wide range of domain knowledge into them. From the point of view of first-order logic, MLNs add the ability to soundly handle uncertainty, tolerate imperfect and contradictory knowledge, and reduce brittleness. Many important tasks in statistical relational learning, like collective classification, link prediction, link-based clustering, social network modeling, and object identification, are naturally formulated as instances of MLN learning and inference. Experiments with a real-world database and knowledge base illustrate the benefits of using MLNs over purely logical and purely probabilistic approaches.
I research an extension of the Markow Logic Network is the replacement of FOL by various logic sets, selected for the problem we want to solve.

2. Adaptative, parallel and distributed network

The arrangement of knowledge in the system is not local, the system is stable and won't easily crash as a result of the introduction of incorrect data at the input. Because in a given situation some information is more important, a certain order appears, which can be presented in the form of a graph. More precisely: we can regard each concept as a node or as a set of linked nodes. For example, the concept "fish" could be contained either in a single node or in a dynamic configuration of nodes which are important in a given context.
Various combinations of elements, together their order, correspond to different types of inference
The answer to a particular problem is the co-operation of various structures. Thus, the generation of solutions is the emergent result of many non-local occurrences. Each of the occurrences is dependent on the

occurrences that preceded it. The way in which knowledge is generated is dependent on what previous knowledge is available to the system and how it is ordered.

3. KERNEL – mathematical principles

The framework needs a decision-making mechanizm which an inferential method uses for a problem. What is needed is an intelligent kernel which is general and flexible, a kernel which can make use of various inferential methods to solve specific problems. A kernel which is based on theories consistent with research into AI, neuroscience, psychology and linguistics. A kore which will use methods which are sufficiently general to be capable of embracing a great many different inferential methods.

The Kernel of the framework will be based on universal algebra, category theory and general logic (in the sense of the study of common structures of logics). General logic is an attempt to study and develop some fundamental concepts and tools to navigate within the jungle of all existing logics, build new ones and to have a better understanding of what logic is. General logic is not a new logical system. Any system is always limited, catches only one aspect of reasoning. The expression *general logic* was coined by analogy with the expression *universal algebra*. Universal algebra is not a super algebra, it is a general study of algebraic systems[Barr 05].
Methods:

1. different forms of composing and decomposing logics and algebras, such as fibring, fusion, splicing, splitting, synchronization and temporalization.

2. different forms for combining logics with other techniques as probability distribution, machine learning,etc.

The most important significance has theory of ordered sets and fixed points of morphism between ordered set. Retracts and fixed points have a crucial significance for recursion and computation. Details please see my Phd thesis where I research techniques as retraction and logics with fixed points operators – Fixed Points Logics [Lis 07].
Starting with the achievements of R. Cox [Cox 61], I showed the way in which we can extend an algebra to a calculus by assigning numerical values to pairs of elements of ordered set to describe the degree to which one element infer another. This kind of approach can be extended to any kind of problem in which the hierarchy of elements is important. Transformation of the product rule to Bayes rule will make probabilistic logical reasoning possible. Bayesian probabilistic theory is an inferential calculus, which go from generalization of inclusion in Boolos lattices to inclusion with parameters. In my calculus I use addition, multiplication and Bayes rule as one of the basic induction tools, [valuation and analogue law to Inclusion Exclusion Law (which operates in various fields of mathematics such as geometry, combinatorics, theory of numbers. The logic of questioning can be presented in this calculus as a distributive lattice (downset of all answers to a given question). My calculus could be used in cybernetic control, e.g. robotics.
The automation of inference and inquiry will allow machines to learn from data and ask the correct questions in order to receive more data. This approach promises the automization of teaching methods within the framework defined in terms of sets of possible experiments and sets of hypothetical models. [drawing of different typs of structures: chains, trees, dags, posets, multigraphs, hypergraphs].

4. Dynamics and flexibility. Morphisms Processor

The morphisms are transformations of ordered sets. Morphisms are important for processing stuctures. The topological structure of the network (weight of links and priority distribution) is changed through the use of morphism. We define a new concept: **the energy of a morphism**. The energy of the morphism of an order set is a scale-invariant of morphism 9function from morphism to rational numbers) The energy of set O is the infinitum of energy of the set of all the morphisms of the order set of type O. The type and canonic form of an order set is based on irreducible elements. **Intuitively, the connection between the complexity of the morphism of an order set and its energy is simple: the more complicated morphism, the higher the energy.**

Kernel provide the ground for switching between different computing techniques. Searching for the solution of a problem involves jumping between classes and between individual techniques. When one method fails, the system automatically jumps to a different method and continues to attempt to resolve the problem, much like our own way of thinking (switching between different orbits of thinking). Jumping between various classes of ordered sets and inferential mechanisms. Kernel decides which inferential scheme to use and when. Each order structure and inference scheme must be able „to recognise“ and make use of situations created by other inferential schemes – they nust communicate and agree on which parts of the scheme are suited to each other and which are not. When one fails, another takes over the task.

5. Genetic Aglorithms and Genetic Programing for Order Sets

A part of the framework is the use of genetic algorithms [GA] linked to network techniques. We can use evolutionary methods for solving problems, teaching weight of networks, and seeking the best architecture of the network. The two most convincing arguments for the use of evolutionary algorithms are: firstly, global searching of space of weights in the network, and secondly, the avoidance of local minima. The design of the optimum architecture of a network can be treated as the search for a structure which work best for any given task. That means the search for a structure and the choice of the best elements of the space, using defined criteria of optimality.

My framework will use ideas similar to genetic programing. A multi-dimensional space, in which each dimension corresponds to a specific aspect of the system. The population of systems will be obtained, and each generation the system's fitness value will evolve. Each generation of systems is produced with a higher fitness value, and systems with a higher fitness value will have a greater chance of becoming the parents of new systems. After some time natural selection will produce systems with better fitness, i.e. higher intelligence. GP and BOP (Bayesian Optymalization Procedures) are an important tool, because they allow us to narrow the field of search when looking for solutions to a given problem – for example, a program or robot faced with a choice of 5,000,000 possible solutions can choose the group of solutions which are most promising for the solution of the problem presented to it. Look at the drawing below [1.population of candidat solutions -> 2. selected candidats -> bayes network for selected candidats -> new candidate solutions]
I have experimented with variou types of Genetic Algirithms (GA) and GP and used them also in the field of art. I wrote (in Java language) the software GENGINE – a module that use genetic programming as an automated "invention machine". Please see documentation of our presentation in the National Gallery of Arts Zacheta in Warsaw http://www.flickr.com/photos/fundamental_data/.
If you are interested in the link between GE and hacking and computer viruses as artistic expression, please refer to my text [Lis 06].

6. Learning (machine learning and meta-learning)

The problem of learning: given information that system has gathered previously, plus certain a priori information – how can the system best use this information to make a decision?
The problem of probability: Given a certain body of data plus a certain priority information – how does the system best make predictions about the immediate future?
The aim of the project is not to the prior programing of an individual logic and one inferential technique in a system for solving a specific problem, but rather **"to teach how to learn".** Individual inferential techniques will emerge during the process of teaching the system
Several methods of machine learning are known. Artificial Neural Nets (ANN), Boolean Belief Nets (BBN), Support Vector Machines (SVM), Radial Basis Functions (RBF) and Prediction by Partial Matching (PPM) [Sol 86]. While they work quite well for the types of problems for which they have been designed, they do not use recursion at all and this severely limits their power. Among techniques employing recursion, Recurrent Neural Nets (RNN), Context Free Grammar Discovery, Genetic Algorithms (GA) and Genetic Programming (GP) have been prominent. Each of these methods has a different generality, different speed and working costs [Hutt 06]
Self-reference and meta-learning. The framework must be constructed in such a way that we will represent also height order representation, (e.g. statements about statements, orders of orders). The framework must possess the possibility of representing its own structure and processes for reflective

purposes (self-reference) in the system. Thanks to the rational use of self-referencing we can, for example, select one from a set of different learning algorithms, which means that we can construct meta-learning algorithms which can learn more effectively: *meta-searching* - searching for faster search procedures, *meta-learning* – learning to learn (learn better learning algoritms), *self-improvement* – self-improvement by self-modification of code. Matalearning means learning the credit assigment method itself through a self-modification of the code [Schm 03].

7. Experience Grounded Semantic

Classical model-theoretic semantic: depends on the use of static, complete models. Informally speaking, the model is a description of a domain and the relationship between objects. Interpretation transforms the terms from the system into objects in the model, and transforms predicates in the system into relationships between objects. We say that a given task has a truth value when what it says corresponds to the facts in the model, but this type of semantic is inadequate for my system. I need a semantic based on experience.
As a self-adpting system with limited knowledge and resources, Supramind is incapable of the accuracy of knowledge based on a complete static model. Everything must be based on experience. Experience differs from the classic model because it changes with time, is never finite and is often inconsistent. In such a system nothing is absolutely true or absolutely false, but is true to a degree (NTV – numerical truth value, measurement of uncertainty), based on previous experience. As an open system operating in real time Supramind, takes on new tasks based on experience. A new task might be any kind of information from the environment, from another system or network, of from a person, etc.

Knowledge Bases. The system can be connected to existing knowledge bases such as Cyc, WordNet, Open Mind(common sense) or Mizar (mathemathical knowledge). It could also have additional modules which use techniques such as semantic web, information retrieval and data mining, to gather knowledge directly from the Internet. Connection could be made via Internet, mobile phone (equiped with bluetooth technology), etc*.
Natural Language Procesing. After learning a natural language, Supramind should be able accept knowledge from various sources in that language. Essential is an interactive interface which will allow the human trainer to monitor the growing knowledge base by answering questions, so that the system would formulate the correct structure of cells and priority distribution of concepts, tasks and beliefs. Under construction is the NL module enabling the use of a natural language, such as English, or a simplified representation with the help of Stories. I'm writing this module with programmers from Institute of Informatics on the Wroclaw University, The software is based on Speagram, a powerful parser and functional programming language.
Bots and self-learning Robots
The automatization of inference allows machines to learn from data and give appropriate questions in order to receive new data. My approach enables the automatization of teaching methods defined as a set of posssible experiments and a set of hypothetical models. This type of system could be integrated with, for example, bots and robots, especially with robots in difficult remote environments. In other words an autonomous robot, operating without help or human intervention. Such a robot doesn't know in advance what it will meet nor what questions it should ask in connection with experiment which it is conducting. But by means of our inference engines it can calculate the appropriate questions, collect and analize data from the environment, and then pose new questions.
The Honda humanoid or DARPA Challenge car robot perform impressive tasks. But these robots are incapable of learning and a have a very limited implementation field. Unfortunately, traditional Reinforment Learning Algorithms (RLA) are limited to simple reactionary behaviour and do not work well with real robots. A robot learning in a realistic environment needs new algorithms in order to identify important events in a stream of initial sensory data, and to remember them as dynamic internal states until the time comes to calculate the correct control actions.
The use of ocupancy grids (OC) are a promising solution to working with robots. Ocupancy grids are a probabilistic method, and have been around for along time. But what is new is that the robot and the mapping process itself are considered as uncertain. The interesting thing about OCs is that this method is not more ad hoc than, for example, the LBS (landmark base system), but its most attractive characteristic is the possibility of linear scaling, which is crucial when the robot has to deal with large sets of data. In implementing the system it should be rmembered that managing large amounts of data could take place in

real time. Other sytems which could be realistically used are RNN (Reccurent Neural Network) and OOPS (Optimal Ordered Problem Solver)[Sch 7].

*Technical note about Data Base: I suggest the use of GDB (graph data base) rather than the usual relational DB, because it guarantees the possibility of representing complex problems and faster operation [a faster path with an answer to a question]. The weight of the path can be defined as the product of the weights of all the links along the path. Hitherto I have used MySQL in my projects.

Selected implementations

In the course of sketching the design of the framework, several possible implementations emerged relevant to the creation of intelligent prediction systems and decision-making systems.
This type of software could have various implementations in many fields, such as games and VR, finance, aviation and also in art and every-day life.
• virtual agent control in simulation worlds (both pets and humanlike avatars and performers)
• natural language question answering (ex. in searching engines or in robotics)
• recognition of patterns in large bodies of economic and scientific data
• decision-making in multi-parameter processes (telerobotics and robotics),
• understanding and generating new stories,
• algebraic theorem-proving assistant and many others.

Bots, Games buisness, Second Life
I do believe that as the Web will unfold over the next 5-10 years, it is going to gradually morph into a sort of "metaspace." Second Life, World of Warcraft, multiverse.net and all the other virtual worlds coming out lately, are just the beginning. Based on Supramind we can become the provider of intelligent agents for the metaverse as it emerges, that will be a pretty exciting business position. The short-term and concrete part: There is also a healthy market right now in terms of training simulations for use within government and industry. Non-player characters in training sims and "serious games" are just as inflexible and unsatisfying as in entertainment games. Business model going forward will be based on addressing this market, the MMOG market, and the nascent consumer virtual worlds market, in parallel.

Below I describe the framework from the point of a possible implementation for arts: a literature, performing and visual arts.

Story Database. A story can be a simple way of representing knowledge. With the help of my framework the user could, for example, create a new story graph which represents particular elements of a related story and the links between those elements. The framework could be used to construct a Story Database(SDB), which will be a collection of different stories, including traditional tales and fairiy stories.
Story Generator. The possibility of collecting stores in a precise way as a graph-ordered set and constructing a story database is not enough in itself. A dynamic method is needed to infer new stories – Story Generator. A Story Generator based on inference engines which our framework will provide us. An interactive story generator agent, which uses simple information and common sense knowledge provided by the user to generate short fictional texts.
Virtual Guide. A simple implementation is the creation of a virtual guide to stories. SD and SG Could become part of a virtual bot guide to very old and forgotten stories.
Universal Map of Stories. Another implementation is a map of existing stories and possible stores, a large semantic graph of human stories. This map could operate in real time. New stories that are constantly emerging in the Internet are represented as nodes in story graph, and new stories could be formed with the help of SG. At first a normal html or Flash interface could be used by users to introduce new stories. More interesting is the use of tools to mine the Web in search of new stories. Web-mining through the monitoring of the behaviour of certain groups in the web, or directly through access to the places which enable users to introduce stories could provide much material. Web-mining could also be helpful in the construction of large common sense systems [Min 87].
Programable Identity. Looking at the identity wich we create for ourselves by using the Internet is interesting. Here we mean identity as the structure which emerges from a net of concepts, a net which emerges from the monitoring of a given internet user: the places he visits and the groups in which he takes

part. It's impossible to move or operate on any level without leaving traces and fragments of seemingly unimportant personal information. These fragment can be collected, retrieved, multiplied and made whole.
Rituals in the Web. It would be interesting to study the behaviour of individuals in such Internet groups from the point of view of new emerging rituals and their links to ancient stories. The framework could also be used for self-reflexive performance in the Web, and for "dancing" with the computer audience of culture, and as a social networks tools. Social networks are graphs where nodes represent social actors (e.g., people) and arcs represent relations between them (e.g., friendship).
Performing art, bots and robots rituals and back to roots.* I don't intend to speak in detail about the connection between stories and the performing arts, but rather I think we should concentrate on the basic elements. What I mean by basic elements is the organization of movement, rhythm, composition of movement, contact and word. Please see also [Grot 75] and [Bar 91]. A significant role is also played by the consciousness of the body by a bot or robot, from the simplest possibilty of observing a body, to the self-observation of higher cognitive functions. Nevertheless, the question of operation is foremost, i.e. the robot must do something, and operate based on a structure. This structure is fundamental and essential. A robot is incapable of performing new operation outside the root scenario of that which it does, outside a structure which contains a beginning, development and end. There is a need for a logical structure – any given operation cannot come before or after. The structure of operation is like a spine – without it everything collapses and turns to pulp. To sum up, I predict that a certain group of elementary low-level behaviours will arise, where the robot will recognise and accept its situation with the help of sensors and be able to react. Nevertheless, at a higher level I can see layers of adaptive-learning behaviours which are capable of modifying low-level behaviours for the purpose of ensuring the appropriate reaction to changes in the environment. It would be worth testing the ideas above in practice. I'd like to emphasise that are based on practical experience which I gained by taking part in workshops organized by leader actors of the best directors of the 20th Century, such as Gurdjijew, Grotowski and Brook.
* According to Grotowski, the concept of The Performer, capital letters, is a man of action. He is not a man who plays another. The Performer - archetypal images of human suffering derived from antique myths (including the death and resurrection of Christ) as well as historical events, specifically the Holocaust. The Performer is a warrior whose body and intelligence are in an osmosis in which it seems impossible to separate them. It's a two-way action: rigouristic action and rigouristic searching.
TeleimmersiveMetaChoreographer. MetaChoreorapher will be software for manually programing movments of virtual performers or pets. MCH will infere and propose new forms of movments based on some atomic movments. Directors or choreographers can work with classes of movments and not with particular one. Choreographer defines the initial position of virtual performers and the final which they should achieve. MCH finds and propose the most interesting passages between these two positions. Choreographer can choose interesting solution and if necessary corect it by manually manipulation of motion graphs of this sequence. Next interesting characteristic of MCH are inteligent interactions between virtual performers. Humans (and many other animals), display a remarkably flexible and rich array of social competencies, demonstrating the ability to interpret, predict and react appropriately to the behavior of others, and to engage others in a variety of complex social interactions. We believe that developing systems that have these same sorts of social abilities is a critical step in directing virtual performers, and other computer agents or robots, who appear intelligent and capable in their behavior, and who are intuitive and engaging for humans to interact with.
The processes and representations used to generate the behavior of expressive virtual performers are a valuable and largely untapped resource for helping those performers make sense of the world around them by using his own motor and action systems as models for the behavioral
capabilities of others. Ex.: let's start with two performers: A and B. A can begin to identify simple goals and motivations for B's behavior, Additionally, A uses a novel motion graph-based movement recognition process in order to accurately parse and imitate B's movements and behaviors in real-time and without prior examples, even when provided with limited synthetic visual input. The motion graph can be used to facilitate both movement parsing and movement recognition.
Teleimmesion i Telerobotics,. Next step can be connection MCH with a VR and real immersion of user in the worlds of non-euclidean spaces and virtual personalities. There is a paradox: one hundred years after Minkowski and Riemann geometries and Einstein's relativity special and general theories the most popular way to represent space and movements is 3D Euclidean geometry. I suggest the use of other forms of representation of space: multi-dimensional and non-euclidean geometries (ex. 4-dimensional, hyperbolic spaces that I used in my project SSSpear).
I think also about implementation that to allow anyone, regardless of technical experience, to use robotics

to create and express their ideas. With software that allows out-of-the-box internet connectivity and support for wireless networking, someone unfamiliar with the complexities of computer networking is able to connect their robot to the internet. Advances in artificial general intelligence and espacially Supramind could be applied to the WWW, transforming it to a globally distributed, massively parallel, wetware-oriented universe[Liss 05].

*Technical note about the programing language: The language in which my framework will be written could be at the beginning Phyton, Haskell, Ruby, next C, Lisp, Prolog.. We prefer C. In principle LISP could be also OK for our framework, but I don't think it's right for content and quick work.

ACKNOWLEDGMENTS

I would like to thank Luk and Szczep, for their programming cooperation.

REFERENCES

[Bar 91] Barba E., Savarese N., Secret art of the performer, Routledge (London-New York,) 1991
[Barr 05] Barr M and Wells Ch , Toposes, Triples and Theories, Reprints in Theory and Applications of Categories, No. 1, 2005, pp. 1289.
[Cox 61] Cox R. T., *The Algebra of Probable Inference*. Baltimore: The Johns Hopkins Press, 1961.
[Dor 99] Dorer K., 'Behavior networks for continuous domains using situation– dependent motivations', Proceedings of the Sixteenth International Conference of Artificial Intelligence, 1233–1238, (1999).
[Grot 75] Grotowski J., Towards a poor theatre, Methuen and and Co. Ltd, 1975
[Ha 03] Hapke M., Komosinski M., Application of evolutionarily optimized fuzzy controllers for virtual robots. In Proceedings of the 7th Joint Conference on Information Sciences, pages 1605-1608, 2003. AIM
[Hutt 05] Hutter M A Universal Measure of Intelligence for Artificial Agents Proc. 21st International Joint Conf. on Artificial Intelligence (IJCAI-2005) 1509-1510
[Hutt 06] Hutter M., Asymptotic Learnability of Reinforcement Problems with Arbitrary Dependence, Proc. 17th International Conf. on Algorithmic Learning Theory (ALT-2006) 334-347
[Jayn 68] Jaynes E. T., "Prior probabilities," IEEE Trans. on Sys. Sci. and Cyber, vol. 4, pp. 227-241, 1968.
[Kow 86] Kowalski R. and Sergot M, A logic-based calculus of events, New Generation Computing Volume 4 , Issue 1, 1986
[Kais 05] Kaiser L., Game Quantification on Automatic Structures and Hierarchical Model Checking Games, Proceedings of the 8th International Conference on Foundations of Software Science Springer-Verlag, vol. 3441, 2005
[Len 95] Lenat D B: 'CYC: A large-scale investment in knowledge infrastructure', Communications of the ACM, 38, No 11, pp 33, 1995
[Lis 05] Lisek, R.B., "WWAI. WWAI: How is the Web Growing? Into a Social Super-Organism or a Mass of Disconnected Information?, Siggraph, ACM 2005
[Lis 06] Lisek, R.B., "Viruses, Hacking and tactical media", catalogue National Gallery, Warsaw 2006
[Lis 07] Lisek, R.B., The fixed point property in the theory of partially ordered sets. PhD thesis before defense, Wroclaw University, 2007
[Maes 90] Maes P., 'Situated agents can have goals', Journal for Robotics and Autonomous Systems, 6(1), 49–70, 1990
[Min 87] Minsky M: 'The society of mind', Simon & Schuster, 1987
[O'Re 01] O'Regan J. K. and A. No¨e. A sensorimotor account of vision and visual consciousness. Behavioral and Brain Sciences, 24(5), 2001.
[Phi 03] Philipona D., K. O'Regan, and J.-P. Nadal. Is there something out there ? Inferring space from sensorimotor dependencies. Neural Computation, 15(9), 2003.
[Sch 02] Shmulevich Ilya, Probabilistic Boolean Networks, Bioinformatics, Vol. 18 no. 2 2002
[Schm 03] Schmidhuber J., Goedel machines: self-referential universal problem solvers making provably optimal self-improvements. Technical Report IDSIA-19-03, 2003
[Schm 04] Schmidhuber J (2004) Optimal ordered problem solver. Machine Learning, 54(3):211–254, 2004
[Sol 86] Solomonof R ., An application of algorithmic probability to problems in artificial intelligence. In Kanal L, Lemmer J (eds) Uncertainty in Artificial Intelligence, Elsevier Science Publishers/North Holland, Amsterdam., 1986

SHORT NOTE ABOUT AUTHOR

Robert B. LISEK is an artist and mathematician who works at the truly cutting-edge vanguard of electronic arts and media, a founder of FUNDAMENTAL RESEARCH LAB; he is involved in the number of projects focused on radical art strategies, hacktivism and artificial intelligence. Drawing upon conceptual and software art, telepresence, robotics, meta-media, criticism of new media art, and introducing numerous additional influences from mathematic, his work intentionally defies categorization. A pioneer of art based on AI. His visionary combination of AI and telepresence explores the possibility of building distributed intelligent entities. He is also a scientist connected with the Department of Logic of Wroclaw University specializing in the theory of partially-ordered sets. He exhibits, lectures, and conducts workshops worldwide. His projects include among others: FLOAT – Lower Manhattan Cultural Council,NY; WWAI-Siggraph, Los Angeles; Falsecodes - Red Gate Gallery & Planetary Collegium, Beijing; GENGINE-National Gallery, Warsaw; FLEXTEXT- CiberArt Bilbao, Medi@terra - Byzantine Museum, Athens, STACK- RunMe-Moscow, Ars Electronica- Linz; FLEXTEXT – ACA Media Festival, Tokyo; STACK – ISEA 02, Nagoya; SSSPEAR –17th Meridian, WRO Center for Media Arts, Wroclaw, HAPPY NEW FEAR – FluxusOnline, New Horizonte; ODER- Graz, Dusseldorf, Athens, Paris, Tokyo, Palermo…
(data from the sites of Polish Cultural Institute in NY)

http://fundamentalresearch.org/