Info-Activity Method in Problem Solving

Alexander S. Geyda
St.-Petersburg Federal Research Center
of the Russian Academy of Sciences
St. Petersburg, Russia
geida@iias.spb.su

***Abstract*—The purpose of the research is to propose a new method of human problems solving that better correlates with ways humans solve problems historically. I notice that most human problems appears and solved during human action and in the form of information processing. It is suggested to research and reproduce humans behaviour of performing activity, especially one related to activity problem formulation, problem solving and using results for changes in activity. Article named this method info-activity method of problem solving. Its essence consists in the explicit representation of human problems to be solved during the activity as the informational artifacts, processes of determining the answers to relevant questions about activity in changing conditions as informational actions and obtained due to problem solving information application in activity as outcome of problem solving. Problem answers definition can take various forms, such as of a search, synthesis, prediction, and justification of answers to the problem question. As a result of the approach application, it should be possible to automatically solve pragmatic problems (i.e. problems that arise during activity) that are usually solved by humans. Directions for further research suggested. It is shown that the use of the proposed method can significantly increase the number of problems solved.**

I. Introduction

Many formal techniques for problem-solving rely on human abilities, intelligence, languages, ability to deduce and on human behavior – drawing inspiration from them and trying to copy. Among the most known ones is machine learning (ML). ML traditionally defined as part of Artificial Intelligence [1] concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks (without being explicitly programmed) [2], [3]. But one of the most outstanding manifestations of human history – human activity – as well as information problem-solving during such activity – was, to my knowledge, not paid with enough attention. The reason can be found in the complexity of activity and its placement between human and his environment, between nature and conscience, in all their intrinsic relations and complexities. It cannot be stated that human activity was never studied, but certainly it was not yet studied enough to create formal techniques for solving human activity problems automatically, with computers. Such problems can be solved by the use of models, methods, and algorithms to create programs that behave similarly to people who solve complex problems at all stages of their activity. Human activity problem represented as a question about activity, which shall be answered. As such, human activity problem is an informational artifact and shall be answered using some kind of information processing.

I am suggesting to name such method in human activity problem-solving as info-activity method (IAM). IAM should allow organizations and systems to organize activity on the never seeing before level of effectiveness and efficiency, to build organizations and systems with unparalleled potential, to bring sustainable development to human society. IAM consists in considering activity in its everlasting relations with flow of information and problems solving related to this activity. Flow of information is considered as the cause of a necessity of changes in activity, the necessity of change is the cause of problems to be solved by humans and the results of problems' decisions are the cause to conduct the change due to information obtained, synthesized, deduced. As a result, activity is in constant change and realization of cause – and effect relationships related to this change through information use. But there are many obstacles of various kinds on the way to active application of IAA and AA. First. They are of philosophical nature. We as humans do not yet understand well enough the relations between nature, humans, activity, knowledge, information, organization, and conscience. These are not yet enough researched to be formalized for problem-solving. Further, we have not yet created enough formal means adequate to solve the majority of activity problems. Of course, some of such means were created and in use. For example, formal logic means semiotics and algorithmic theories. But they are not yet mature enough to describe activity and information use for problem-solving. Many known problems are solved by formal means, which are based on some kind of problem mapping into space with measure of needed quality. In many cases, such measures can be used to create smooth, integral-differential models of problems and to solve them with existing means of mathematical analysis. Unfortunately, such measures are not yet developed enough to provide needed mappings for problem solving. Finally, we have not yet established a clear understanding of relations between problem statements, their models, problems decisions, activity and information application. As a result, the gap exists between required means to solve problems humans meet when performing activity, especially automatically, for example, with suggested AA use, and available means. This article is devoted to the goal to close this gap by suggestion of info-activity approach to human activity problem solving. Concept, architecture of IAM and some candidate formalisms to use with IAM considered.

II. Fundamentals and Concept of IAM

IAM is based, first of all, on philosophy of action and information. Philosophy of information (PHI), considered in this article as part of IAM. PHI is a term coined by Floridi [4]–[8] and developed in the works of many theorists [9]–[12] and practitioners [13]. These authors define the philosophy of information as a philosophical field that (a) critically examines the conceptual nature and basic principles of information manifestation, including its dynamics, use, and scientific research, and (b) develops and applies information-theoretical and computational methodologies to philosophical problems of information manifestation. It is argued that the philosophy of information should become the integration basis of natural science disciplines, a kind of philosophy of natural science, in particular, using the transdisciplinary method of information research [14]. In [12] it is highlighted:

- Philosophy of information (with consideration of ontology, epistemology, ethics and aesthetics of information) [15].
- Information methodology.
- The philosophy of the study of information (with consideration of the ontology, epistemology, ethics and aesthetics of information research).
- Methodology of information research.

Quantitative and qualitative methods of information research are distinguished. The quantitative approach goes back to the works of Shannon [16] and then [17]. The qualitative approach has emerged relatively recently, and its emergence is most often associated with the works of Burgin (Burgin, 2003; Burgin, 2010a, 2010b). Burgin defined information as something that: "transforms one communication of an association of information into another communication of the same association." They consider information pragmatically, dynamically, in connection with changes in the recipient's structure or behavior. A similar approach is used in the present study. In the works of the famous Chinese researcher of the philosophy of information, Wu (Wu Brenner, 2017), a set of disciplines of information Science is defined, which looks as follows: General Theory of Information (1); Philosophy of Information (2); unified Theory of information (a Unified Theory of Information, 3); information Theory (Information Science, 4); different kinds of (5) and an 4); various types of computer science (Informatics, 5) and the paradigm of information knowledge (Information Knowledge Paradigm, 6). A.D. Ursul [18] was, most likely, the first to investigate these issues. In the USSR and then in the Russian Federation, the work was successfully continued. Thus, it is worth mentioning the works of K.K. Kolin [19], [20] and his students, as well as R.M. Yusupov and his colleagues [21]. It should be noted that the very definition of information has not yet been sufficiently established. As a rule, it is associated with the concepts of data and knowledge (having in some sense between them). For example, in [22], a study of various ways of determining information was carried out. Subjective [23] and objective [24] concepts of information definition. In Russian sources, these two approaches have acquired the form of functional and attributive approaches to the definition of information. Analyzing their connections, the authors [25] rightly noted that it makes sense not to contrast these two methods, but to establish connections and relationships between them. As a result, they proposed a "systematic" definition of information based on the concepts of Checkland [23], Stonier [24], Reynolds [26] and [22] interaction of subject and object. These authors propose a pragmatic teleological view of information. It is based not on the subject and its reasoning, nor on the object and the empirical data obtained from it, but on the subject's action on the object). In this case, it is not the object or the subject that is important, but their relations in their complex (in the system). In our opinion, the most important difference between information is its focus on the changes implemented by the entity in its activities and on the corresponding process of change. Such changes, firstly, require an information relationship between at least two elements (in fact, this corresponds to a functional approach to defining information). And secondly, they should lead to changes in activities. So, J. Nauta [27] believes: "There is a relationship between information, meaning, and purposefulness… [If] we abstract from the meaning and purpose … What remains is simply the transmission of physical states and records of events with a certain degree of uncertainty. It is possible to assign a certain amount of potential information to this only if a person is ready at the same time to give it an appropriate degree of potential significance and purposefulness." Newman [28] states: "A person cannot receive information unless he is attuned to alternative possibilities, and this attunement, i.e., having an adequate mental representation of alternatives that supports message recognition, constitutes knowledge." Note that alternative possibilities may be changed during the implementation of the activity. As well, Nauta [27] states: "Pragmatics deals with the functional aspects of all possible information processes. This is the most complex and comprehensive level of semiotics: syntax and semantics can be included in it. For this reason, pragmatics has not yet been fully explored… it studies the use, outcome, and usefulness of signals, signs, and symbols in semiosis." The pragmatic aspect of information, the change in activity resulting from the use of information, has not been fully disclosed before. It is proposed to develop a systematic definition of information in the concepts of Churchman, Stoner, Reynolds and Callaos [22] in the direction of including activity in relations related to the manifestation of information pragmatics. At the same time, the most important aspect becomes what information causes, how it is used by a person to alter his activities. As a result, the pragmatic aspect of information is considered in the systemic unity of the object, the subject, and the activity implemented by it, i.e., in the pragmatic aspect of the use of information (hereinafter also referred to as information pragmatics). As already mentioned, the pragmatics of information is manifested in solving relevant tasks, such as finding answers to questions (arising from the implementation of activities). First, [9] we are interested in subject-transforming activities aimed at nature, society, and man. The tasks that arise in the implementation of these types of activities, as a rule, are studied within the framework of various types of natural science disciplines. At the same time, it should be agreed that, as stated in [15]: “whenever science works at the forefront of the known, it invariably faces philosophical problems concerning the nature of knowledge and reality. Scientific debates raise questions such as the relationship between theory and experiment and the extent to which science can approach the truth.” In our opinion, speaking about the pragmatics of natural science disciplines, i.e., about the aspects of their use in the implementation of activities, questions inevitably arise about the use of data, information, and knowledge in activities. This view is based on the writings of several philosophers. Thus, in [12], it is pointed out the need to develop natural science theories within the framework of the general theory of information, in fact, as a philosophy of using information and the philosophy of information in its pragmatic aspect. For the general theory of information, the basic axiological principles of information (axiology is the science of values) are proposed [9]. This aspect becomes even more relevant if pragmatic tasks are investigated, i.e., tasks that arise in the activity. Such tasks, being questions to which an answer should be found, are essentially informative. Moreover, speaking about the philosophy of using information in activities, it is necessary to describe the philosophy of activity in its relationship with the philosophy of information. Thus, a scientifically based study of the tasks of pragmatic analytics using modern scientific results should be based on the philosophy of using information in activities. This philosophy combines the philosophy of information and the philosophy of activity into a complex based on consideration of the philosophical aspects of the tasks of pragmatic analytics. The tasks of pragmatic analytics serve as the basis for integration. This consideration largely overlaps with the positions of information philosophers. Thus, Floridi in his work “Philosophy and computing” [4], [29] points out that “information appears when data answers an explicit or implicit question posed by the recipient of the data. To become informative for an intelligent being… the data should be functionally related to the relevant question.” Thus, according to Floridi, information can only manifest itself in relation to the recipient of the data, answering his question, i.e., the information arises in connection with the task. Other authors link questions and scientific knowledge: “It is precisely this that marks out a problem as being of the true scientific spirit: all knowledge is in response to a question. If there were no question, there would be no scientific knowledge [29]. We consider the tasks and problems solved by people in the implementation of subject-transforming activities, one way or another, related to the search for answers to questions arising from human interaction with nature. The relationship between the question of the task and philosophy is revealed in [30]: “philosophy still learns the most important lesson that the right question is a vital part of philosophical discussion)”. Many authors considered such concepts as human activity, causation, information, and problem-solving within some unifying frameworks. For example, [31] considered relations of activity, language, mind, humans behaviors, abilities [32] considered system of concepts, namely action, knowledge, memory, perception, and reference. Information dynamics, change, and computing considered by [33]. Casualty and causal methods, including ones in information studies and problem-solving were considered in [34]. Unfortunately, a complex unifying framework of activity, information, problem-solving, information and automation was not yet built. At the same time, problem-solving (finding the answer to a question) is closely related to another important concept — the Level of Abstraction (LoA, the level of abstraction) by different subjects of solving the problem. LoA is defined by [6]: ”LoA is a finite nonempty set of observations. The order is not correlated with observations, which are intended to be used as building blocks in a theory described using block definitions. An LoA is called discrete (or, respectively, analog) if (and only in this case) if all observations are discrete (respectively, analog), otherwise it is called hybrid”. Related concepts are considered in [35]: abstraction gradient, abstraction grid, levels of organization, levels of interpretation. It is proved that LoA has two definitions: functional and structural. Collectively, many levels (LoAs) form a system of representatives called abstraction gradients, since some LoAs are more abstract than others. This, in turn, makes it possible to vary the viewing and switch from one potential customer in the system to another. It seems that LoAs of various types and related concepts should become the basis for building models of problems and their subsequent solutions. Various LoAs should be reflected in the structure of a set of action and task networks, forming a data structure for solving activity tasks using information. This structure can become the basis for building conceptual, formal models, machine learning models and the subsequent solution of information use research tasks. Logic, reasoning, events calculus may play a prominent role in unifying activity, information, problem-solving, and automation [36]. Another concept used in solving problems may be logically close to the concept of activity state infon concept [37] and similar ones, for example, in the theory of situations [38], [39]. I have previously introduced the concept of the information state (substate) of systems in the implementation of activities [40]. This state is informational, higher LoA, part of the states describing the system and differs in that the future states of the system depend on it. It is stated that appropriate (situational, infon) logic, LoAs/Infon/State data structures similar to alternative stochastic networks [41] notation and formal languages in LoAs/Infon/State spaces should be developed to describe and solve pragmatic problems. In the best case, measures with needed properties shall be defined on such spaces. The example of such measure can be system potential measures [42]. Such spaces should make it possible to generate descriptions of higher levels of abstraction, including spaces that allow the use of modern mathematical means of solving problems (for example, in metric spaces). In addition, this should then make it possible to build functional dependencies of various types between the characteristics of information, information technologies, states of systems and nature, measures of compliance of states with requirements, measures of feasibility, measures of proximity of various implementations of functioning to requirements, ideal, dangerous implementations, each other and (or) other measures that make it possible to draw reasonable quantitative conclusions about the quality of functioning using this or that information. This will further make it possible to solve many pragmatic problems using available mathematical models and methods. It is advisable to classify problems arising in the philosophy of information and the philosophy of activity into classes of unsolved problems of various kinds of activity, in connection with which the relevant problems are actualized, and then solved based on the application of various models and methods of various applied sciences, according to different types of abstraction levels used. It is advised to refer to the complex of problems as the applied informatics problems (i.e., the use of information and information technologies in activity problems) and tasks of applied informatics, respectively. "In activity" can be omitted since there is an adjective "applied" referring to application of some activities. Similarly, the tasks solved by applied computer science should be presented. It is clear from the above that all tasks considered in the framework of applied computer science should be represented by one or another pragmatic task. In connection with the above, it becomes possible to define the concept of pragmatic tasks in more detail. A pragmatic task is an unresolved issue caused, on the one hand, by the desire to achieve the goal of the activity, for which to carry out actions (or inaction), and the lack of necessary information about these actions and (or) inaction. The necessary information may include information about how, with what, and at what cost, and with what characteristics these actions should be implemented. Pragmatic problems arise first at conceptual level, after problem manifestation and conceptualization. A problem statement is a result of conceptualization. It consists of descriptions of problem parts, their relationships, the question of the problem and the selection criterion. Unresolved problem statements can cause a scientific problem if the necessary information is not only missing, but also there are systematic difficulties in obtaining it. The requirement to find a solution to a pragmatic problem necessitates the use of one or another method of finding a solution and possibly models of the problem. The pragmatic problem model is understood as the interconnected set of models of objects of activity (in question), the relationship between them in the implementation of activities, possible solutions for the implementation of activities and the question (issue) of the problem. The problem model can be formulated at different levels of abstraction. For example, it can include only a text description, or it can also include parameters and variables (dependent, independent) of the problem formulation. In the latter case, the solution to the problem may have the functional form of calculating dependent variables from independent ones. When talking about problem models and methods for solving them, it should be noted that these models (methods) can be conceptual, formal, or machine learning models. Formal models and methods can use different theoretical means. For example, logic, set theory, graph theory, functional analysis, measure theory. Conceptual models describe problems with natural languages use. Formal models describe problems based on the use of formal languages. Machine learning models describe problems based on the use of "big" data for model generation. Machine learning models can allow one to obtain solutions to relevant problems by using these models directly. Models and methods of artificial activity can be used to obtain solutions to individual pragmatic problems, as well as to systematically solve complexes of pragmatic problems of various types. These types of models and methods include models and methods that make it possible to systematically automate complex activities by setting and solving complex tasks at different levels of abstraction. For example, artificial activity (AA) can be used to set tasks that arise during research planning, subsequent description of experimental tasks, tasks of processing experimental results, tasks of searching for models and methods, tasks of determining methods for solving specific applied problems. The philosophy of activity and the philosophy of information use should help us to formulate specific applied research problems of the information application. At the same time, such problems are formulated and solved based on the means of specific applied sciences, but using concepts and methods studied within the framework of the philosophy of information use and the philosophy of activity.

III. Info-Activity Method and Artificial Activity

Systematic prediction, research, modeling of application of information, activity and its objects changes and resulting cause and effect relations constitute info-activity approach suggested. Method extend known approaches, such as Activity Centric computing [43], activity-centric systems [44], [45] in the direction of:

- Modeling direction:
  - integrated activity and information application modeling based on complex info-activity states ("infact");
  - explicit cause effect relationships modeling between complex states;
  - combining process models with cause-effect and probably other models for explicit processes, their changes and information application modeling using system of models;
  - using info-activity models to predict activity results in changing conditions, including changing environment and changing information conditions.
- Data application direction:
  - combining questionaries, ontologies, texts, tabular data for processes and their changes representation and modeling in various conditions;
  - activity recognition, change recognition, information recognition techniques application;
  - application of fundamental models for processes, changes and information application modeling;

  - reinforcement learning;
  - active adaptive application of experimental data, including virtual experiments and synthetic data.
- Models application direction:
  - models application to represent human problems related to activity performed;
  - solving problems with application of formal techniques and existing solvers;
  - splitting problems;
  - decision of problems on higher level of abstraction.

According to this new approach elaborated, information application, activity problem-solving and activity realization shall be modelled as a systemic whole and in its interrelations with time and space. One of the results of IAA could be AA realized. To realize AA it is suggested to elaborate: Interrelated concepts of AA, AA models, learning AA models, using AA results. Languages for describing an activity, logic for describing it, specifications of data structures and algorithms for implementing different types of AA's. AA models, including models that allow machine learning based on available data. Methods for solving AA problems. Technologies for solving AA problems. AA shall make it possible to take automation (of activities) to a new level, allowing automation not only of individual parts and types of activities, as is common now, but also a complex of organizations, systems of hierarchical networks of actions of various types, including but not limited to research and experimental activities. Automation is usually understood as a technology that can be used to perform a process with minimal human involvement. Intelligent automation (IA; Cognitive Automation, CA) refers to the use of automation technologies such as artificial intelligence (AI), business process management (BPM) and automation of operations management services (robotic process automation, RPA) in order to optimize and scale decision-making processes in organizations [46]. The main components of IA, intelligent automation technologies, can be found in [47], [48]. It is argued that it is their integration that provides the required IA results:

- The first component is artificial intelligence (AI). Using machine learning (ML) and complex algorithms for analyzing structured and unstructured data, enterprises can develop a knowledge base and make predictions based on it. This part is a software decision—making mechanism for IA.
- The second component of IA is business process management (BPM), also known as workflow automation. This component allows for greater flexibility and consistency in business processes. Business process management is used in most industries to optimize processes and improve collaboration in their implementation.
- The third component of IA is automation of operations management services (robotic process automation, RPA). RPA uses software services like robots to implement office operations such as data extraction and form filling. Such bots complement AI, because due to RPA, you can use AI to solve more complex tasks. It is pointed out the importance of building IA compliance with the changed, dynamic business goals of enterprises, especially strategic ones, to use the potential of IA. Main differences between existing IA and the proposed AA concept:
- At the level of the first IA component. There is a much greater penetration of both AI technologies and a set of other tools that complement AI in solving problems and in actions that are automated as a result. A set of activity problems to be solved, rather than activity results forecast, becomes the goal of applying a set of intelligent methods, including AI. A set of different types of actions is automated, including a system of intellectual actions which depends on each other by cause-effect relationships, such as the search, preparation and use of data, information and knowledge. It automates and intellectualizes the solution of a complex of interrelated problems and the implementation of appropriate decisions (answers), including intellectual, higher LoA actions performed in connection with the search for answers to specified problems, the search for necessary information, the automation of problem statements, modeling, and the automation of the justification of problem-solving methods. When solving a problem, it is assumed to use sets of higher LoA actions related to formulation, modeling, search for necessary information and problem solving. These actions and the automation of their solutions are not limited to the use of machine learning.
- At the level of business processes, a set of interrelated processes is considered, which are alternated when appropriate information obtained and used (for example, concerned to operating conditions changes). These processes include modernization, innovation, business process alternation, readjustments, maintenance and upgrades. The implementation of such complexes of processes occurs depending on the operating conditions and the results of the implementation of other processes and solving a set of relevant problems. Such problems may require dependent sets of higher LoA actions to solve problems, i.e. to search for the necessary data, information, and knowledge. As a result, a complex (network) of dependent actions and problems of various LoA formed.
- At the RPA level, the complex of tasks being solved (including automated ones), depending on changing information and actions implemented as a result, leads to the need to implement a comprehensive hierarchical network of automation software services.

To implement AA, it is necessary to develop the necessary conceptual and methodological tools for automating AA tasks. Among them, philosophical, conceptual, logical, and linguistic foundations for describing problems, actions, models, and methods used in solving problems, and information technologies for automating problem-solving and performing appropriate actions are expected to be developed. The philosophy of information, the general theory of information, can become the philosophical basis of AA. The theory of information use and the theory of potential can become a conceptual basis.

IV. Architecture for Info-Activity Method Application

This architecture integrates activity and information states into unified (codename ”infact”) states, analogously to ”infons” [49], [50] with explicit causal relationships, enabling predictive modeling under changing conditions and changing information. The design addresses the three specified directions (Modeling, Data Collection, Model Application) through a multi-layered, adaptive system.

A. Overall Architecture

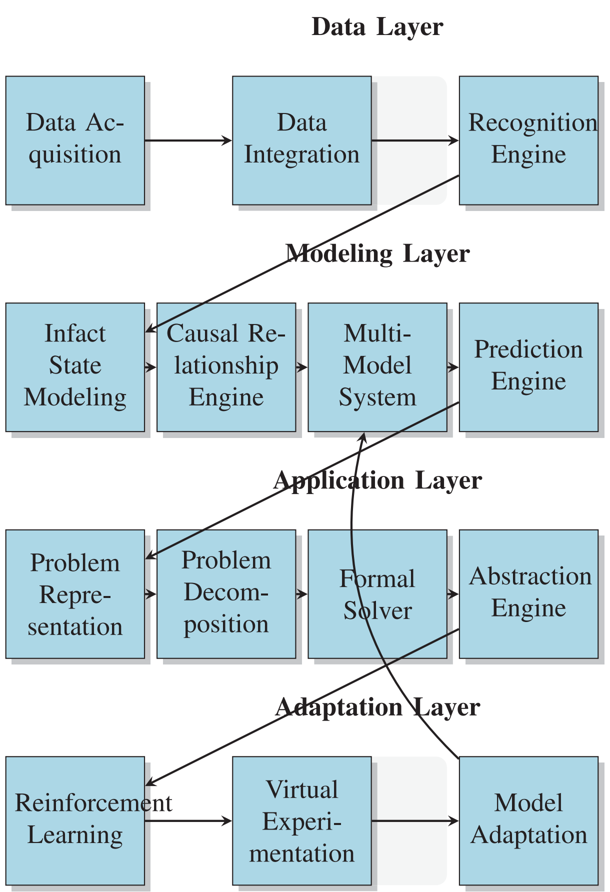

Fig. 1. Overall Architecture

The architecture consists of four primary layers working together:

- **Data Layer**: Acquires and integrates multi-modal data sources
- **Modeling Layer**: Represents infact states and causal relationships
- **Application Layer**: Solves human problems using formal techniques
- **Adaptation Layer**: Continuously learns and improves models

B. Data Application Direction

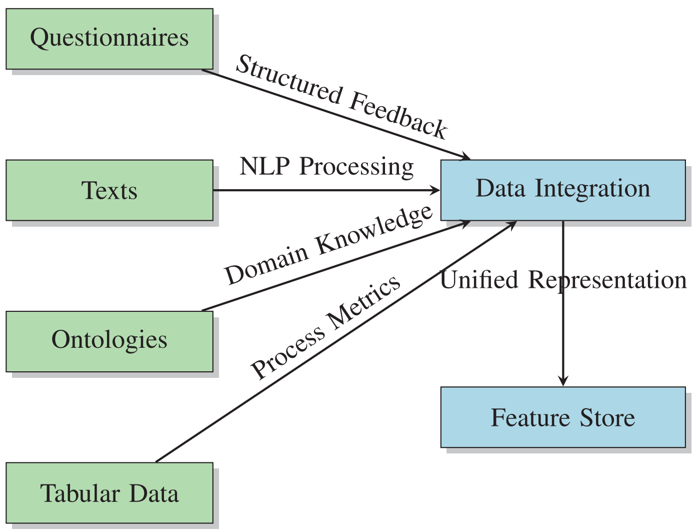

Fig. 2. Data Acquisition & Integration

1) Data Acquisition & Integration: The system integrates heterogeneous data sources:

- **Questionnaires**: Structured feedback on activities
- **Texts**: Unstructured descriptions using NLP processing
- **Ontologies**: Domain knowledge representation
- **Tabular Data**: Quantitative process metrics

```
class DataIntegrator:
    def __init__(self):
        self.questionnaire_processor =
            SurveyProcessor()
        self.text_processor = NLPExtractor()
        self.ontology_mapper = OntologyMapper
            ()
        self.tabular_processor = DataEngineer
            ()

    def integrate(self, sources):
        integrated_data = {}
        for source in sources:
            if source.type == 'questionnaire':
                integrated_data.update(
                    self.
                        questionnaire_processor
                        .process(source)
                )
            elif source.type == 'text':
                integrated_data.update(
                    self.text_processor.
                        extract(source)
                )
        return self.
            create_unified_representation(
            integrated_data
        )
```

Listing 1. Data Integration Class

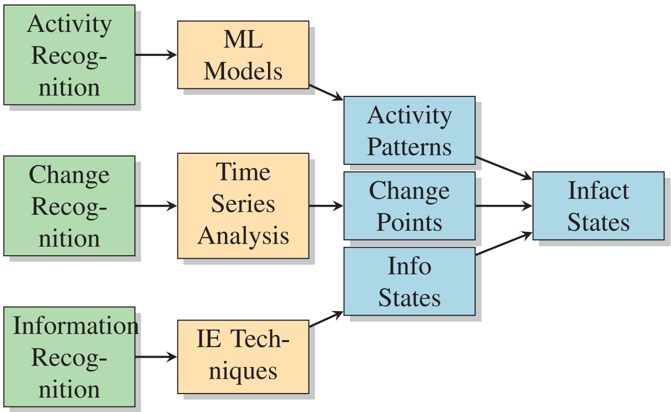

Fig. 3. Recognition Engine

2) Recognition Engine: The Recognition Engine identifies:

- **Activity Patterns**: Using transformer-based models
- **Change Points**: Through time series analysis
- **Information States**: Via information extraction techniques

```
def detect_changes(activity_stream):
    # Multivariate change point detection
    change_points = ruptures.detection.Pelt(
        model="rbf"
    ).fit_predict(activity_stream)

    # Causal impact analysis
    causal_changes = causalimpact.analysis(
        pre_period, post_period,
            activity_stream
    )

    return change_points, causal_changes
```

Listing 2. Change Detection

C. Modeling Direction

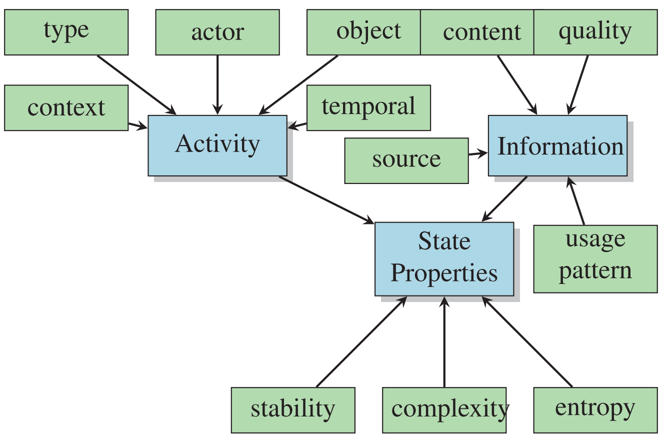

Fig. 4. Infact State Modeling

1) Infact State Modeling: Infact states integrate activity and information:

```
InfactState = {
    activity: {
        type: string,
        actor: entity,
        object: entity,
        context: environment,
        temporal_properties: time_interval
    },
    information: {
        content: knowledge_graph,
        quality: quality_metrics,
        usage_pattern: interaction_sequence,
        source: provenance_chain
    },
    state_properties: {
        stability: float,           # 0-1
            scale
        complexity: int,           #
            dimensionality
        entropy: float,            #
            uncertainty measure
        coupling_strength: float     # activity
            -info coupling
    }
}
```

Listing 3. Infact State Representation

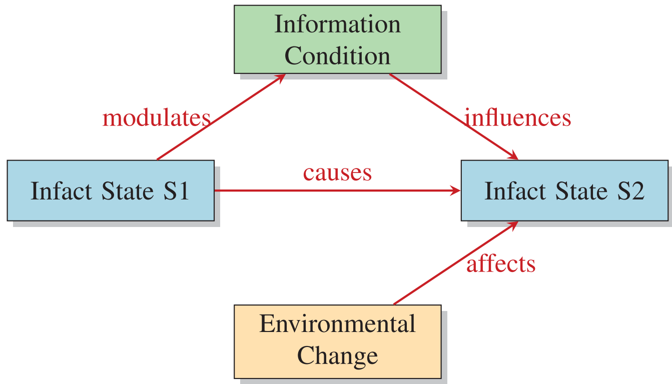

Fig. 5. Causal Relationship Engine

2) Causal Relationship Engine: Explicit causal relationships are modeled using:

```
class CausalDiscovery:
    def discover_relationships(self,
        infact_states):
        # Structural causal models
        causal_graph = causalnex.structure.
            notears.from_pandas(
            infact_states, tabu_edges=[],
                max_iter=100
        )

        # Causal Bayesian networks
        causal_model = BayesianNetwork()
        causal_model.fit(infact_states)

        # DoWhy for causal identification
        causal_model = CausalModel(
            data=infact_states,
```

```
        treatment='information_quality',
        outcome='activity_success',
        graph=causal_graph
    )

    return causal_model
```

Listing 4. Causal Discovery

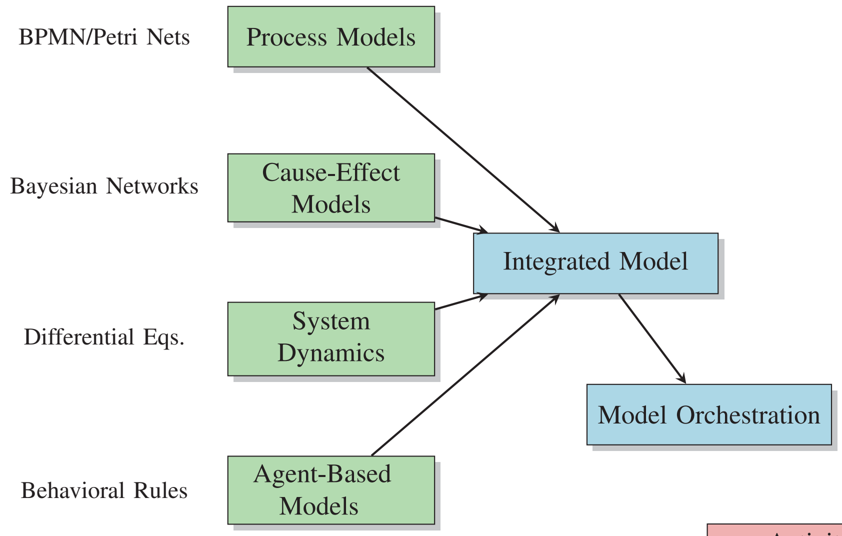

Fig. 6. Multi-Model System

3) Multi-Model System: The system integrates multiple model types:

```
class ModelOrchestrator:
    def integrate_models(self, infact_state):
        # Process model execution
        process_outcome = self.process_model.
            simulate(
             infact_state
        )

        # Causal analysis
        causal_effects = self.causal_model.
            predict(
             infact_state, interventions
        )

        # System dynamics
        system_evolution = self.system_model.
            evolve(
             infact_state, time_horizon
        )

        # Agent-based simulation
        agent_outcomes = self.agent_model.
            simulate(
             infact_state, agent_behaviors
        )

        # Consensus building
        integrated_prediction = self.
            build_consensus(
             process_outcome, causal_effects,
             system_evolution, agent_outcomes
```

```
        )

        return integrated_prediction
```

Listing 5. Model Orchestration

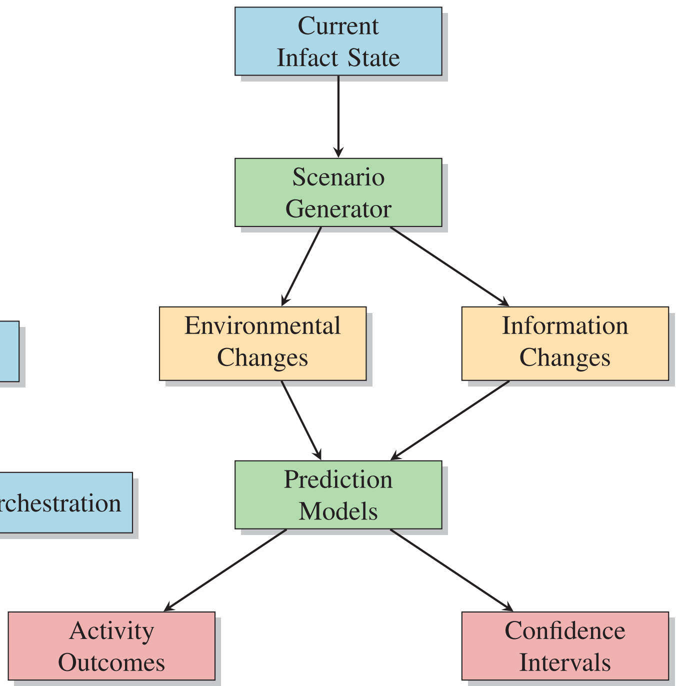

Fig. 7. Prediction Engine

4) Prediction Engine: Predictive capabilities include:

```
def predict_outcomes(current_state,
    change_scenarios):
    predictions = {}
    for scenario in change_scenarios:
        # Transform state according to
            scenario
        transformed_state = apply_changes(
            current_state, scenario
        )

        # Multi-model prediction
        process_pred = process_model.predict(
            transformed_state)
        causal_pred = causal_model.predict(
            transformed_state)
        system_pred = system_model.predict(
            transformed_state)

        # Ensemble prediction
        ensemble_pred = ensemble_method([
            process_pred, causal_pred,
                system_pred
        ])

        # Uncertainty quantification
        confidence = calculate_confidence(
            ensemble_pred)

        predictions[scenario] = {
            'outcome': ensemble_pred,
```

```
            'confidence': confidence,
            'key_drivers': identify_drivers(
                transformed_state)
        }

    return predictions
```

Listing 6. Multi-Scenario Prediction

D. Models Application Direction

E. Problem Representation

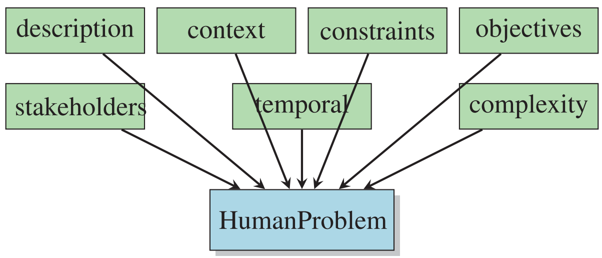

Fig. 8. Problem Representation

Human problems are formally represented:

```
HumanProblem = {
    description: natural_language,
    context: infact_state,
    constraints: [constraint_1, ...,
        constraint_n],
    objectives: [objective_1, ..., objective_m
        ],
    stakeholders: [stakeholder_1, ...,
        stakeholder_k],
    temporal_properties: time_horizon,
    complexity_metrics: {
        dimensionality: int,
        uncertainty: float,
        interdependencies: graph
    }
}
```

Listing 7. Problem Formalization

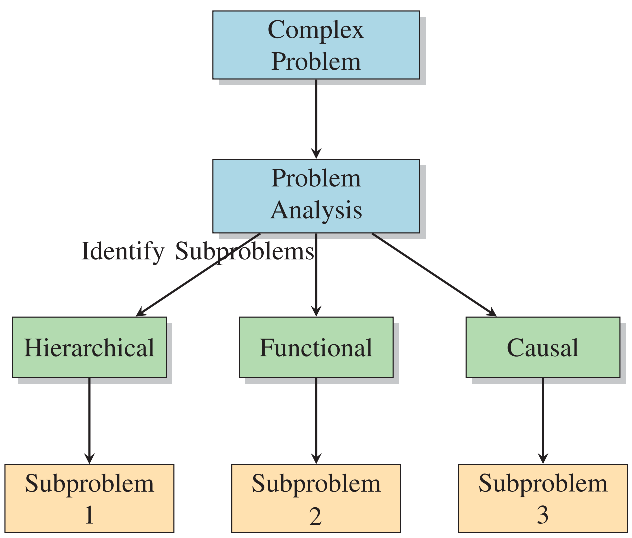

Fig. 9. Problem Decomposition

1) Problem Decomposition: Complex problems are decomposed using:

```
def decompose_problem(problem):
    # Identify hierarchical structure
    hierarchy = htanalyzer.analyze(problem)

    # Functional decomposition
    functional_parts = functional_decomposer.
        decompose(problem)

    # Causal decomposition
    causal_parts = causal_decomposer.decompose
        (problem)

    # Generate subproblems
    subproblems = []
    for level in hierarchy.levels:
        for component in level.components:
            subproblem = Subproblem(
                parent=problem,
                component=component,
                functional_parts=
                    functional_parts[component
                    ],
                causal_parts=causal_parts[
                    component]
            )
            subproblems.append(subproblem)

    return subproblems
```

Listing 8. Problem Decomposition

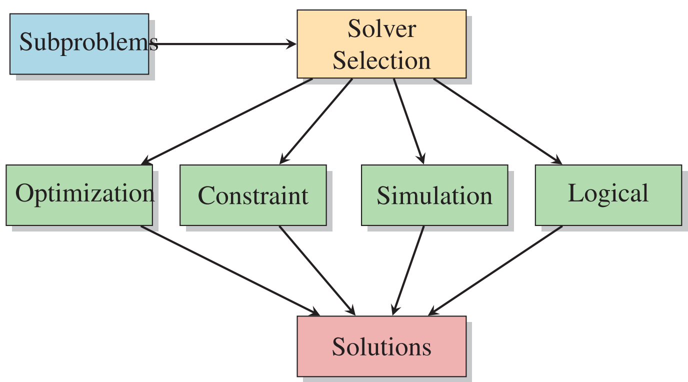

Fig. 10. Formal Solver Integration

2) Formal Solver Integration: Multiple solvers are orchestrated:

```
class SolverOrchestrator:
    def solve_subproblem(self, subproblem):
        # Select appropriate solver
        solver_type = self.select_solver(
            subproblem)
        solver = self.solvers[solver_type]

        # Transform subproblem to solver
            format
        solver_problem = self.
            transform_for_solver(
```

```
            subproblem, solver_type
        )

        # Execute solver
        solution = solver.solve(solver_problem
            )

        # Transform solution back
        solution = self.transform_from_solver(
            solution, solver_type
        )

        return solution
```

Listing 9. Solver Orchestration

F. Abstraction Engine

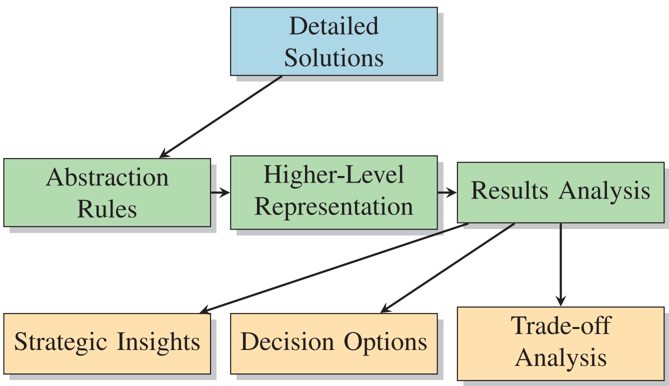

Fig. 11. Abstraction Engine

Solutions are abstracted to higher levels:

```
def create_abstraction(detailed_solutions):
    # Identify patterns
    patterns = pattern_recognizer.identify(
        detailed_solutions
    )

    # Group similar solutions
    solution_groups = clusterer.group(
        detailed_solutions)

    # Extract key principles
    principles = principle_extractor.extract(
        solution_groups)

    # Generate strategic options
    strategic_options = option_generator.
        generate(principles)

    # Analyze trade-offs
    tradeoffs = tradeoff_analyzer.analyze(
        strategic_options)

    return {
        'patterns': patterns,
        'principles': principles,
        'strategic_options': strategic_options
            ,
        'tradeoffs': tradeoffs
    }
```

Listing 10. Abstraction Method

V. Adaptation Layer

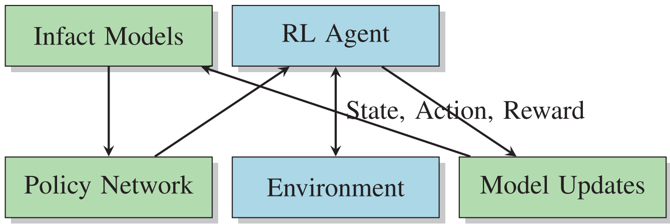

Fig. 12. Reinforcement Learning Integration

1) Reinforcement Learning Integration: RL optimizes information application:

```
class InfoActivityRL:
    def train(self, environment):
        # Collect experiences
        experiences = self.collect_experiences
            (environment)

        # Update model network
        self.model_network.update(experiences)

        # Plan with model
        imagined_trajectories = self.
            imagine_trajectories(
            experiences, self.model_network
        )

        # Update policy and value networks
        self.policy_network.update(
            imagined_trajectories)
        self.value_network.update(
            imagined_trajectories)
```

Listing 11. RL Integration

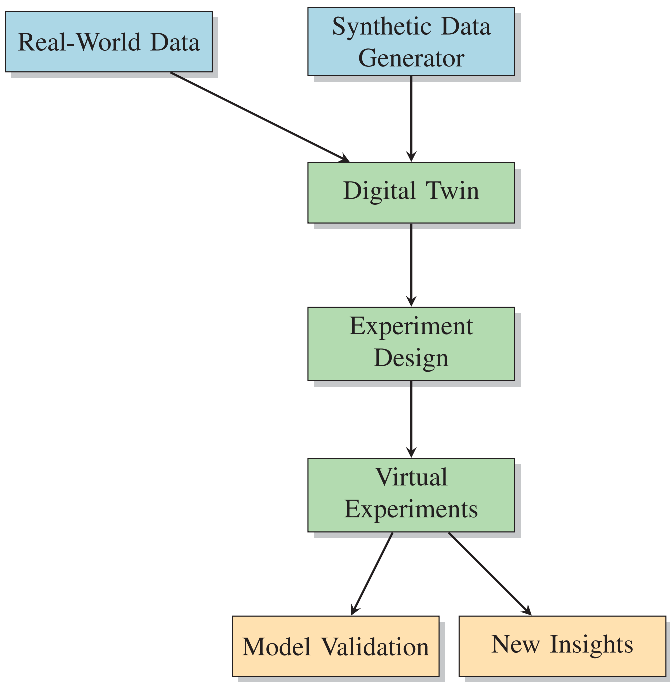

Fig. 13. Virtual Experimentation

2) Virtual Experimentation: Digital twins enable virtual experiments:

```
class DigitalTwin:
    def run_experiments(self, twin,
        experiment_designs):
        results = []
        for design in experiment_designs:
            # Run virtual experiment
            virtual_result = twin.simulate(
                design)

            # Analyze results
            analysis = self.analyze_results(
                virtual_result)

            results.append({
                'design': design,
                'result': virtual_result,
                'analysis': analysis
            })

        return results
```

Listing 12. Virtual Experimentation

A. Model Adaptation

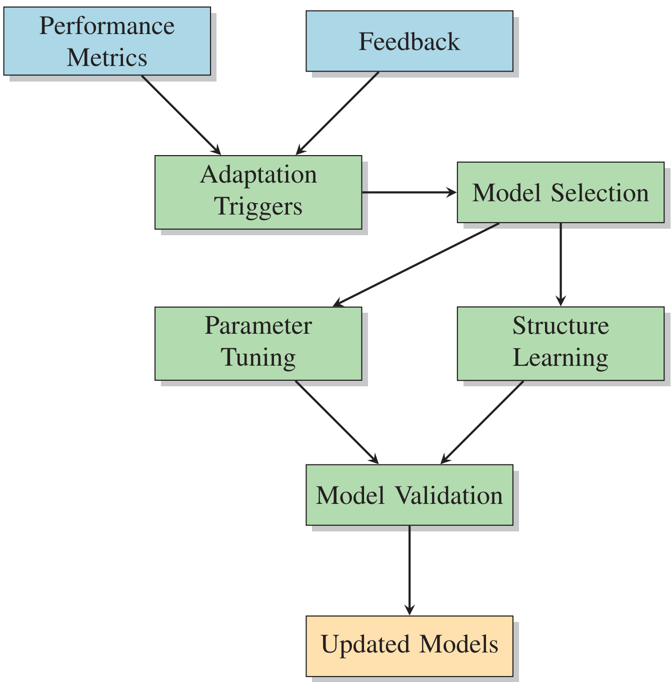

Fig. 14. Model Adaptation

Models continuously adapt:

```
class ModelAdapter:
    def adapt_models(self, new_data, feedback)
        :
        # Check adaptation triggers
        triggers = self.check_triggers(
            new_data, feedback)

        if triggers.adaptation_needed:
            # Model selection
            best_model = self.select_model(
                new_data)

            # Parameter tuning
            tuned_model = self.tune_parameters
                (
                best_model, new_data
            )

            # Structure learning
            adapted_model = self.
                learn_structure(
                tuned_model, new_data
            )

            # Validation
            if self.validate_model(
                adapted_model):
                self.update_model_pool(
                    adapted_model)

        return self.model_pool
```

Listing 13. Model Adaptation

VI. Implementation Stack

TABLE I. Technology Stack

Layer	Technologies
Data Application	Apache Kafka, spaCy, Neo4j, Pandas, Scikit-learn
Modeling	PyTorch, TensorFlow, DoWhy, CausalNex, AnyLogic
Model Application	OR-Tools, Gurobi, MiniZinc, D3.js
Adaptation	Ray RLlib, Stable Baselines, Docker, Kubernetes

A. Key Innovations

1) **Infact State Formalism**:
   - Unified representation of activity-information complexes
   - Quantifiable state properties (stability, complexity, entropy)
2) **Explicit Causal Modeling**:
   - Beyond correlation to true causal relationships
   - Counterfactual reasoning for prediction under change
3) **Multi-Model Integration**:
   - Seamless combination of process, causal, and system models
   - Consensus-based prediction from diverse model outputs
4) **Adaptive Problem Solving**:
   - From human problems to formal representations
   - Multi-level abstraction for strategic decision-making
5) **Continuous Learning**:
   - RL for optimal information application
   - Virtual experimentation for model validation
   - Dynamic model adaptation to new conditions

VII. Discussion

The results obtained should allow us to move on to a scientifically sound choice of tools and modeling technologies for the tasks being solved, and then to the choice of methods and techniques for solving problems of pragmatic analytics. It seems necessary to consider model-oriented architecture [51], model-oriented engineering, service-oriented architecture [52], [53], ontology-based software modeling and development [54], and solvers [55], [56] as candidates for such tools and technologies. Pragmatic analytics studies people's experience in solving pragmatic problems based on an analytical representation of problems and their solutions. Generalization of this experience in the direction of using IAA, including based on formal means, should make it possible to solve various types of tasks more successfully, including using the experience generalized by pragmatic analytics and all those models and methods that can be used in the organization and management of activities. Significant progress in this area can be achieved using machine learning and knowledge management methods. As a result, IAA can be considered as a systemic discipline of multidisciplinary nature — a bridge between various natural science-based aspects of various activities, as diverse as those of a humanitarian, economic, technical, military nature, and the results of activities that are required. It should be noted that information is displayed in natural science and humanitarian theories through the prism of activity, since it is during the implementation of activity that the results described by natural science theories manifest themselves, and it is in activity that information is used at various stages. This kind of information research is pragmatic and consists of solving several pragmatic problems and performing various information activities, which in their entirety are considered. Pragmatic tasks are the tasks of defining information as answers to the questions of the tasks. These answers should make it possible to improve performance in certain conditions. Models and methods can be used to identify (search for) solutions to pragmatic problems: heuristic, formal, machine learning, complex models and methods for finding solutions of various types. Solutions to general problems are considered [57] in the framework of Human Problem-Solving, General Problem-Solving. In the case of heuristics use, solutions are sought based on generalization and presentation of knowledge about the problem in a conceptual, for example, verbal, form. Formal models and methods allow you to map problems into their mathematical image, solve them using mathematical methods, and return to the original conceptual problem. At the same time, different models, methods, and representations of tasks can be used. Machine learning models and methods allow you to display the available data in a model, which can be both formal (as a rule) and heuristic. Complex models and methods combine different types of models and methods. In the light of the above, activity models should be considered as a means of integrating models of reality and models of thinking, a means of describing mental activity, describing various aspects of using the results of thinking and information, and as a means of solving pragmatic problems. The formal search for information, as answers to the questions of such problems, can be implemented in different ways. For example, based on logical inference or based on the analysis of business models using information. Or — based on solving mathematical problems of finding solutions, optimization, and game theory problems. Solutions to problems of this type can be represented as sets of elements, relations and predicates defined on them. Thus, this type of solution is structural. Models that allow reflecting quantitative results into measurable spaces play a special role. For example, with points of such spaces — states of systems in the implementation of activities and trajectories in space as realizations of activities in certain environmental, information and organization conditions. This view allows us to provide detailed quantitative answers to the questions of the tasks, for example, how much, when and what resources should be supplied to obtain the best possible quality of functioning. What quantitative characteristics of organizational and management impacts should be implemented and when, to ensure the implementation of activities with the necessary quantitative characteristics in changing conditions and under various information supplied. Unfortunately, such models are still insufficiently developed. It should be noted that, as a rule, task questions require complex answers, i.e. both structural and quantitative, and on the other hand, the difference between these two types of answers (and tasks) is conditional, since, for example, the presence of a relationship can be quantified (1 in the corresponding row of the matrix), and vice versa. A non-zero amount of a resource of some kind required by a given time means that there is a required relationship with the resource. Nevertheless, different theoretical tools are adapted in different ways to provide answers to different (structural, quantitative, complex) questions. It seems that the use of complex models — including an interconnected set of heuristic and formal components, as well as, possibly, a component of machine learning models — is necessary to solve problems of a complex, both structural and quantitative, functional type.

Main concepts of the new info-activity approach (IAA) to machine learning and problem solving suggested. It was shown that this approach correlates better with typical human behaviors when human problems are solved. Main features of the proposed info-activity method for machine learning are described. Artificial activity concept and its main features elaborated as a result. The essence of the approach and artificial activity concept consists in the explicit representation of actions, information and human problems of various LoA to be solved in the implementation of the activity, as the processes of determining the answers to questions which arise during activity. IAA distinguishing feature is that, according to it, pragmatic aspects are considered another way than according to the traditional concept of obtaining material results of activities according to specified material requirements. Pragmatism of info-activity in IAA consists in obtaining and using information, decisions in such a way, activity will change and evolve in the best possible way. The required result of an activity can be awareness of changing goals, new knowledge about nature and its interaction with humans, desired relationships between humans, humans and nature, and knowledge of the patterns of different types of activities. In this understanding, the pragmatic aspect in IAA concerns economics, social organization, politics, sustainable development, education, science, and culture. As a result, the tasks of pragmatic analytics, solved within the framework of the IAA, can allow solving many significantly diverse and important tasks. To solve such problems, it is necessary to carry out a system of research activities which are considered as most appropriate. Among them: Conceptual framework for research within IAA. First, it is necessary to reveal concepts required, patterns of sets of sequences formation of causal relationships of informational (Infons, information substates) and other states, patterns of alternation of such sequences, the formation, and solution of problems when these sequences alternated, the role of humans and technical devices in the formation and alternation of sequences of states of different types. It is proposed to research logical foundations of IAA (for example, logic in reality — IL, information Logic — Information Logic, LIR, Logic in Reality, information theoretical logic: Information Theoretical Logic, ITL, situational logic — Situation Logic, logic of actions — AL, Action Logic) to describe the logical aspects of information manifestation [39], [58]; It is proposed to study candidate languages for operating with sequences of cause-effect relationships chains mentioned and their alternation due to information obtained, languages for describing tasks of activity using information. For example, GOLOG, STRIPS, PDDL [59], as well as various types of Action Languages [60]–[62], and Situation Calculus [63] can be considered candidates. It is necessary to propose data structures for highly efficient search, storage, and operation of sequences, including when using modeling languages and education environements [64], [65]. Graph databases, hierarchical structures (XML), and specialized representations based on labeled hierarchical trees and networks can be considered as data structures used.

VIII. Conclusion

New, info-activity approach to solving human activity problems and for new kinds of machine learning suggested, its main features discussed. Approaches are based on the analogies to the ways human problems arise during activity and the ways humans solve these practical problems during human action. Its essence consists in the explicit representation of human problems to be solved in the implementation of the activity, as the processes of determining the answers to questions which arise due to changed conditions and information obtained. Questions and answers, as reactions to questions, are informational. The process of obtaining answers can take various forms, such as search, synthesis, generation, prediction, and justification of answers. The research agenda to elaborate the suggested approach. It is planned that because of the suggested approach elaboration, it should be possible to solve more pragmatic problems (problems that arise during activity) that are usually solved by humans and to solve many problems based on predictive mathematical models and methods application. The info-activity method of machine learning presented, which is based on the representation of possible processes for solving pragmatic problems in the form of mappings from source “Big data” into possible descriptions of solutions (answers). It is shown that the use of the proposed method can significantly increase the number of problems solved, including, but not limited — by machine learning methods.

References

[1] J. McCarthy, “The philosophy of ai and the ai of philosophy,” in *Philosophy of Information*. Elsevier, 2008, vol. 80, pp. 711–740.

[2] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[3] G. Wiederhold and J. McCarthy, “Arthur samuel: Pioneer in machine learning,” *IBM Journal of Research and Development*, vol. 36, no. 3, pp. 329–331, 1992.

[4] L. Floridi, *Philosophy and computing: An introduction*. London and New York: Routledge, 1999.

[5] ——, “The method of levels of abstraction,” *Minds and Machines*, vol. 18, no. 3, pp. 303–329, 2008.

[6] ——, *The philosophy of information*. Oxford and New York: Oxford University Press, 2011.

[7] ——, “Information quality,” *Philosophy & Technology*, vol. 26, no. 1, pp. 1–6, 2013.

[8] L. Floridi and P. Illari, *The Philosophy of Information Quality*. Cham: Springer International Publishing, 2014, vol. 358.

[9] M. Burgin, “Information theory: a multifaceted model of information,” *Entropy*, vol. 5, no. 2, pp. 146–160, 2003.

[10] M. S. Burgin, *Measuring power of algorithms, computer programs and information automata*. New York: Nova Science Publishers Inc, 2010.

[11] Gordana Dodig Crnkovic and Wolfgang Hofkirchner, “Floridi’s open problems in philosophy of information, ten years later,” *Information*, vol. 2, no. 2, pp. 327–359, 2011.

[12] M. S. Burgin and G. Dodig Crnkovic, Eds., *Theoretical information studies: Information in the world*, ser. World scientific series in information studies. New Jersey: World Scientific, 2020, vol. vol. 11.

[13] M. Komova and A. Peleshchyshyn, “The concepts of philosophy of information by luciano floridi,” *Visnyk of Kharkiv State Academy of Culture*, vol. 0, no. 55, 2019.

[14] Mark Burgin, *Theory of Knowledge: Structures and Processes*. New Jersey: World Scientific, 2017.

[15] Pieter Adriaans and Johan van Benthem, *Philosophy of Information: Handbook of the Philosophy of Science*. Elsevier, 2008.

[16] C. Shannon and W. Weaver, *the mathematical theory Of communication*. University of Illinois Press, 1949.

[17] A. N. Kolmogorov, “Three approaches to the definition of the concept ”quantity of information”,” *Probl. Peredachi Inf.*, vol. 1, pp. 3–11, 1965.

[18] A. Ursul, *On the nature of information. Philosophical essay. (In Russian)*. Moscow: Poitizdat, 1968.

[19] Kolin K, “Philosophical theses about the nature of information,” *Bulletin of the International Academy of Sciences*, vol. 1, pp. 52–58, 2015.

[20] K. Kolin, “Philosophy of information and fundamental problems of modern informatics,” in *Proceedings of The 4th International Conference on the Foundations of Information Science*. Basel, Switzerland: MDPI, 21.08.2010 - 24.08.2010, p. 283.

[21] A. Geyda, “Dynamic capabilities indicators estimation of information technology usage in technological systems,” in *Recent Research in Control Engineering and Decision Making*. Springer International Publishing, 2019, pp. 379–395.

[22] N. Callaos, N. Callaos, S. Hashimoto, N. Lace, B. Sánchez, and M. Savoie, Eds., *Proceedings of the 14th International Multi-Conference on Complexity, Informatics and Cybernetics: IMCIC 2023*, ser. Proceedings of the International Multi-Conference on Complexity, Informatics and Cybernetics. International Institute of Informatics and CyberneticsWinter Garden, Florida, United States, 2023.

[23] P. Checkland and J. Poulter, *Learning for action: A short definitive account of soft systems methodology and its use for practitioner, teachers, and students*. Hoboken NJ: Wiley, 2006.

[24] T. Stonier, *Information and meaning: An evolutionary perspective*. Berlin and New York: Springer, 1997.

[25] N. Callaos and B. Callaos, "Toward a systemic notion of information: Practical consequences," in *Proceedings of the 2001 InSITE Conference*, ser. InSITE Conference. Informing Science Institute, 2001.

[26] M. Reynolds and S. Holwell, *Systems Approaches to Making Change: A Practical Guide*. London: Springer London, 2020.

[27] Doede Nauta, *The meaning of information (Approaches to semiotics)*. Mouton, 1972.

[28] J. Newman, "Some observations on the semantics of "information"," *Information Systems Frontiers*, vol. 3, no. 2, pp. 155–167, 2001.

[29] Bachelard G., *The formation of the scientific mind: A contribution to a psychoanalysis of objective knowledge.* Manchester: Cinamen, 2002.

[30] Patrick Allo, Ed., *THE Pi RESEARCH NETWORK: THE PHILOSOPHY OF INFORMATION AN INTRODUCTION*. Pi Research, 2013.

[31] A. I. Goldman, *Theory of Human Action*. Princeton University Press, 2015.

[32] J. P. Davis, K. M. Eisenhardt, and C. B. Bingham, "Optimal structure, market dynamism, and the strategy of simple rules," *Administrative Science Quarterly*, vol. 54, no. 3, pp. 413–452, 2009.

[33] T. Dittrich, *Information Dynamics*. Cham: Springer International Publishing, 2022.

[34] P. Illari and F. Russo, *The Routledge Handbook of Causality and Causal Methods*. New York: Routledge, 2024.

[35] J.-G. Ganascia, "Abstraction of levels of abstraction," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 27, no. 1, pp. 23–35, 2015.

[36] E. T. Mueller, *Commonsense reasoning: An event calculus based approach*, 2nd ed. Waltham MA: Morgan Kaufmann an imprint of Elsevier, 2015.

[37] Y. Gurevich and I. Neeman, "Logic of infons," *ACM Transactions on Computational Logic*, vol. 12, no. 2, pp. 1–28, 2011.

[38] S. D'Alfonso, "On quantifying semantic information," *Information*, vol. 2, no. 1, pp. 61–101, 2011.

[39] R. Loukanova, "Situation theory, situated information, and situated agents," in *Transactions on Computational Collective Intelligence XVII*, ser. Lecture Notes in Computer Science, N. T. Nguyen, R. Kowalczyk, A. Fred, and F. Joaquim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, vol. 8790, pp. 145–170.

[40] A. Geyda, "Concept and models of information application for actions in systems," *IJERTS*, 2023.

[41] ——, "Analytical research on system capability and information technology use capability: Problem statement examples," in *2020 26th Conference of Open Innovations Association (FRUCT)*. IEEE, 20.04.2020 - 24.04.2020, pp. 1–9.

[42] ——, "Models and methods of optimal information operations use for system functioning," in *Proceedings of the 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019)*. Paris, France: Atlantis Press, 2019, pp. 15–22.

[43] V. Kaptelinin and B. Nardi, "Activity-centric computing," in *Activity Theory in HCI*, ser. Synthesis Lectures on Human-Centered Informatics, V. Kaptelinin and B. Nardi, Eds. Cham: Springer International Publishing, 2012, pp. 73–87.

[44] S. Yarosh, T. Matthews, T. P. Moran, and B. Smith, "What is an activity? appropriating an activity-centric system," in *Human-Computer Interaction – INTERACT 2009*, ser. Lecture Notes in Computer Science, T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. O. Prates, and M. Winckler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5727, pp. 582–595.

[45] J. E. Bardram, S. Jeuris, P. Tell, S. Houben, and S. Voida, "Activity-centric computing systems," *Communications of the ACM*, vol. 62, no. 8, pp. 72–81, 2019.

[46] M. I. Aldinhas Ferreira, J. Silva Sequeira, and R. Ventura, *Cognitive Architectures*. Cham: Springer International Publishing, 2019, vol. 94.

[47] "What is intelligent automation? business automation." [Online]. Available: https://www.ibm.com/think/topics/intelligent-automation

[48] Preetpal Singh, "Simplifying intelligent automation adoption for businesses of all sizes: It pro today," 2924. [Online]. Available: https://www.itprotoday.com/it-automation/simplifying-intelligent-automation-adoption-for-businesses-of-all-sizes

[49] C. Cotrini, Y. Gurevich, O. Lahav, and A. Melentyev, "Primal infon logic with conjunctions as sets," in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7908, pp. 236–249.

[50] N. Bjørner, G. de Caso, and Y. Gurevich, *From primal infon logic with individual variables to datalog*. Berlin, Heidelberg: Springer-Verlag, 2012, p. 72–86.

[51] U. Borchardt, T. Kwast, and T. Weigel, "Integrating the is success model for value-oriented kms decision support," in *Business Information Systems Workshops*, ser. Lecture Notes in Business Information Processing, W. Abramowicz and A. Kokkinaki, Eds. Cham: Springer International Publishing, 2014, vol. 183, pp. 168–178.

[52] M. Adams, A. V. Hense, and A. H. ter Hofstede, "Yawl: An open source business process management system from science for science," *SoftwareX*, vol. 12, no. 3, p. 100576, 2020.

[53] S. Carter, J. King, M. Younkers, and J. Lothian, *Model-driven DevOps: Increasing agility and security in your physical network through DevOps*. Boston: Addison-Wesley, 2023.

[54] F. S. Parreiras, *Semantic web and model-driven engineering*. Hoboken N.J.: Wiley and IEEE Press, 2012.

[55] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann, "A review and comparison of solvers for convex minlp," *Optimization and Engineering*, vol. 20, no. 2, pp. 397–455, 2019.

[56] D. Riley and K. A. Hunt, *Computational Thinking for the Modern Problem Solver*. Chapman and Hall/CRC, 2014.

[57] Allen Newell and Herbert Alexander Simon, *Human Problem Solving*. Prentice-Hall, 1972.

[58] J. M. Sagüillo, "One sense of 'information': A quick tutorial to information-theoretic logic," *tripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, vol. 7, no. 2, pp. 179–184, 2009.

[59] P. Eyerich, B. Nebel, G. Lakemeyer, and J. Claßen, "Golog and pddl," in *Proceedings of the 2006 international symposium on Practical cognitive agents and robots - PCAR '06*, X. Chen, W. Liu, and M.-A. Williams, Eds. New York, New York, USA: ACM Press, 2006, p. 93.

[60] J. J. Alferes, F. Banti, and A. Brogi, "An event-condition-action logic programming language," in *Logics in Artificial Intelligence*, ser. Lecture Notes in Computer Science, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. Fisher, W. van der Hoek, B. Konev, and A. Lisitsa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4160, pp. 29–42.

[61] E. Giunchiglia and A. Tacchella, "Optimal planning with expressive action languages as constraint optimization," in *Logics in Artificial Intelligence*, ser. Lecture Notes in Computer Science, S. Gaggl, M. V. Martinez, and M. Ortiz, Eds. Cham: Springer Nature Switzerland, 2023, vol. 14281, pp. 621–637.

[62] V. L. Michael Gelfond, "Action languages," *Linkopig Electronic Articles in Computer and Information Sciences*, 1998.

[63] F. Lin, "Chapter 16 situation calculus," in *Handbook of Knowledge Representation*, ser. Foundations of Artificial Intelligence. Elsevier, 2008, vol. 3, pp. 649–669.

[64] K. V. B., I. G. Rosenberg, and M. Goldstein, *Structural theory of automata, semigroups, and universal algebra*, ser. NATO science series. Series II, Mathematics, physics, and chemistry. Dordrecht and Great Britain: Springer, 2005, vol. vol. 207.

[65] V. B. Kudriyavtsev, P. A. Aliseichik, K. Vashik, J. Knap, A. S. Strogalov, and S. G. Shekhovtsov, "Modeling learning process," *Journal of Mathematical Sciences*, vol. 172, no. 5, pp. 700–733, 2011.