Knowledge Flux for Measuring the Intensity of the Ubiquitously Flowing Intelligence using Denotational Mathematics

JOHN SARIVOUGIOUKAS[1], ARISTIDES VAGELATOS[2]
[1]IT Department, [2]Continuous Education & Training
[1]General Hospital of Athens, [2]CTI
Athens
GREECE

Abstract: - The formally defined mathematical framework of Denotational Mathematics, provides the opportunity to use existing mathematical tools to define the dimensionless magnitude of knowledge flow. In this paper, "knowledge flux" is defined in order to provide a quantified expression of the developing knowledge with which the knowledgeable control of the physical space can be achieved. Moreover, the data flux can be used as a measure of the interaction between the processes carried in the physical and the corresponding virtual domain. In addition, the data flux can be used to characterize the efficiency of the symbiotic partner agent software which facilitates the interaction between the physical and virtual spaces. As a result, the indicative interaction between the physical and the virtual spaces can be quantified and measured with objective evidence.

Key-Words: Knowledge flux, Context-awareness, Prerequisites & challenges, Knowledge evolvement, Static & Dynamic Representation of Knowledge, Process & dynamic effects, Dynamic knowledge

Received: July 14, 2023. Revised: December 22, 2023. Accepted: February 7, 2024. Published: April 2, 2024.

1 Introduction

The Weiser's [41] seminal research work placed the foundations for an alternative scientific vision using the term "ubiquitous computing". Since then, the technological advancements of internet and mobile devices extended the scientific research frontiers beyond the local applications and the entire globe participates in a wider network referred as the Internet of Things (IoT). Weiser's vision led to smart environments handling the developing knowledge to support decisions making with context-aware software [1] sensing and adapting according to the evolving situations [8]. In smart environments, software is handling and distinguishing the physical and virtual aspects of the formed situations [11] supported by layered software structures [23]. However, there is still a lack of prototypes, standardization, and the intermediateness of human interaction and expectations [5]. Sensing the smart environment involves data management, reasoning, and requires adaptability meeting the requirements achieving adequate each time behavior [24]. The complexity of the formed software systems is due to semantics, knowledge reasoning, and situational conditions along with the developed relationships in the evolving activities of the participating roles [2]. Thus, context can be described with semantic graphs consisting of concepts administered with Semantic Web Languages (RDFS) [4] and supported by dedicated ontologies [2]. The ontologies support the developed static and dynamic knowledge along with applying rules and user preferences [10]. Also, ontology-based knowledge facilitates the inference process with the development of semantic frameworks [15] employing specialized taxonomies implemented from instantiated classes expressing functional properties and data attributes. The employed classes represent the static aspects of the developed knowledge while object orientation provides the dynamic aspects of knowledge evolution [3] over time. Knowledge is treated to convert the contextual data into higher forms of knowledge in order to extract additional information from the available context, and lastly, to extend the holding knowledge spectral content [21]. The knowledge that supports the evolving context can be sensed in various compositional forms such as tacit or explicit, declarative or procedural, deep or surface knowledge [21]. It is accustomed to consider knowledge to be explicit, tacit, and personal [16]. In addition, autopoietic knowledge is produced by the processed data [16]. Moreover, knowledge can be viewed as the combination of contextual information and experience mixed with contextual inference, interpretation, and reflection [7] while experience is considered to be pumped from past or historic data. Past contextual data can be used as knowledge bases supporting premises for ordering logic predicates which can be processed further to develop Bayesian networks to support reasoning [20] and decision making. Such networks can be processed by software with resource description frameworks (RDF) as proposed by W3C since they consist of first order logic transactions presenting the dynamic aspects of knowledge. The knowledge tracing problem is faced using past performance of the system reflecting on the knowledge level and trying to predict the future performance based on the current perception [12]. Therefore, knowledge is changing over time according to the developing information created by the supervising system and the prediction of the learnt knowledge by the system is still an open problem [12] with knowledge to be dynamically evolving.

The paper continues with a brief orientation regarding the characteristics, the challenges, the necessary supportive methods, and the explored architectures of the underlying framework.

2 Background Framework

The operation of *context-aware* software designs requires the integration and support from ad-hoc networked sensors and artifacts integrated into systems [11]. The availability of sensors provides a wealth of raw data which must be adequately stored, processed, and correlated in order to contribute to the development of the holding context. Also, ad-hoc networked hardware artifacts with embedded software designs participate in deriving the formation of contexts and supporting the user's activity implementing Marc Weiser's vision regarding ubiquitous computing environments [41]. The ubiquitous computing environment can be distinguished into the physical and logical areas [11] which can be referenced interchangeably pointing at layered structures [13] in order to support the management of the performing systems consisting of hardware and software artifacts. Among the proposed layers, also, the processing layer is included, which can handle Artificial Intelligence (AI) algorithms to provide the required intelligence [23]. The resulting complex applications are characterized by the capability to sense and adapt, as well as, discover resources and augment the formed context [8]. Therefore, the application of context-aware systems tends to attain Weiser's vision of providing the availability of well-defined software and hardware systems supporting the user's performed tasks.

The *challenges related to context-awareness* are associated with the lack of prototypes, standardization, and the intermediateness of human interaction and expectations [5]. Also, the gathered data from the contextual environment must be input to software systems' applications to perform the necessary data management, the required reasoning, and the needed adaptability to present the adequate each time behavior [24]. Such software systems perform complex operations in order to be capable of satisfying the contextual and user needs employing approaches from the area of AI and machine learning areas building stacks of methods. The applying methods vary from simple First Order Logic to Deep Learning software applications in order to obtain the proper set of data to achieve awareness and support the user's Activities of Daily Living (ADL) and the everyday ordinary socializing consisted of compound tasks [19]. Thus, the complexity of the formed software systems can be approached as the resultant of the following three components: semantics, knowledge reasoning, and situational conditions along with the developed relationships in the evolving activities of the participating roles [2]. According to the authors of [2], context can be described with semantic graphs consisting of concepts administered with Semantic Web Languages (RDFS) [4] and supported by dedicated ontologies [2]. Thus, the management of ontologies supports the developed static and dynamic knowledge along with applying rules and user preferences [10]. Knowledge connects the high-level user requirements with the low-level hardware capabilities to support the activity of inference for context-awareness.

In ubiquitous computing environments, inference can be obtained applying various methods such as probabilistic networks, clustering, rule or case-based reasoning, AI, and machine learning. Ontology-based knowledge facilitates the inference process with the development of semantic frameworks [15] consisting of specialized taxonomies implemented with instantiated classes expressing functional properties and data attributes. The employed classes represent the static aspects of the developed knowledge while object orientation provides the dynamic aspects of knowledge evolution [3] over time. The employment of ontology frameworks provides generic structures with loosely coupled taxonomies which are capable of providing semantic prescriptions and instructions to augment the operating software's semantic context along with interpreting and operating capabilities appearing in the processes carried in the physical environment revealing the advanced concept of process-awareness [25]. Hence, there is a stack which can be traversed and which is based on context covering all the layers from the raw data to business processes hiding the intermediate layers and providing the means for interaction to the end-user.

Context-awareness requires such a computer architecture that demands loosely coupled software applications to interoperate under a given scenario. Such setups can be met in Service Oriented Architecture (SOA) which are supported by open standards such as WSDL, SOAP, and BPEL and implemented independently from the employed programming languages. Another architectural paradigm concerns the Service Delivery Platform (SDP) which uses networks of Internet Protocol (IP) addresses. Both SOA and SDP architectures support the event driven requirements needed for context-awareness providing services orchestration capabilities. Thus, the software designer separates the concerns and is devoted to the management of the carried processes [22]. Such architectural frameworks provide the ability to re-use software modules and develop only the additionally required each time. Moreover, the availability of the internet provides a wealth of available services that can be obtained even in cases where the local hardware presents limitations as in the “Software As A Service” (SAAS) case paradigm. Therefore, on top of context-aware applications, there can be developed software frameworks where the software designer is concerned with the dynamic aspects of processes and the provided knowledge management.

The state of knowledge reflected by the contextual environment, including the end-user, is changing dynamically along with the change in context. Thus, as context is changing the corresponding state of knowledge is changing to reflect the dynamically occurring changes. However, the changes in context are not directly proportional to the supporting knowledge. Moreover, knowledge is treated to convert the contextual data into higher forms of knowledge in order to extract additional information from the available context, and lastly, to extend the holding knowledge spectral content [21]. The knowledge that supports the evolving context can be sensed in various compositional forms such as tacit or explicit, declarative or procedural, deep or surface knowledge [21]. It is accustomed to consider knowledge to be explicit, tacit, and personal [16]. In addition, knowledge is developed by the processing of contextual data and the autopoietic knowledge which is produced by the processed data [16]. Also, knowledge can be viewed as the combination of contextual information and experience mixed with context, interpretation, and reflection [7] with experience considered to be pumped from past or historic data. Past contextual data can be used as knowledge bases supporting premises for ordering logic predicates which can be processed further to develop Bayesian networks to support reasoning [20] and decision making. Such networks can be processed by software with resource description frameworks (RDF) as proposed by W3C since they consist of first order logic transactions presenting the dynamic aspects of knowledge. Computerized systems face the knowledge tracing problem using past performance of the system reflecting on the knowledge level and trying to predict the future performance based on the current perception [12]. Therefore, knowledge is changing over time according to the developing information created by the supervising system and the prediction of the learnt knowledge by the system is still an open problem [12] with knowledge to be dynamically evolving.

“Knowledge Flux” is a recently introduced term coined by the authors to express the concerns of the quantitative interactions of smart and ubiquitous environments exchanging knowledge. The level of intensiveness of knowledge must be measurable in order to perform comparisons and manage ubiquitous computing environments.

The rest of the paper starts with references to the related literature, it carries on with the development of a model accommodating knowledge, it goes on with the determination and formal definition of the data flux for knowledge transfers, it keeps on with a discussion, and it finishes up with the drawn conclusions.

3 Related Literature

Knowledge can be expressed using the terms of concept, object, attribute, and relationships along with the formally defined Concept Algebra to perform algebraic manipulations [34]. The concept is considered to be the unit of knowledge and it is defined [34] as the tuple of

$$\mathrm{C} \triangleq (A, O, R^c, R^i, R^o) \tag{1}$$

where, A is the set of attributes of a concept "C", O is the set of objects of a concept "C", R^c is the set of relationships among the objects and attributes, R^i is the set of input relationships from other coexisting concepts, and R^o is the set of output relationships with other coexisting concepts.

It is apparent from the above definition that the concepts can dynamically form networks or trees of concepts. Moreover, intention of a concept is the set of its semantic uses expressed with the concept's attributes and properties while extension of a concept is the set of its semantic uses beyond its intended uses, sharing attributes and properties with other formal concepts. Networks of concepts are defined within the universe of discourse of knowledge which is defined [34], as

$$\mathfrak{U} \triangleq (\mathfrak{D}, \mathfrak{A}, \mathfrak{R}) \tag{2}$$

where, $\mathfrak{O}$ refers to the set of objects, $\mathfrak{A}$ corresponds to the set of attributes, and $\mathfrak{R}$ denotes the set of relationships holding in the formed hyperstructure which is depicted in Fig.1.

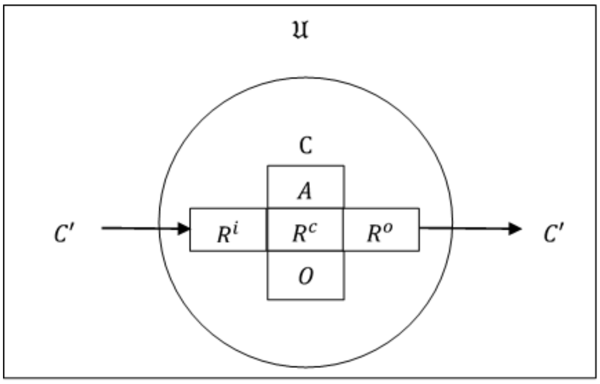

Figure-1. Model of an abstract concept in the universe of discourse of knowledge.

Networks of concepts are formed by the input and output associations developed among the participating concepts. The networks of concepts are developed within the locally holding universe of discourse of knowledge which can be distinguished and characterized as open or closed. Further, the networks of concepts can be analyzed and processed with a layered structured model consisting of three layers, the concept, the object, and the attributes layers [34].

It is worth mentioning that analytical mathematics provides the capability of developing matrices of simultaneous equations for each of the layers including the input and output parameters in order to describe the behavior of the formed system of concepts. However, such an attempt requires excessive processing power in order to obtain practical results. Alternatively, the use of Denotational Mathematics, as it is described below, provides the required behavior of complex systems.

Processing of networks of concepts can be performed by a set of algebraic operations formally defined by Concept Algebra [34]. Denotational Mathematics provide the means to algebraically support the performed manipulations which are distinguished into relational, reproductive, and compositional semantic operations. Also, in order to face the concepts' intentions and extensions, granular modeling [21] becomes a necessity in order to overcome the excessive appearance of inherited complexity [34].

Symbiotic Computing is the revealed necessity of the coexistence between computers and the internet along with society. Moreover, the ubiquitous computing paradigm enhances cooperation and coexistence of people and computers. The man-machine paradigm evolves to the development of the Information Society in spite of the accounted technical, social, and economic problems and leads to the development of more advanced, if not just different, computing paradigms [41]. In the symbiotic context, the discrete spaces of Physical Space (PS) and Virtual Space (VS) are coupled and coexist. In between the two spaces there exist the Community-Agents supporting four function models: Perceptual Functions, Social Functions, Cognitive Functions and Decision Functions [31]. Thus, according to the Symbiotic Computing paradigm, between the user and the coexisting computer there is an interacting symbiotic partner agent software facilitating the interaction of PS and DS spaces, Fig.2. The developed model of that software is analogous to the Layered Reference Model of the Brain (LRMB) (On Cognitive Computing) with characteristics borrowed from the Cognitive Machines [14].

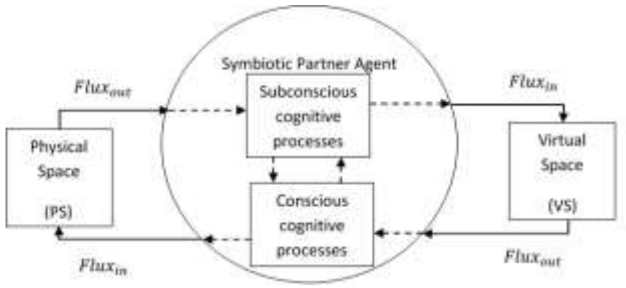

Figure-2. Symbiotic partner agent model.

The *partner agent* can facilitate the transfer of the developed knowledge between the PS and VS spaces. However, the representation of the developed knowledge can be implemented with software modules taking advantage of the functional rules provided by Concept Algebra. The rules of Concept Algebra can be applied on the Object-Attribute-Relation model specified in the Concept definition in order to develop software artifacts to represent the holding knowledge between PS and VS spaces. The software artifacts constitute modules to parse the formed network of concepts, compile the syntactic adequacy of the concepts interconnected in the formed network, develop the knowledge base with the identified intentions and extensions (open or closed) of the formed network, and visualize the acquired knowledge in some form, for instance, as an application of LRMB [29].

Examining the model of the symbiotic partner agent, it is apparent that there is a continuous exchange of messages - information. Thus, the knowledge state changes when information containing additional concepts are injected into the symbiotic partner agent. The resulting new knowledge state can be expressed by the composition of the existing network of concepts with the addition of the received concepts. Hence, the dynamic representation of knowledge is performed by the Concept Algebra operation of composition which is defined [37] by composing the participating structures, as follows:

$$OAR_{new}\mathbf{ST} = OAR_{existing}\mathbf{ST} \uplus OAR_{additional}\mathbf{ST} \quad (3)$$

Hence, the dynamic knowledge, $OAR_{new}\mathbf{ST}$, is achieved by successive compositions of concepts. The performed concepts composition is an adaptive process allowing the integration of new concepts to existing networks of concepts [37]. Thus, the state of a concept is changing according to the interactions with the coexisting concepts forming networks of concepts representing the holding knowledge state.

The co-existence of a number of associated and interacting networks of concepts provide the development of systems of concepts. Each part of the system inherits the characteristic aspects of the constituting concepts as defined in the OAR model along with the behaviors expressed by each network and the constraints applying to each subsystem of concepts [36]. The associated subsystems of a system of concepts which is within a proper hyperstructure has the capability to perform operations from the following categories of relational, reproductive, and compositional operations as well as composition with additional subsystems [36] forming large systems within the functioning hyperstructure. Hence, the knowledge represented by such a system of network of concepts is expressed as

$$K = \Re_{CA}: X_{i=1}^{n} C_i \rightarrow X_{j=1}^{n} C_j \quad (4)$$

where $\Re_{CA}$ represents the applying set of operations formally defined in Concept Algebra [30].

Knowledge can be described by a static snapshot of a properly defined and constructed network of concepts, the way described in the field of Denotational Mathematics. However, knowledge is dynamic and evolves [38]. For instance, the learning process chooses the appropriate set of concepts to develop knowledge and criticism allows the developing knowledge to evolve further changing and building additional relationships upon the associated - existing relationships among the concepts in the system of networks of concepts [38]. Knowledge presents dynamic aspects and evolves by incrementally altering the relationships among the nodes of the networks of concepts mimicking the human brain's operation and defining a magnitude of knowledge expressed by

$$KnowledgeDifferential = \frac{d(OAR)}{dt} \quad (5)$$

4 The concept of flux

The dynamic behavior of networks of concepts is following the operations performed by the human brain to transform received input of concepts or relations among concepts into behaviors. The dynamic behavior of concept networks is described with the dimensionless magnitude of Flux which is defined as the difference between the data structures exported from the physical or virtual space and the data structures imported into that space. Also, input and output Flux can define a quotient to determine the activity of the applying system. Thus, when output over input Flux is equal to one then the system can be characterized as stable or knowledge neutral. However, when the Flux quotient is greater than one then the system is characterized affected by knowledge while when the Flux quotient is less than one, the system is less affected by knowledge as it is described in the following formula:

$$Flux = \frac{Flux_{out}}{Flux_{in}} = \frac{(OAR\mathbf{UDM})_{output}}{(OAR\mathbf{UDM})_{input}} = \begin{cases} < 1\ negatatively\ affecting \\ = 1\ stable\ or\ neutral \\ > 1\ positively\ affecting \end{cases} \quad (6)$$

From the equation (6), it is apparent that the dimensionless magnitude of Flux represents the quantitative affection of knowledge on a given space as it is given by the instantiation of the Unified Data Model $OAR\mathbf{UDM}$. For example, in a home UbiHealth environment [27], the Physical Space (PS) contains, among others, the data structures of five parameters: the patient's glucose level as it is measured by some device, an implanted insulin pump, the measured heart rate and blood pressure, as well as the time the data is transferred to symbiotic partner agent. In this example, the supporting system is going to process the received data structures and return a single data structure containing the appropriate value with which the implanted insulin pump is going to be adjusted at the patient's health needs.

Processing of the concept networks is performed by the Real-time Process Algebra (Tian, et al, 2011) (RTPA) algebraic environment which is formally defined with Denotational Mathematics. Denotational Mathematics in RTPA provides the separation of concerns regarding the static and functional behaviors of the underlying concepts in the physical or virtual spaces with Unified Data Model (UDM) and Unified Process Model (UPM), respectively. Thus, RTPA provides the means for knowledge representation and Real-Time Operating System plus (RTOS+) [31] provides with Denotational Mathematics the means to perform system, resources, and process management including multiple levels of interrupts handling. Therefore, the formal algebraic representation of advanced concepts and the necessary underlying operating system form a complete computing environment and the efficiency is expressed with Flux.

The symbiotic partner agent is in between the PS and VS spaces with the capability to function autonomously. In Figure-3, the symbiotic partner agent model is modified in order to acquire those modules that provide the necessary autonomy functionality performing decision-making and the goal-driven processes [38]. The autonomic behavior is already defined as the behavior depended upon the goal-driven, the decision-driven, and the interrupt-driven behaviors [40]. The interrupts are initiated by internal or external events and timing requests affecting the overall behavior of the system $B_{autonomic}$ in which the symbiotic partner participates and it is expressed by

$$B_{autonomic} \widehat{=} \{B_{goal}, B_{decision}, B_{interrupt}\} \quad (7)$$

In Figure-3, the symbiotic partner agent must be equipped with the autonomous agent (AA) which can be incorporated with the rest of the partner's modules [17]. The formed system includes the representation of the Generic Abstract Intelligence Model (GAIM) as formally defined in [38].

The knowledge developed in both PS and VS spaces can be expressed as the Cartesian product among all the participating concepts [33] within the holding universe of discourse of knowledge $\mathfrak{U}$. The operation of participants in PS and VS spaces forms the local knowledge base which is a hyperstructure $\mathfrak{A}$ denoting hierarchical structures of interconnected concepts with weighted relationships denoting the strength of their semantic relations [33], it is given by the following equation:

$$\mathfrak{K} \widehat{=} R_{i=1}^{n} C_i \mathrm{x} R_{i=1}^{n} C_i = R_{i=1}^{n} R_{j=1}^{n} (C_i, C_j) \quad (8)$$

The mathematical term R is an operator that represents repetitive or recursive operations, called the big-R notation [35] in the RTPA framework of Denotational Mathematics.

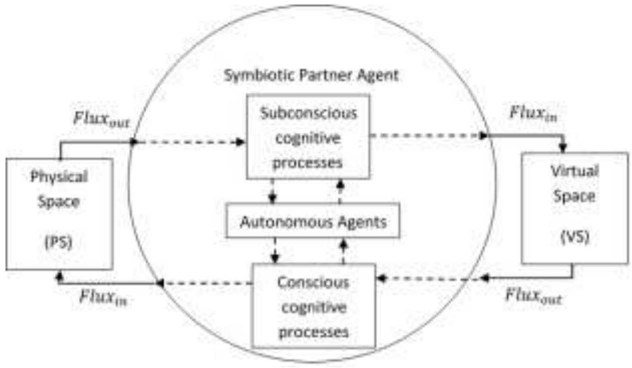

Figure-3. Autonomous Symbiotic partner agent model.

The hyperstructure $\mathfrak{A}$ can be manipulated and analyzed into three layers: the knowledge, the object, and the attribute layers [33]. Distinguishing the hyperstructure's contents into layers provides the capability of separation of concerns which means that an employed software agent can manipulate the attributes relations with the side effect to influence the contents of the other layers. However, the attributes mathematical manipulation requires the capability of moving on various levels of granularity taking advantage of the property of inheritance among concepts to represent in real-time abstract concepts or instantiations of concepts [21].

The knowledge of the hyperstructure $\mathfrak{A}$ can be processed with the support of the autonomic agent. The support of the autonomic agent refers, among others, to the consideration of the concepts of rough sets defining the membership of relative or implied concepts [40]. Thus, the employment of the terms defined by the rough sets membership functions can support the intended or extended use of concepts performing rough sets manipulations by including-excluding terms with sets operations. Moreover, the use of rough sets provides the capability to define the dependency among sets in order to provide support to the decision-making or the goal-driven aspects of the autonomic agent [40]. The decision-making process relies on the development of rules with a certainty provided by the membership of the examined concepts [40]. Hence, the hyperstructure $\mathfrak{A}$ provides the base to develop rules on each of the three layers using hierarchical structures of rules of varying certainty supporting the intension and extension operation on concepts in order to achieve approximations, dependencies, and reductions that lead to varying data granularity [40]. The relationships among concepts belonging to hierarchical structures is examined using probabilities, like in [28], with the known problems related to the unavailability of Bayesian posterior probabilities conditions. Therefore, calculating the concept's participation with membership functions of rough sets provides deterministic behavior to the design of a system.

The limitations of the symbiotic partner agent are related with two open issues which refer to the size and the homogeneity of knowledge. The size of the managed knowledge cannot be strictly defined for a general-purpose system. Also, the homogeneity of the concepts participating in the knowledge representation cannot be rigorously defined. However, the internet provides the capability to perform references to well-defined knowledge bases and exemplars reducing the demands for the size of the locally managed memory. In addition, the homogeneity of the concepts participating in the hyperstructure $\mathfrak{A}$ can be defined with the proper use of the functionalities of intentions and extensions. Therefore, the current limitations of size and concepts of homogeneity can be partially remedied with the use of the internet and the concepts functionality of intentionality and extensionality.

The hyperstructure $\mathfrak{A}$ supports the development of dynamical hierarchical networks of concepts which can be augmented or reduced using the functionalities of concept intentionality and extensionality. Knowledge is dynamic and capable of offering representations of static snapshots of the instantly holding knowledge, as well as, the knowledge generation [17]. The knowledge generation can be represented by the concept functionalities of intentionality and extensionality which can develop additional concepts, relations augmenting or shrinking the represented knowledge. Therefore, the relational composition of concepts leads to the development of complex concept constructs or newly defined concepts.

The development of concepts networks in the hyperstructure $\mathfrak{A}$ represent the developing knowledge and provide the capability to discover or extract the contained knowledge. The knowledge extraction can be performed by applying queries on the knowledge graphs or semantic networks consisting of interrelated concepts. In [9], the authors claim that literature information can be turned into knowledge when the received data is placed in a network and new knowledge can be generated by augmenting the network nodes-data-concepts developing extended knowledge graphs. Therefore, knowledge extraction from the concept networks is a technical challenge that can be addressed with Resource Description Framework (RDF), non-relational databases [9], and language representations of knowledge [18].

Knowledge graphs present similar data structures to networks of concepts and they share analogously practical problems such as the size of the constituting networks with large number of interrelated nodes, the consistency of the denotations of the participating nodes, and the operating environments. In addition, the parameters of each node increase the difficulty of the management administration of concepts networks. However, there are techniques to efficiently traverse the concepts network for searching, placing queries or interacting with questions-answers. In the administration of knowledge graphs besides the developed interrelations among nodes, it is attempted to enrich each node of the formed network with the semantic D codes. Moreover, each code can take up K values forming K-way D-dimensional codes [6] in order to present the similarity among the comparing nodes. Thus, simply traversing the concepts network or applying a query on them, an encoder/decoder model mechanism applies using a discretization / reverse discretization function [26] for representing complex or advanced forms of knowledge. Embedding characteristics is performed in [6] by a function F:V→R^d, where V represents the semantics of concepts placed in a table - vocabulary which corresponds to a vector in R^d [6]. However, as the size of the concept network is increasing there must be found ways to preserve efficiency and among them compression [6] is considered as promising.

In Denotational Mathematics, with a formally defined environment, the recurrent operations are defined with the big-R notation in the Real-Time Process Algebra (RTPA). Knowledge consisted of concepts networks is represented [35] by the following recurrent expression:

$$\mathfrak{K} \triangleq R_{k=1}^{n} \mathfrak{K}^k\left(\mathfrak{K}^{k-1}\right) = \mathfrak{K}^n(\mathfrak{K}^{n-1}\left(\ldots.\mathfrak{K}^1(\mathfrak{K}^0)\right)) \quad (9)$$

and base condition as,

$$\mathfrak{K}^0 = X_{l=1}^{m} C_l \quad (10)$$

where X is the cartesian product among C_l's. Denotational Mathematics provides the means to develop knowledge bases [35] with which knowledge can be acquired, fused in the base, manipulated and retrieved. Hence, there is a sound and complete logic environment to administer knowledge bases [35]. However, such knowledge bases develop relationships and references among the concepts over time undergoing a change of its contents due to its interaction with the environment. Thus, the behavior of the formed system of the concepts network is changing according to the built relationships among the concepts expressing the self-adjustment or the earned experiences from the interaction of the surrounding environment resulting in a cognitive dynamic system [14]. The applying rules of the concept's networks spawn the observed intelligence, learning, adaptivity, activity or action in real-time affected by the surrounding environment [14]. The operation of a system consisting of a knowledge base requires the support of a complete computing environment. Such a computing environment or system must be initialized, the components of the system are properly synchronized and sense internal and external events interrupting the executing flow of operations and re-schedules the operations. The internally carried operations must be managed using temporary and permanent memory banks. Thus, such a system collects operations and builds processes with which the system requests, creates, performs, runs, interrupts, completes, delays, suspends, kills, and dispatches the available sets of operations or processes [32]. Therefore, the operation of a knowledge base requires the support from a Real-Time Operating System to manage the carried internal processes and synchronize the requests from the surrounding environment with a Real-Time Operating System (RTOS+) [32].

5 Modelling

The description of the data and information exchanged among the ubiquitous computing participants and the system requires an adequately matching model. A brief description of the model is presented in the following as well as the carried procedures during the performed interactions, and the mathematical expression of the dynamic developing behavior.

A system is formed containing the physical context and the context developed into the supporting computer machine. Within this system there is always a continuous interaction between the two contexts. From the physical context, data is flowing to the computer machine while from the computer context data is actuating and affecting the operational terms of the physical context. The total flow of data between the physical and the computer contexts determines the holding knowledge of the formed system.

The physical context is formed by the activity carried by the artifacts and the users in the computer environment. The activity of the artifacts produces data that is driven to the co-functioning computer. The computer is processing the received data and produces data for the connected transducers to properly affect the artifacts and the users of the physical - environmental context. The computer receives data and produces data that maintain the level of the developed knowledge of the formed system. Figure-1 depicts the interaction between the environmental-physical and the computer contexts presenting the flows of data between the interacting contexts which is proportional to electromagnetic flux.

The flow of data or data flux is the measurable magnitude to represent the total knowledge which is developed in the interaction between a physical and a computer context. The data flux can be used to describe the induced knowledge to the active system such as that of Figure-1. The exchanged data between the physical and computer context contribute to knowledge components of the formed system. The data flux is the magnitude that is formed by the matching data structures of the physical and computer contexts. In other words, given a set B of data structures representing the physical context projected to the set A of data structures representing the computer context is the data flux which is expressed as $\Phi = B * A$. The flux Φ represents the number of procedures carried out in a system in order to develop the holding knowledge of that system. Another term borrowed from electromagnetics is the flux density which in our case is the division of contextual data structures over the holding knowledge's data structures. The contextual data structures are driven by the carried procedures which govern the operation of the occurring system.

When there is observed data flux across contexts then there are spawned processes. The spawned processes act analogously to the generation of voltage when a conductor is moving within a magnetic field. The processes can generate additional processes which can be internal or external to the system depending upon the nature of a closed or open system. Flux is changing along time at discrete time intervals expressed as $\Phi(t) = \Phi(t1)-\Phi(t0) = (\Delta\Phi1(t1)+ \Delta\Phi2(t1)+ \Delta\Phi3(t1)+.....) - (\Delta\Phi1(t0)+ \Delta\Phi2(t0)+ \Delta\Phi3(t0)+.....)$ representing the flux of the constituting knowledge components. Knowledge can be represented as the rate of change of flux given by $K = \Delta\Phi/\Delta t$.

6 Determination and Formal Definition of Data Flow

The sensed knowledge flow leads to the determination of the dimensionless magnitude of data flux. The static and dynamic aspects of flux are associated with the corresponding aspects of knowledge.

6.1 Data Flux

The interaction between the PS and VS spaces is controlled by the intermediate symbiotic partner agent. The interaction between the PS and VS spaces is sensed by the transmission/reception of data which can be performed by the direct connection of each PS or VS space with the symbiotic partner agent or through the use of some intermediate memory bank. The interaction between the PS and VS spaces is verified by the existence of data flow. The enumeration of the set of data structures from the PS space is denoted by F_{in} while the corresponding set received is denoted as F_{out}. When data is exchanged between PS and VS spaces then there are two options: (a) data status is changed, and (b) the concept network managed by the symbiotic partner agent is also changed. Thus, we define the magnitude of data flux F given by

$$F \triangleq \frac{F_{out}}{F_{in}} \tag{11}$$

with F_{in} being the data transferred from the PS to VS while F_{out} is the returned data from VS to PS. Thus, at any given instant of time t, the function $f(t)$ represents the flux F which takes values given by

$$f(t) = \begin{cases} < 1, \ cardinalities: |F_{in}| > |F_{out}| \\ = 0, \ cardinalities: |F_{in}| = |F_{out}| \\ > 1, \ cardinalities: |F_{in}| < |F_{out}| \end{cases} \tag{12}$$

The comparisons between the sets of F_{in} and F_{out} can be performed at the syntactic or the semantic levels. In any case, the comparison focuses on the magnitude

$$F_t = (F_{in} \cap F_{out}) \tag{13}$$

The instant value of flux F_t can be given by the symbiotic partner agent which can be analyzed with the application of Denotational Mathematics. The use of Denotational Mathematics facilitates the formal description of the computing environment that supports the operation of the symbiotic partner agent, as well as, the functionality of the symbiotic partner agent itself. The computing environment can be described and implemented with an appropriate modification of the Real-Time Operating System (RTOS+) [32]. In a similar fashion, the formal description of the symbiotic partner agent can be performed with the use of the available tools provided by the Real-Time Process Algebra (RTPA) given in [38].

6.2 Static representation of Flux

The representation of the PS and VS spaces can be achieved with the formal description of the existing data structures. Thus, the static representation of the flux is described by the intersection of the sets constituting the fluxes entering and leaving from the symbiotic partner agent. At a given instance of time, the flux F_{out} represents how the developing knowledge in the symbiotic agent affects the carried processes in the PS physical domain. Similarly, the flux F_{in} represents how the developing processes in the physical domain affects the build of knowledge in the symbiotic partner agent. Therefore, the static representation of flux is performed by the description of the data structures existing in the artifacts of the physical domain and the memory of the symbiotic partner agent.

6.3 Dynamic representation of Flux

The symbiotic partner agent is the software system that continuously and constantly supports the operation of a ubiquitous computing environment such as UbiHealth. This continuous cooperation and interaction between the symbiotic partner and the UbiHealth environment is driving the development of flux. The presence of flux leads to the development of knowledge which is the result of the changing in the transfer of structured data. The flux is the rate of change of the data exchanged between PS and VS. Therefore, the existence of flux leads to the development of knowledge and hence, the dynamic change of flux is the built knowledge by the symbiotic partner agent.

The knowledge represented by the holding status of the concept networks changes by the received flux. Thus, the state of the concept network changes upon the reception of additional data from the PS space and the governing rules of the concept network bring the network to another state depending upon the previous state. Hence, the concepts network state depends entirely on the previous one revealing a recursive relationship among the states. Moreover, the concept network requires an initialization leading to the conclusion that there must be a supervising real-time operating system that monitors its operation. In addition, the constituting components of the concepts network undergo selectable changes depending upon the reception of flux which can cause only parts of the concepts network to change while others remain unchanged. This fact leads to the conclusion that the concept network operations can be described by recursive partial functions. For instance, the artifacts of a UbiHealth environment can provide sets of data structures that leave the built knowledge unaffected. Therefore, the employment of partial functions provides the advantage of using basic functions which can be closed under complex operations of composition and primitive recursion leading to the following expression:

$$\mathfrak{K}^k = \begin{cases} \mathfrak{K}^0 = c, c \in S_{init} = \{c_1, c_2, \dots c_n\} \\ \mathfrak{K}^k \mathfrak{K}^{k-1}, & k \in S_{VS} \\ 0, & k \notin S_{VS} \end{cases}$$
(14)

The $\mathfrak{K}^k$ function is a total and computable partial function as it is formed by the collection of partial functions and closed under some fundamental operations for forming new functions from old ones. Therefore, the mathematical representation of the flux is given as a function of the built knowledge: in continuous form

$$F = \frac{dF}{dt} = \frac{d}{dt}(\mathfrak{K}) \quad (15a)$$

or in discrete form

$$F = \Delta F = \Delta\left(\mathfrak{K}^k - \mathfrak{K}^{k-1}\right). \quad (15b)$$

7 Discussion

The sensed data of the carried processes in the PS space support the built flux that causes the development of the corresponding knowledge in the VS. Any change in the processes' workflow results in an analogous change in flux and consequently, the corresponding effects on the developed knowledge. Thus, flux can be used as that magnitude which senses the functionality of the PS and affects the knowledge of the VS space. Therefore, there always exists a relationship among the processes carried in PS, the developed flux, and the knowledge developing in the VS space.

The carried processes in the PS space and the development of knowledge in the VS space, they are related with direct proportional but not linear relationship. The rate of change of the carried processes provides the rate of change in the developing flux between the PS and VS spaces. In a similar fashion, the given instant of flux causes a definite rate of change in the VS space expressed by the development of knowledge. In other words, flux expresses the dynamic changes in the processes of the PS space and the dynamic changes in the built knowledge, as it is presented in (16) below:

$$F = \frac{d(P_n)}{dt} = \frac{d(K_n)}{dt} \quad (16)$$

where P_n and K_n represent the sets of the values domains of the computable functions describing the processes in the PS and the concepts network in the VS spaces respectively.

The engineering design refers to the design of the symbiotic partner agent software which is in between the PS and the VS spaces. The symbiotic partner agent is equipped with adequate modules which are connected with properly defined internet of things (IoT) development in order to sense the functionality of the carried processes in the PS space. The symbiotic partner agent receives the data structures obtained by the IoT installation in order to develop the built flux. The developed flux feeds the software modules of the symbiotic partner agent in order to develop adequate representations of concepts networks. The functionality of the concept networks adjusts the knowledge status to correspond to the needs of the PS. The changes in the built knowledge feed the decision-making modules of the symbiotic partner agent to react accordingly to the carried processes in the PS space.

The operation of the symbiotic partner agent software interacts with the works, procedures, and processes taking place in the PS space by receiving the produced data. The developing flux passes the flows of data to the symbiotic partner agent to develop the corresponding concepts network. The developing knowledge by the concepts networks feeds the decision-making module to cause the symbiotic partner agent to export the data to those data structures required in order to transfer the knowledge impacts to the carried processes of the PS space. Therefore, knowledge governs the functionalities of the processes operating in the physical domain.

8 Conclusion

The formal mathematical representation of knowledge with the application of Denotational Mathematics provide the opportunity to represent constituting elements and quantified magnitudes. The formal Denotational Mathematics framework provide adequate mathematical tools to perform manipulations on the rigorously defined elements, magnitudes and their relationships which are required to express the built knowledge.

Between PS and VS, the developed knowledge is caused by the presence of flux which is defined by the update of the data structures transferred and exchanged between the two spaces. Flux is a magnitude proportional to knowledge and it is used to perform knowledge-able control on the physical domain. The formal mathematical definition of flux provides the capability to use mathematical tools to determine the instant knowledge, the processes status, and the capabilities of the symbiotic partner agent.

The operation of the symbiotic partner agent facilitates the interactions between the PS and VS spaces. The interaction of the symbiotic partner agent with the computing environment, e.g. UbiHealth environment, provides the capability of applying direct knowledge-based control of the carried processes.

The future research plans focus on the development of a symbiotic partner agent prototype. The prototype is expected to provide the basis for experimentation on the uses of flux in the development of knowledge and knowledge-based control on the surrounding computing environments.

The prototype is going to present the attributes mentioned above with the static and dynamic facilities provided by Denotational mathematics. The symbiotic partner agent will be designed as an internet service implementing the quantization of the continuous magnitude of Knowledge Flux along with the relationship with time, i.e. the consideration of knowledge as a function time.

References:

[1] Abowd G.D., Dey A.K., Brown P.J., Davies N., Smith M., & Steggles P. (1999) Towards a Better Understanding of Context and Context-Awareness. In: Gellersen HW. (eds) Handheld and Ubiquitous Computing. HUC 1999. Lecture Notes in Computer Science, vol. 1707. Springer, Berlin, Heidelberg.

[2] Anagnostopoulos, C., A. Tsounis, S. Hadjiefthymiades. (2007). Context awareness in mobile computing environments. Wireless Personal Communications, 42 (3), 445-464.

[3] Armand, A., Filliat, D. & Ibañez-Guzman, J. (2014). Ontology-based context awareness for driving assistance systems. IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, pp. 227-233.

[4] Brickley, D., & Guha, R. (2001). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft. http://www.w3.org/TR/PR-rdf-schema

[5] Bricon-Souf, N., & Newman, C.R. (2007). Context awareness in health care: A review, International Journal of Medical Informatics, 76(1), 2-12.

[6] Chen, T., Min, M.R., & Sun, Y. (2018). Learning k-way d-dimensional discrete codes for compact embedding representations. arXiv preprint arXiv:1806.09464.

[7] Dalley, J., & Hamilton, B. (2000). Knowledge, context and learning in the small business. International Small Business Journal, 18(3), 51-59.

[8] Dey, A.K., Salber, D., Abowd, G.D. & Futakawa, M. (1999). The Conference Assistant: combining context-awareness with wearable computing. Digest of Papers. Third International Symposium on Wearable Computers, San Francisco, CA, USA, 21-28.

[9] Dörpinghaus, J., Stefan, A., Schultz, B., & Jacobs, M. (2020). Towards context in large scale biomedical knowledge graphs. [Online]. Available: http://arxiv.org/abs/2001.08392

[10] Gandon, F.L., & Sadeh, N.M. (2004). Semantic web technologies to reconcile privacy and context awareness, Journal of Web Semantics, 1(3), 241-260.

[11] Gellersen, H.W., Schmidt, A. & Beigl, M. (2002). Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts. Mobile Networks and Applications 7, 341–351.

[12] Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2330-2339.

[13] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, N., Altmann, J. & Retschitzegger, W. (2003). Context-awareness on mobile devices - the hydrogen approach. 36th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 10-18.

[14] Kinsner, W. (2007). Towards cognitive machines: Multiscale measures and analysis. International Journal of Cognitive Informatics and Natural Intelligence, 1, 28–38.

[15] Korpipää P., & Mäntyjärvi J. (2003) An Ontology for Mobile Device Sensor-Based Context Awareness. In: Blackburn P., Ghidini C., Turner R.M., Giunchiglia F. (eds) Modeling and Using Context. Lecture Notes in Computer Science, 2680. Springer, Berlin, Heidelberg

[16] Koskinen, K. U., Pihlanto, P., & Vanharanta, H. (2003). Tacit knowledge acquisition and sharing in a project work context. International journal of project management, 21(4), 281-290.

[17] Lieto, A., Lebiere, C. & Oltramari, A. (2018). The knowledge level in cognitive architectures: Current limitations and possible developments, Cognitive Systems Research, 48, 39-55.

[18] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020). K-BERT: Enabling Language Representation with Knowledge Graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(03),

2901-2908.
https://doi.org/10.1609/aaai.v34i03.5681.

[19] Lukowicz, P., Pentland, S. & Ferscha, A. (2012). From Context Awareness to Socially Aware Computing. IEEE Pervasive Computing, 11(1), 32-41. doi: 10.1109/MPRV.2011.82

[20] Pedrycz W. (2021). From Data to Information Granules: An Environment of Granular Computing. IEEE 20th Int'l Conf. on Cognitive Informatics and Cognitive Computing (ICCI*CC'21), Banff, AB., Canada, IEEE CS Press, Oct., p.2.

[21] Pomerol, J.C., & Brézillon P. (2001). About Some Relationships between Knowledge and Context. In: Akman V., Bouquet P., Thomason R., Young R. (eds) Modeling and Using Context. Lecture Notes in Computer Science, 2116. Springer, Berlin, Heidelberg.

[22] Preuveneers, D., & Berbers., Y. (2008). Internet of Things: A Context Awareness Perspective. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, Auerbach.

[23] Randell, C., & Muller, H. (2000) Context awareness by analysing accelerometer data. Digest of Papers. Fourth International Symposium on Wearable Computers, Atlanta, GA, USA, 175-176.

[24] Ranganathan, A., & Campbell, R.H. (2003). An infrastructure for context-awareness based on first order logic. Personal and Ubiquitous Computing, 7(6), 353-364.

[25] Rosemann, M., Recker, J., Flender, C., & Ansell, P. (2006): Understanding Context-Awareness in Business Process Design. Proc. of the 17th Australasian Conference on Information Systems. Australia Association for Information, Adelaide, Australia.

[26] Sachan, M. (2020). Knowledge Graph Embedding Compression. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2681-2691.

[27] Sarivougioukas, J., Vagelatos, A., Parsopoulos, K., & Lagaris, I. (2017). *Home UbiHealth,* Encyclopedia of Information Science and Technology, Fourth Edition, IGI Global.

[28] Sarivougioukas, J., & Vagelatos, A. (2022). Fused Contextual Data with Threading Technology to Accelerate Processing in Home UbiHealth. International Journal of Software Science and Computational Intelligence (IJSSCI), 14(1), 1-14.

[29] Yousheng, T., Wang, Y., & Hu, K. (2009). A knowledge representation tool for autonomous machine learning based on concept algebra. In Transactions on Computational Science V, 143-160. Springer, Berlin, Heidelberg.

[30] Wang, Y. (2012). On denotational mathematics foundations for the next generation of computers: cognitive computers for knowledge processing. J Adv Math Appl 1(1), 118–129.

[31] Wang, Y., Ngolah, C., Zeng, G., Sheu., P.C., Chiy, P., & Tian, Y. (2010). The Formal Design Model of a Real-Time Operating System (RTOS+): Conceptual and Architectural Frameworks. International Journal of Software Science and Computational Intelligence 2(2), 105-122.

[32] Wang, Y., Baciu, G., Yao, Y., Kinsner, W., Chan, K., Zhang, B., & Hameroff., S. (2010). Perspectives on cognitive informatics and cognitive computing. International Journal of Cognitive Informatics and Natural Intelligence, 4(1), 1-29.

[33] Wang, Y., Zatarain, O., & Valipou, M. (2017). Building cognitive knowledge bases sharable by humans and cognitive robots. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 3189-3194.

[34] Wang, Y. (2015). Concept algebra: A denotational mathematics for formal knowledge representation and cognitive robot learning. Journal of Advanced Mathematics and Applications. 4(1), 61-86.

[35] Wang, Y. (2014). On a novel cognitive knowledge base (CKB) for cognitive robots and machine learning. International Journal of Software Science and Computational Intelligence. 6(2), 41-62.

[36] Wang, Y. (2008). On concept algebra: A denotational mathematical structure for knowledge and software modeling." International Journal of Cognitive Informatics and Natural Intelligence. 2(2), 1-19.

[37] Wang, Y. (2007). The OAR model of neural informatics for internal knowledge representation in the brain. International Journal of Cognitive Informatics and Natural Intelligence. 1(3), 66-77.

[38] Wang, Y. (2009). Toward a formal knowledge system theory and its cognitive informatics foundations. In Transactions on Computational Science V, 1-19. Springer, Berlin, Heidelberg.

[39] Wang Y., Karray F., Kaynak O., Kwong S., Leung H., Plataniotis K.N., Hou M., Rudas I.J., Tunstel E., Trajkovic L., and Kacprzyk J. (2021), Perspectives on the Philosophical, Cognitive and Mathematical Foundations of Symbiotic Autonomous Systems (SAS), *Philosophical Transactions of Royal Society (A)*, 379(x):1-16, Oxford, UK.

[40] Wang Y., Hou M., Plataniotis K.N., Kwong S., Leung H., Tunstel E., Rudas I.J., and Trajkovic L. (2021), Towards a Theoretical Framework of Autonomous Systems Underpinned by Intelligence and Systems Sciences, *IEEE/CAS Journal of Automatica Sinica*, 8(1), 52-63.

[41] Weiser, M. (1999). The computer for the 21st century. ACM Mobile Computing and Communications Review, 3(3), 3-11.