

A Holistic Approach to Cognitive Coordination

Benjamin Hadorn, Michèle Courant, and Béat Hirsbrunner

PAI Research Group, Informatics Department
University of Fribourg, Switzerland

{benjamin.hadorn,michele.courant,beat.hirsbrunner}@unifr.ch
<http://www.unifr.ch>

Abstract. A new holistic approach defining and dealing with coordination in smart environments is presented. Coordination has been studied for many years, but a holistic approach from a generic theoretical model to a pervasive application has never been proposed. Our approach defines a generic model in order to understand and develop coordination aspects at a high level of abstraction. The model should help to analyse and design context-, activity- and situation-aware applications for smart environments. But, it should also be generic enough to be applicable to other problem domains. In this paper we focus only on the modelling part. Our model is built of an abstraction continuum, starting with the notion of entity, interaction, evolution and rules. The notion of enactive entity is introduced on the most abstract level of the continuum. It encompasses consciousness and intentional behaviour, thus leading to cognitive coordination.

Keywords: Generic coordination · coordination model · cognitive coordination · abstraction continuum · enactive entity · consciousness · evolution · interaction · activity · context awareness · observer · point of view

1 Introduction

Coordination as such is an important aspect for understanding and managing any dynamic and complex system. For instance in pervasive computing environments daily activities are coordinated by many natural and artificial circumstances. Since the publication of the coordination theory [23] by T. W. Malone and K. Crowston, a lot has been published around coordination. A few publications treat theoretical and philosophical aspects such as "The muse in the machine" by D. Gelernter [14]. Also, a few models have been published, like the generic coordination space by D. Gelernter et al. [15], and the encapsulation coordination model ECM by O. Krone et al. in [19]. But, most of the work in computer science has focused on coordination languages like LINDA [15], MANIFOLD by F. Arbab et al. [2], and TuCSoN [25] by A. Omicini et al.

We claim that there is still a significant gap between the theoretical and concrete coordination models and languages. Our goal is to gain a better understanding of coordination in general and for pervasive computing in particular.

We aim at describing the world around us using a holistic and generic model through the use of abstract terminology and high level definitions. The model describes how a system can be constructed by introducing the notion of an entity and its topology. The system dynamics is described as interactions between entities, the basic brick for any dynamic behaviour and activity. One special interaction is observation. It is used by smart entities to perceive the system. Further on the abstraction continuum enactive entities emerge when smart entities gain awareness about themselves and about their surroundings. Enactive entities are able to gain knowledge and to act intentionally. These concepts are the base for further understanding of *coordination*.

The paper is structured as follows: in section 2 the related research work on coordination is presented. Section 3 defines the basic components, section 4 denotes the topology of these components and section 5 introduces the concept of smart and enactive entities, which includes among others the definition of awareness and of cognition. All components are put together into a generic coordination model in section 6. Section 7 concludes the present stage of our work and gives an outlook of future research. The concepts of our cosmology are not totally orthogonal and therefore some definitions (e.g events, time) depend on concepts defined later in this text.

2 Related Work

This section is divided into theories, models and languages. Often the referenced papers cover more than just one aspect.

2.1 Theories

T. W. Malone and K. Crowston's coordination theory [23] is a well known and often referenced paper in terms of coordination. They defined coordination as "*managing dependencies between activities*". When there is no interdependence no coordination is needed. Dependencies occur between activities of actors (e.g. humans, computers). When several actors perform an interdependent activity it may lead to conflicting interests. Coordination processes are needed to solve these conflicts (e.g. first come/first serve, priority order). The focus of Malone and Crowston was not only to define coordination for computer science. Their definition aimed at being valid for many other kinds of systems like sociological, biological and economical systems.

The cognition of autonomous entities is an important concept for building a generic coordination model. An interesting theoretical and philosophical approach was given by D. Gelernter in "*The muse in the machine: Computerizing the Poetry of Human Thought*" [14]. He describes a cognitive spectrum, from high focus to low focus, or low level attention and dreaming. Creativity may happen by the loss of control over the thought stream. Once control is regained the smart entity (e.g. a human) becomes aware of new thoughts. Further, he

defined *emotions* as "*a mental state with physical correlates*" and "*a felt state of mind*". He stated that emotions are "*glueing low-focus thought streams together*".

Another way to understand and model cognitive systems, called enaction, was namely developed by F. Varela et al. [32]. Contrasting with representational approaches of knowledge, anchoring cognition in the capacity of the mind to reflect the world, enaction is a constructive approach defining cognition as the capacity of living beings to give meaning to the reality they are experiencing. Learning, as an incremental process, is inseparable from their capacity (1) to initiate actions, which (2) themselves guide interpretation, and (3) to feed the initiative of action by intention. Perception becomes an active process involving expectations, i.e. producing hypotheses, instead of a mechanism mirroring a given environment. Enaction, also called by Varela "faire-émerger" (do-emerge), thus results from a circular dynamics articulating intentional action and perception. It is thus going beyond emergentism, which considers emergence as a sufficient mechanism for explaining the construction of complex systems and of cognition starting from elementary laws.

Activity theory from V. Kaptelinin and B. A. Nardi [17], [24] and K. Kuutti [20] gave fundamental input to build our model. Activity theory plays an important role in human-computer interaction. As described in [24] the goal of activity theory is to understand how human activity and consciousness are linked and what role each object plays within an activity. B. A. Nardi stated that the theory *offers a set of perspectives on human activity and a set of concepts for describing that activity*. An interdisciplinary framework usable to study different human activities was proposed by K. Kuutti in [20]. He stated that *actions are always situated into a context* and that it is impossible to understand actions without that context. Action and the minimum of context together is called *an activity*. Activities are dynamic and continuously change in a nonlinear manner.

Interactions are closely linked with activities. They have been well studied in computer science, e.g. in fields like HCI research [11], [6] or swarm intelligence [12]. Here interaction is understood as an exchange of information (communication) between two systems (e.g human, computer). Often interactions are done with a clear intention, such as pushing a keyboard button expecting some reaction on the computer screen. A. Dix proposes a new interaction paradigm, called *incidental interaction* [11], where the attention of the user is much lower than in traditional interactions. The actions "*are performed for some other purpose*", but trigger an incidental reaction of the system. A. Dix put interaction into a continuum, from *intentional* interaction to *incidental* interaction, depending on the purpose of the user.

2.2 Models

Our current research on coordination is influenced by the work of E. Schwarz [29], [30] who defined a holistic meta model for general systems and especially complex ones. The objects follow not only pure physical laws but are also influenced by networks connecting them on abstract and existential planes. His focus is on phenomena and stages leading to complexity. He proposes a spiral of

self organisation describing how any complex system can step from one level of organisation and complexity to another one. The dynamic of self organisation is driven by two forces: one leading to disorder and complexification/ heterogenisation, and one leading to order/homogenisation corresponding to entropic drift.

He identified seven stages of increasing abstraction in which a system becomes more and more complex and autonomous during its long term evolution. Going through those stages a system gains autonomy and finally ends up as a holistic being (figure 1).

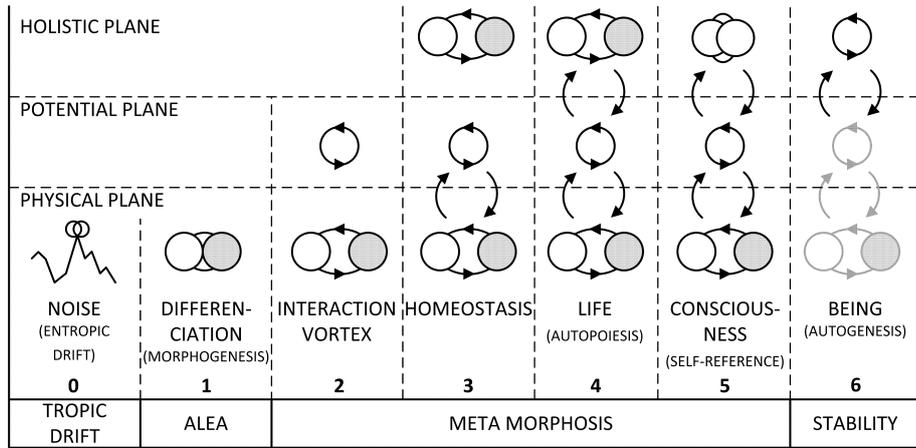


Fig. 1. The emergence of a system from noise to a being inspired by E. Schwarz [29].

- (0) It starts with an entropic drift as a natural trend of the preceding system. This could drive the system far from equilibrium.
- (1) Differentiation: Simple structures emerge on the physical plane, differentiating themselves from the medium through some basic interaction between entities.
- (2) Interaction Vortex: The physical entities interact with each other, one relation is emerging on the potential plane. This relation characterises the system dynamics, which acquires a certain stability, due the circularity, e.g. there is a cyclic exchange of energy and matter.
- (3) Homeostasis: In this stage the system starts to regulate itself. The regulation is based on feedback loops between physical and potential plane, for example some physical parameters are regulated, like temperature or pressure. On the holistic plane the object (physical structure) and its image (network of potentialities) start to reference each other.
- (4) Autopoiesis: A System reaching this stage is able to maintain not only parameters but structures through the interaction between potential and

- physical plane. The system becomes capable of producing itself (It incarnates a network of causality which can produce the system).
- (5) Self-reference: The system starts to reference its own image. The degree of self-reference can be interpreted as the level of consciousness.
 - (6) Autogenesis: The autogenesis leads to the autonomy of a system, it becomes a "being". It is able to create its own laws.

A system goes through those steps several times (spiral). After reaching the stability stage, noise could lead to actualisation of potentialities or disposition of the system (Entropic drift). In the "Alea" the system does not necessarily end up with a new viable system, but could destroy itself or could continue with minor changes.

A technical view on coordination was given by D. Gelernter and N. Carriero in [15]. They stated that any system (computer system) can be split into *computation* and *coordination*. They looked at coordination as the glue fitting computational parts together. They also claimed that coordination should be orthogonal to computing, meaning that computing deals with information processing whereas coordination manages the exchange of information between active agents (computer, human).

Modelling coordination is a main focus of our research group for over two decades. The following models are significant works.

The Φ -CoLa coordination model was a first attempt by C. Renevey et al. for introducing the physical dimension (Φ) into the computing space [27]. The model was oriented towards highly dynamical systems and complex system organisation. It used a wide range of communication protocols based on topologies, and formalised the notion of communication medium.

The encapsulation coordination model ECM by O. Krone et al. [19] was first used in massive number crunching and multi-agent systems. The encapsulation allows to hierarchically structure the coordination space into *blobs*. Active entities (processes) can communicate through connections with each other. The connections are established between matching ports, openings of entities and blobs to the external world.

Based on ECM, M. Schumacher split coordination into two major classes: *objective* and *subjective coordination* [28]. Objective coordination treats mainly the objectives or inter-agent dependencies. This includes the organisation of the environment and the interaction between agents. Subjective coordination deals with intra-agent dependencies towards other agents. The dynamics between objective and subjective coordination is further analysed by A. Omicini et al. in [25]. On the level of subjective coordination agents reason about coordination by inspecting the coordination media. Some coordination artifacts (processes, rules) are designed and applied to the media. On the level of objective coordination the embedded artifacts are exploited.

A first attempt of generic semantic coordination model for pervasive computing was proposed by A. Tafat et al. in [31]. Key components of the model were entities, environment, physical and social law, and ports for communication. An

ontological representation was chosen in order to share the knowledge and to achieve context reasoning.

With UbiDEV a unified model for pervasive and heterogeneous systems was introduced by S. Maffioletti [22]. His bio-inspired model describes a cognitive system based on perceiving (observing), representing and acting (figure 2). The system autonomously adapts its structure to changes in the environment by enriching it with adequate services and interaction models. Different resources and services are combined dynamically to perceive the environment and act upon changes.

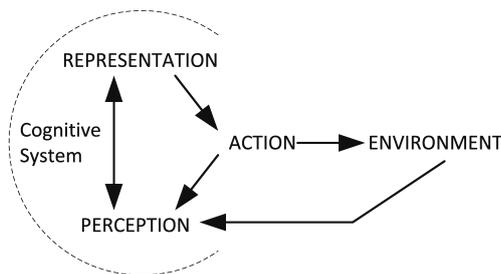


Fig. 2. Coordination of perceiving, representing and acting within a cognitive system in UbiDEV [22].

The framework uMove by P. Bruegger et al. [5], [6] is continuing the middleware generation by using *motion as primary input modality*. New concepts for pervasive computing have been introduced in a conceptual model, like *everything is an entity* and *situation is made of context and activity*. It allows to combine captured context and activity data into situations alerts. Further model consists of the notion of observer and its point of view.

2.3 Coordination Languages and Frameworks

Since the focus of this paper is modelling coordination we give only a short overview of coordination languages and frameworks.

An overview of different coordination languages is given by G.A Papadopoulos and F. Arbab in [26]. They classified the coordination languages into *data-driven* and *control-driven*. A main example of data-driven languages is LINDA [7]. It uses a tuple space to exchange information among processes in distributed and parallel systems. It decouples processes from each other so they become time independent from each other. Many other languages follow this principle, like TuCSoN (Tuple Centres over the Network) [25] and BTS-SOC (the biochemical tuple space for self-organising coordination) [33].

A main example for control-driven languages is MANIFOLD [2]. MANIFOLD's purpose is to *manage complex interconnections among independent,*

concurrent processes". The language describes how processes are connected through ports and how to react to events.

The following languages are contributions of our group. CoLa [18] is a control-driven language which was designed for massively parallel computing, especially for distributed artificial intelligence. STL [19] and STL++ [28] also belong to control-driven languages. Like MANIFOLD they define how agents can communicate with each other through dedicated ports and communication channels. Both are based on our coordination model ECM.

These theories, models and languages are substantial and provide fundamental input for the design of our generic coordination model. In contrast to them, this model is put on a higher level of abstraction to be more independent of any problem domain.

3 Basic Components

The model is based on an abstraction continuum. The continuum starts with three basic components, entities, interactions and rules, and how they evolve.

3.1 Entity

Any "nameable thing" in a system is called an *entity*. The term *entity* includes all physical constructs (e.g. humans, buildings, molecules) as well as all virtual or logical constructs (e.g. ideas, thoughts, mathematical formulae).

Definition 1 (Entity). *An entity is anything that actually or potentially exists.*

The set of all entities is called *the universe* \mathbb{E} , that is $e \in \mathbb{E}$. The universe includes all *actual entities* E and *potential entities* \bar{E} .

In particular two kinds of entities can be distinguished: *interactions* denoting transformation processes and *rules* denoting restrictions for interactions. There is no dualism between these kinds of entities. An entity might be both at once.

3.2 Interaction

The expression "interaction" is used in general, and applicable to many domains such as physics, chemistry or social science.

Definition 2 (Interaction). *Interaction is an exchange of anything between two or more entities.*

The exchange could be of energy-matter fluxes in the physical domain [29], or it could be the exchange of information between a human and computer. All dynamics in a system are interactions, even those which are unobvious, such as observation and broadcast communication. An interaction i can change a set of entities by changing, removing or adding entities to it:

$$i : E \rightarrow i(E) = E' \text{ with } E, E' \subset \mathbb{E}. \quad (1)$$

Generally speaking interaction changes the current set of existing entities E to E' . This change is called *a transformation*. Interaction can be seen as a transformation function within the domain of entities, $i : \mathbb{E} \rightarrow \mathbb{E}$.

3.3 Evolution and Rules

An important aspect of dynamic systems is the study of how they evolve and change. Interactions can be sequenced e.g. $E_0 \rightarrow E_1 \rightarrow E_2 \dots$. The sequence of interactions is called *an evolution* \mathcal{E} of E_0 . In this context E_j is called *a state* of evolution \mathcal{E} containing all existing entities at that stage.

Definition 3 (Evolution). *An evolution $\mathcal{E}(E)$ is the set of sequences formed by interactions starting at E .*

Without any regulation everything could happen in universe \mathbb{E} . Rules restrict and limit the possibilities of the evolution and some of the evolution paths become impossible (figure 3).

Definition 4 (Rule). *A rule is a regulation for interaction.*

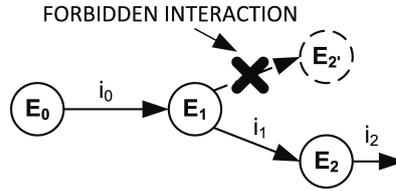


Fig. 3. Rules applied to an evolution path.

Rules lie on a spectrum from *strong* to *weak*. Strong rules are not able to evolve, they can be seen as invariants of the system. For instance physical law and mathematical axioms can be expressed by strong rules. Weak rules on the contrary can be created, changed and destroyed. Rules are able to express general properties of evolution, e.g. determinism or non-determinism.

The evolution can be expressed using directed graphs, where the vertices represent the states and the edges represent the interactions transforming the states. Drawing graphs can be used to sketch the evolution. The simplest case is a deterministic single path evolution (figure 4a). Non-determinism generates a multipath evolution. Several interactions could possibly happen at each state (figure 4b). Typically the non-determinism chooses exactly one of the possible interactions. The graph can be interpreted as an XOR-Graph.

Remark: In the case of *Many-Worlds Interpretation* (B. S. DeWitt in [10]) all possible interactions happen simultaneously. This splits up the universe into two or more parallel existing universes. The graph can be interpreted as an OR-Graph.

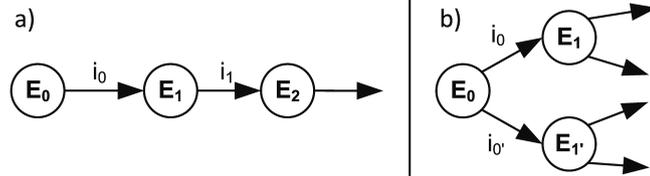


Fig. 4. (a) a single path evolution. (b) a multipath evolution generated by non-deterministic interactions i_1 and i_2 .

Parts of a system can evolve independently. The evolution can be split into several *partial evolution branches*. Each branch has its own sequence of transformations and states. Branches can merge together once their states become dependant again (figure 5).

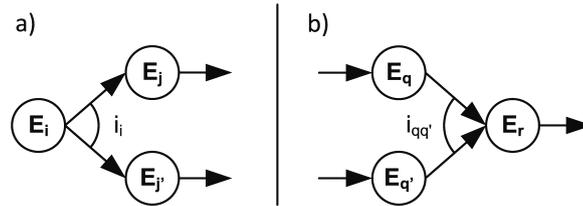


Fig. 5. Partial evolution branches can emerge (a) or disappear (b) during the system evolution.

4 Topology

The entities are arranged in the universe under different aspects such as space-, time- and logical-relations to name a few. Such an arrangement is called a *topology of entities*, which constitutes the next level in our abstraction continuum. First the common denominators, the *attributes* and *relation*, are introduced. Second we define dependency and composition, two special relation types. The definition of spacial and temporal topologies leads to metrics and time. Finally this section ends with the special temporal topology, the activities, roles and context.

4.1 Attributes and Relations

An entity could help to describe a characteristic of another entity. For instance a color entity "blue" could describe a car entity. We call the describing entity an *attribute*.

Definition 5 (Attribute). *If an entity describes another entity, it is called an attribute.*

Two entities are related if they share some attributes. As in our example, "blue" could describe other cars, all "blue" cars are related through the shared attribute "blue".

Definition 6 (Relation). *If two or more entities share some attributes, the sharing is called a relation.*

A relation L between two entities e_1 and e_2 is called a *binary relation* and is denoted by $L(e_1, e_2)$. Let $A(e_i)$ be the set of all attributes of the entity e_i . $L(e_1, e_2)$ exists if and only if $A(e_1) \cap A(e_2) \neq \emptyset$.

Generalised, an *n-ary relation* L can be denoted by $L(e_1, e_2, \dots, e_n)$. Accordingly L only exists if there is at least one shared attribute among e_1, e_2, \dots, e_n .

$$\bigcap_{i=1}^n A(e_i) \neq \emptyset \leftrightarrow L(e_1, e_2, \dots, e_n). \tag{2}$$

Entities are organised through relations and entities of higher abstraction could emerge from relations [29], [30]. A new level of abstraction is added on our abstraction continuum (figure 6). Whereas interacting entities are situated on the *entity plane*, the relations and entities of higher abstraction are situated on the *relation plane*.

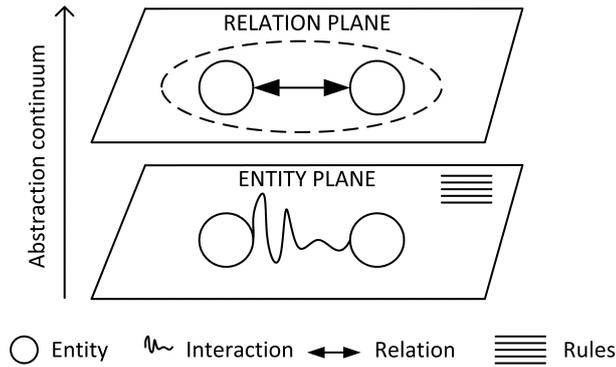


Fig. 6. Two abstraction planes: entity plane and relation plane.

4.2 Dependency

When two entities e_1 and e_2 interact with each other, they are influenced by their interaction. If e_2 is required by e_1 to evolve in a certain given direction, the relation between e_1 and e_2 is called a *strong dependency*. If there exist alternative entities to satisfy the requirements of e_1 , it's called a *weak dependency*.

Definition 7 (Dependency). *If an entity is transformed by other related entities, the emergent relation is called a dependency.*

The degree of dependence through a relation is denoted by d and is expressed on a spectrum such as from "strong" to "weak".

$$D : (e_1, e_2) \rightarrow d = D(e_1, e_2) . \tag{3}$$

If an entity e_1 depends on e_2 through the dependence d , we can inversely express the power p of e_2 upon e_1 (C. Castelfranchi [8] and R. M. Emerson [13]).

$$P : (e_2, e_1) \rightarrow p = P(e_2, e_1) . \tag{4}$$

Between two entities e_1 and e_2 two dependencies $d_1 = D(e_1, e_2)$ and $d_2 = D(e_2, e_1)$ could exist. In general d_1 and d_2 are not balanced. In order to become stable, systems tend to balance dependencies while they evolve. The dependency situation is therefore changing during evolution. This imbalance is one of the "motors of the evolution".

An entity can namely balance its dependencies in two ways: (1) reducing the dependency by cultivating alternative dependencies, and (2) increasing the dependency by reducing alternative dependencies (figure 7).

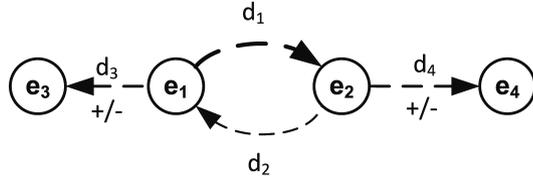


Fig. 7. The imbalance between d_1 and d_2 may be changed by increasing and/or decreasing d_3 and/or d_4 .

Remark: Considering the dependency between two entities, there are two basic mutual dependencies schemas, which may lead to interesting organisation phenomena: (1) Agonism, where each one is cooperative with the other one and (2) antagonism where each one is competing the other. Now living system show a combination of agonism and antagonism, called ago-antagonism, which appears to be a key for the emergence of persistent structures in complex systems [4].

4.3 Composition

This section focuses on the emergence of larger compound entities, that is to say that two or more entities e_j can form a new abstract entity e . The new entity e is called a *composition*, each child entity e_j a *component* and the dependency between the composition and its components an *affiliation*.

Definition 8 (Composition). *An entity which is made of entities is called a composition.*

Definition 9 (Affiliation). *The relation between composition and its components is called an affiliation.*

A weak affiliation is called an *aggregation* whereas a strong affiliation forms an inseparable whole, called a *compositron*.

A composition is of higher abstraction than its components. All components e_j of e are considered to be *internal to e* . If two compound entities e_A and e_B share at least one component then e_A is intersecting e_B and vice versa. All entities not inside or intersecting e_A are *external to e_A* .

4.4 Space

The space is a boundless extent in which entities have a position relative to each other.

Definition 10 (Space). *A set of entities conforming to a specified structure is called a space.*

The space is defined by the characteristics of composition and rule. The rule characteristic defines the structure of the space. For instance the Euclidean Space uses axioms of inner product as a rule. The composition characteristic represents the realm of the existing entities E .

The space can be subdivided into several subspaces, which are called *places*. An entity can be related to a place.

Definition 11 (Location). *The location is a relation between an entity and a place.*

Inside, outside and next-to are examples of location relations.

4.5 Time

Time is a manifestation of the interaction between entities and an epiphenomenon of the evolution. No time exists when there are no interactions. Time can be denoted as a monotonous non-decreasing function based on events. Events are defined as:

Definition 12 (Event). *An event is the happening of a significant state.*

Here significant is related to the concept of observer which is introduced later. In the evolution of a system the events can be ordered by the *happened-before* relation (cf. e.g. [9], pp. 596 ff). Time can then be expressed by this ordered sequence of events. Past, present and future can be derived (figure 8). Let the currently existing state E_P be the present then the past contains all events which caused the present state. The past can be expressed by the history H_E , which includes all states from the beginning to the present $H_E = \langle E_j | j < P \rangle$. All possible events caused by the present which could occur is called the future. The future is denoted by all possible evolution paths starting from E_P , $\mathcal{E}(E_P)$.

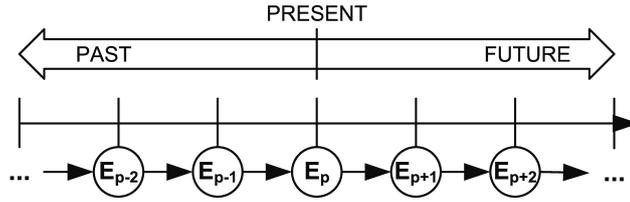


Fig. 8. The notion of past, present and future.

Remarks: From an objective point of view there is no ordering between independent events. Einstein showed in his Special Theory of Relativity, that it depends on the observer in which order the independent events occur. An observer may order such independent events. However, if an event causes another event, then its occurrence happens after the occurrence of the causing event independently of any observation.

There exist other time models. For instance, the model by J.K. Allen focuses on temporal intervals and their relations [1]. Basically it consists in extending the happened-before relation with other relation types, such as "overlapping", "meeting" and "during".

4.6 Activity

The arrangement of interactions leads to a *temporal topology*. A special topology is the temporal composition of interactions, called *activity* [20]. It is an entity of higher abstraction than interactions emerging from interactions achieving a common state, called *target state* (figure 9). Such interactions and the entities involved are said to be *cooperative*.

Definition 13 (Activity). *An activity is a temporal composition of the cooperative interactions and of the involved entities.*

Activities themselves can be put in relation to each other. This leads to other topologies such as compositions of activities (affiliation) or activity flows (happened-before). The affiliation and happened-before relations allow to model sequencing, parallelism and synchronisation of activities.

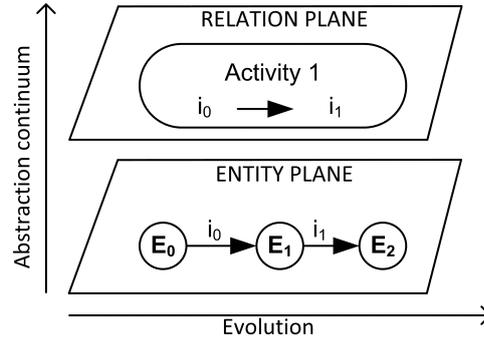


Fig. 9. Basic structure of an activity. Correlates to figure 6, but additional axis of evolution is added.

4.7 Roles

Each entity involved in an activity can behave differently. The entity's behaviour is related to the rules limiting its interaction. To describe the entity's behaviour the notion of "role" is used.

Definition 14 (Role). *The behaviour of an entity involved in an activity is called a role.*

The following list contains some important roles:

- *Actor* for an active entity (proactive or reactive), driver of the activity,
- *Observer* for an entity perceiving the activity,
- *Object* for a passive entity being transformed by the activity,
- *Tool* for any entity mediating the relationship between actors and other entities in the transformation process.

4.8 Context

Entities are influenced by circumstances, called context. Context encompasses any attributes and relations of an entity.

Definition 15 (Context). *The set of attributes and relations describing an entity is called context.*

Often these circumstances must be considered in order to understand the behaviour and characteristic of an entity. For instance the same activity but different context could lead to different ending states.

5 Subjective Components

This section focuses on the subjective components of our model. The subjectivity is based on the concept of *smart and enactive entity* and *observer*.

5.1 Smart Entity and Observer

A smart entity is a composition of two sub entities: (1) an internal activity which can be seen as a "motor" which allows the smart entity to interact with external entities. And, (2) an internal perception space, denoted by \tilde{S} , which is used to keep and cross-link perceived information.

Definition 16 (Smart Entity). *A smart entity is an entity able to act upon perceived events.*

As any entity, smart entities can play different roles. An important role is *the observer*. If a smart entity is able to perceive entities, it is called *an observer*. An observer can only perceive entities within an *observation space*, denoted by S (figure 10), which fulfils the structure of observation, e.g. the measurable space by B.M. Bennett et al. in [3].

Definition 17 (Observer). *An observer is an entity able to perceive entities within an observation space.*

The interaction of an observer with its observed entities is called *observation*. The observation constructs information through sensing, focusing and interpreting the observed entities, which leads to changes in the internal perception space \tilde{S} . The perception space keeps information about an observed entity e , called *image of an entity* \tilde{e} . In general, but not necessarily, the observed entities are not influenced by the observation.

Observation is relative to the observer's view which includes (1) his point of view [5], [16] and (2) his capabilities for constructing information. This leads to individual (biased) perceptual inference.

Definition 18 (Point of view). *A point of view is a relation between an observer and an observation space.*

An observer can use a *viewer* to sense and focus on entities. The viewer detects, filters, transduces and preprocesses the data before it is used for the further process of inference. Examples for viewers are eyes, ears, glasses, microscopes and sensors.

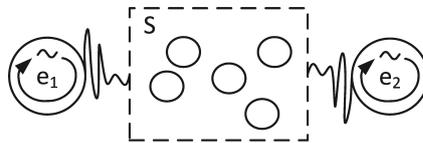


Fig. 10. Two observers observing different entities in the observation space S .

5.2 Observer's Awareness

The individual perception of the observer leads to awareness of events. It allows the observer to be conscious of events, where *being conscious* is perceiving and paying attention to what is happening.

Definition 19 (Awareness). *Awareness is the ability to perceive and be conscious of events.*

The observer's awareness can focus in particular on:

- *topology awareness*: An observer can perceive the topology by mapping relations in S to \tilde{S} . Relations outside S are not perceived. The observer becomes aware of a certain abstraction of space S .
- *state awareness*: The observer becomes aware of states when observed states of the evolution are mapped onto an internal perception space \tilde{S} .
- *role awareness*: An observer can observe the behaviour of an entity. Relative roles are applied to an observed entity's image \tilde{e} .

The observer does not have to choose between one of those awareness aspects. Often they are present all together.

5.3 Enactive Entity

Enactive entities are smart entities able to gain knowledge from their interaction through the enaction space, a combination of internal perception and action spaces. The enaction space allows the entity to become aware of its surroundings or itself (consciousness), and to act intentionally. This leads to cognition [32].

An enactive entity is able to adapt its behaviour or to create new behaviours. This capability, which is enforced by consciousness and intentional acting, is called *cre-adaption*.

Definition 20 (Enactive Entity). *An enactive entity is a smart entity able to cre-adapt itself.*

Similarly to the model of E. Schwarz [29], [30], enactive entities are situated on the top level of our abstraction continuum (figure 11).

5.4 Consciousness

Consciousness emerges as an entity becomes aware of itself and of the surrounding entities. Consciousness is a spectrum expressing how close the entity and its perceptual image are related. The closer the relation is the higher the consciousness of the entity [30] is.

The consciousness spectrum starts on the relation plane where smart entities become aware of events. Higher consciousness is developed on the holistic plane by enactive entities with their cognitive capabilities. It cannot be analysed separately from entities or their relations, but must be conceived as a whole.

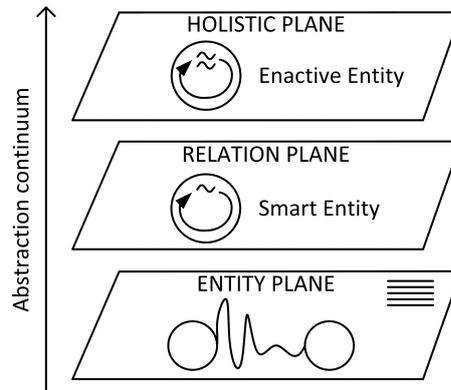


Fig. 11. Smart entities are emergent compositions on the relation plane whereas enactive entities emerge on the holistic plane.

On the holistic plane the following stages are reached while consciousness increases: *self-awareness* which is the ability of an enactive entity to self-reference; *creativity and intuition* which allows the entity to find new ideas and be innovative; and *empathy* allowing an enactive entity to understand and feel the emotions of other entities.

6 Coordination

This section puts the different components of the model together and shows from the coordination point of view how they fit together. First a generic definition of coordination is given, then different levels of coordination are presented.

Coordination controls and directs the interactions and therefore influences evolution. Coordination can be seen as a kind of meta rule.

Definition 21 (Coordination). *Management of evolution, that is controlling and directing interactions, is called coordination.*

Coordination is applied in a space, which can be seen as the coordination medium. Coordination can be put on a spectrum describing different coordination levels, *objective, subjective and cognitive coordination*.

6.1 Objective Coordination

The process of applying rules to interacting entities is called *objective coordination* [28]. Also, the arrangement of entities (topology) based on rules is part of this coordination level, which is situated on the low end on the coordination spectrum. Often it is implicitly managing the evolution and doesn't have a need for an specialised coordinator entity.

In a concrete case, where events are the trigger to coordinate entities (figure 12) it starts with an event, a significant state change. The event triggers certain rules which are applied to the interacting entities. This closed coordination loop can be seen as objective coordination.

6.2 Subjective and Cognitive Coordination

The next level on the coordination spectrum deals with adapting "weak" rules according to perceived states and appropriate conclusions. The adaption of rules follows some basic strategies by smart entities. This level is called *subjective coordination* [25], [28].

In the example (figure 12) a more complex coordination process can be realised using an observer based on a smart entity. It starts with events too. A smart entity perceives the events and becomes aware of the current state. It can finally adapt the rules to coordinate the interacting entities at its will, which leads to subjective coordination.

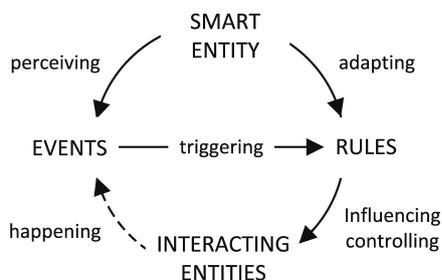


Fig. 12. Example where events are triggering the coordination.

As enactive entities and their cognitive capabilities emerge on the holistic plane we introduce a new level of coordination, called *cognitive coordination*. It deals with holistic coordination strategies and processes based on shared knowledge, creativity and empathy among enactive entities. For instance an enactive entity can create new strategies by itself in order to manage the evolution. It differs from the subjective coordination, since it can not be treated independently from the rest, but rather has to be considered holistically. Coordination processes on this level help to manage and support inspiration and creativity, which is far from adapting rules (subjective coordination).

Figure 13 illustrates the coordination, abstraction, and consciousness spectra.

There are several application fields where cognitive coordination could help to solve operational problems. As an example the domain field of *pervasive computing*, where uncertainty and incompleteness of knowledge about the interacting entities are the basic features that have to be addressed, could be mentioned. A more comprehensive and holistic approach could lead to new range of services

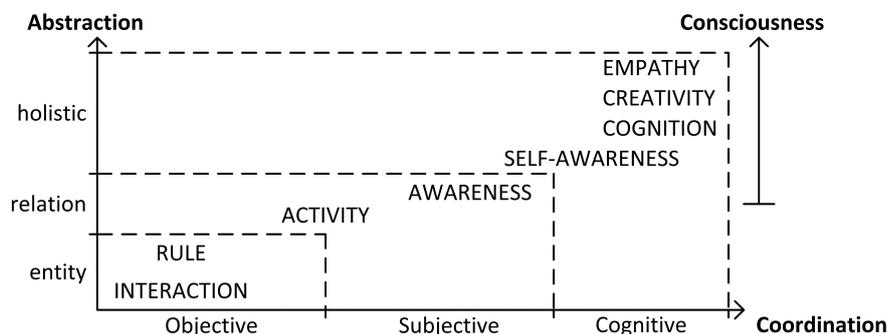


Fig. 13. Coordination, abstraction, and consciousness spectra.

on cognitive aspects. This could lead to reinforce the human being in his feeling to be in control of his life experience in the world of technology.

7 Conclusion and Future Work

We presented a generic cosmology based on a holistic approach and stated that this approach is necessary to integrate enactive entities into a coordination model. The term enactive is used to describe entities which are able to become aware of its surroundings or itself, and can act intentionally. In this paper we extended the objective and subjective coordination with cognitive coordination which allows to integrate such enactive entities into our generic coordination model. The generic coordination model has been presented as a possible base for further research on coordination in this direction. In this sense its important to have a common language and speech about coordination and its related constructs. The model is based on an abstraction continuum which is subdivided into three planes: the entity plane, the relation plane and the holistic plane.

The entity plane describes the basic components of our model, the entities. We showed how the system could evolve through the dynamics of interactions and the restrictions of rules, where interactions and rules are particular kinds of entities.

On the relation plane entities of higher abstraction emerge from relations describing the entity topology. Special relations are explained like dependencies, happened-before (time) and compositions. The concept of subjectivity is explained starting with smart entities, a composition of internal activity and internal perception space. Smart entities are able to take a role as an observer, that is an entity able to perceive entities located in an observation space. We defined awareness as the ability of an observer to be conscious of events happening in the observation space, which is a first step to reach the holistic plane on the abstraction continuum.

Enactive entities emerge on the holistic plane when they gain autonomy and cognitive capabilities.

Coordination is defined as the management of evolution. It can have objective aspects applying rules on interacting entities and subjective aspects involving perception, reasoning and cognition on the level of smart entities. Finally a new level of coordination is introduced, called *cognitive coordination*. It addresses coordination based on processes dealing with cognition, creativity and empathy of enactive entities. The level of cognitive coordination will be addressed in more depth in future research work.

The generic model will help us to study the support of human activities in smart and pervasive environments. We intend to apply the model in a future research step to real life situations of human and machine activities in such environments.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, Nov. 1983.
2. F. Arbab, I. Herman, and P. Spilling. An overview of Manifold and its Implementation. *Concurrency: Pract. Exper.*, 5(1):23–70, Feb. 1993.
3. B. Bennett, D. Hoffman, and C. Prakash. *Observer Mechanics: A Formal Theory of Perception*. Academic Press, 1989.
4. E. Bernard-Weil. Quantum mechanics, mathematics, cognition and action. chapter Ago-antagonistic Systems, pages 325–348. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
5. P. Bruegger. *uMove: a wholistic framework to design and implement ubiquitous computing systems supporting user’s activity and situation*. PhD thesis, University of Fribourg, Switzerland, 2011.
6. P. Bruegger and B. Hirsbrunner. Kinetic User Interface: Interaction through Motion for Pervasive Computing Systems. In *Proceedings of the 5th International on Conference Universal Access in Human-Computer Interaction. Part II: Intelligent and Ubiquitous Interaction Environments*, UAHCI '09, pages 297–306, Berlin, Heidelberg, 2009. Springer-Verlag.
7. N. Carriero and D. Gelernter. Linda in context. *Commun. ACM*, 32(4):444–458, Apr. 1989.
8. C. Castelfranchi. The "Logic" of Power. Hints on how my power becomes his power. Working paper, 2011.
9. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems: Concepts and Design (4th Edition) (International Computer Science)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.
10. B. DeWitt and U. of North Carolina at Chapel Hill. Institute of Field Physics. *The Many-universes Interpretation of Quantum Mechanics*. Institute of Field Physics, Department of Physics, University of North Carolina, 1970.
11. A. Dix. beyond intention pushing boundaries with incidental interaction. In *Building Bridges: Interdisciplinary Context-Sensitive Computing*, pages 1–5, Lancaster, UK, 2002. Lancaster University.
12. M. Dorigo and T. Stützle. *Ant Colony Optimization*. Bradford Company, Scituate, MA, USA, 2004.
13. R. M. Emerson. Power-Dependence Relations. *American Sociological Review*, 27(1):31–41, 1962.

14. D. Gelernter. *The Muse in the Machine: Computerizing the Poetry of Human Thought*. Free Press, 1994.
15. D. Gelernter and N. Carriero. Coordination Languages and their Significance. *Commun. ACM*, 35(2):97–107, Feb. 1992.
16. B. Hirsbrunner, M. Aguilar, and O. Krone. CoLa: A Coordination Language for Massive Parallelism. In *In Proceedings ACM Symposium on Principles of Distributed Computing (PODC)*, pages 14–17, 1994.
17. V. Kaptelinin and B. A. Nardi. *Acting with Technology: Activity Theory and Interaction Design*. MIT Press, Cambridge, 2006.
18. O. Krone, M. Aguilar, and C. Renevey. The cola approach: A coordination language for massively parallel systems. In *University of Fribourg*, pages 29–33, 1994.
19. O. Krone, F. Chantemargue, T. Dagaëff, and M. Schumacher. Coordinating Autonomous Entities with STL. *SIGAPP Appl. Comput. Rev.*, 6(2):18–32, Sept. 1998.
20. K. Kuutti. Activity Theory as a potential framework for human-computer interaction research. In B. A. Nardi, editor, *Context and Consciousness: Activity Theory and Human Computer Interaction*, chapter Activity Theory as a potential framework for human-computer interaction research, pages 17–44. MIT-Press, Cambridge, MA, USA, 1995.
21. L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Commun. ACM*, 21(7):558–565, July 1978.
22. S. Maffioletti. *UbiDev: A Homogeneous Service Framework for Pervasive Computing Environments*. Phd thesis no. 1528, University of Fribourg, Fribourg, June 2006.
23. T. W. Malone and K. Crowston. The Interdisciplinary Study of Coordination. *ACM Comput. Surv.*, 26(1):87–119, 1994.
24. B. A. Nardi. Activity Theory and Human-Computer Interaction. *English*, (2):4–8, 1993.
25. A. Omicini, A. Ricci, G. Rimassa, and M. Viroli. Integrating Objective & Subjective Coordination in FIPA: A roadmap to TuCSoN. In G. Armano, F. D. Paoli, A. Omicini, and E. Vargiu, editors, *AI*IA/TABOO Joint Workshop "From Objects to Agents"*, WOA'03, pages 85–91, Villasimius, CA, Italy, September 2003. Pitagora Editrice Bologna.
26. G. A. Papadopoulos and F. Arbab. Coordination Models and Languages. In *ADVANCES IN COMPUTERS*, pages 329–400. Academic Press, 1998.
27. C. Renevey and M. Courant. Phi-CoLa: Towards a Logico-Physical Coordination Language. In *Proceedings of the Workshop on Future and Practice of Autonomous Systems*, Monte Verita, Switzerland, Sept. 1995. FPAS'95.
28. M. Schumacher. *Designing and Implementing Objective Coordination in Multi-Agent Systems*. Phd. thesis no. 1291, University of Fribourg, Fribourg (Switzerland), December 2000.
29. E. Schwarz. Can Real Life Complex Systems Be Interpreted with the Usual Dualist Physicalist Epistemology - Or is a Holistic Approach Necessary ? *Res-Systemica, Volume N2, Special Issue: Proceedings of the fifth European Systems Science Congress, October 2002, Crete*, page 9 p., 2002.
30. E. Schwarz. On the Nature of Consciousness—On Consciousness in Nature. *AIP Conference Proceedings*, 1303(1):334–342, 2010.
31. A. Tafat, M. Courant, and B. Hirsbrunner. Implicit environment-based coordination in pervasive computing. In *Proceedings of the 2005 ACM symposium on Applied computing, SAC '05*, pages 457–461, New York, NY, USA, 2005. ACM.
32. F. J. Varela, E. Thompson, and E. Rosch. *The Embodied Mind: Cognitive Science and Human Experience*. The MIT Press, Cambridge/MA, 1993.

33. M. Viroli and M. Casadei. Biochemical Tuple Spaces for Self-organising Coordination. In J. Field and V. T. Vasconcelos, editors, *COORDINATION*, volume 5521 of *Lecture Notes in Computer Science*, pages 143–162. Springer, 2009.